Leonardo da Motta de Vasconcellos Teixeira

The Hermite Scheme: An Application to the N-Body Problem

Juiz de Fora - MG, Brazil

2020

Leonardo da Motta de Vasconcellos Teixeira

The Hermite Scheme:
An Application to the N-Body Problem

Thesis for attaining the Master's title through the Graduate Physics Program of the Universidade Federal de Juiz de Fora.

Universidade Federal de Juiz de Fora – UFJF
Departamento de Física, Instituto de Ciências Exatas
Programa de Pós-Graduação em Física

Supervisor: Fernando Sato

Juiz de Fora - MG, Brazil
2020

Leonardo da Motta de Vasconcellos Teixeira

The Hermite Scheme:
An Application to the N-Body Problem

Thesis for attaining the Master's title through the Graduate Physics Program of the Universidade Federal de Juiz de Fora.

Work approved/denied. Juiz de Fora - MG, Brazil, 18 of February of 2020:

Fernando Sato
Supervisor

Ana Cláudia Monteiro de Carvalho
External Member

Maikel Ballester Yusat Furones
Internal Member

Juiz de Fora - MG, Brazil
2020

Acknowledgements

The life of a human being is a Path, an undefined Path, whose only certainty is its end, and which is built and guided by our choices and decisions among the way. Every human being, when they're born, begin their own Path, willing or not, and the ones responsible for its construction are themselves alone.

Each major choice we make is a turn in the Path, and marks a milestone on the way, some to be celebrated, others can be regretted, but all of them to be remembered as they represent teachings that will serve as guides for the rest of the way. It is important to remember that all of the choices that we made up until now are what brought us here, and what made us who we are, no matter if we consider that choice to be bad or not. We must always remember that life is a true chaotic dynamic system, and that if but a single choice had been different, it wouldn't be us here, but someone else telling the tale. Such things aren't to be feared, but embraced, for the unknown is to be marvelled upon.

Many choices led me here. That little child who dreamed of being a scientist has chosen, and now I reach the end of another milestone. Everyone's Path is their own, but that doesn't means that paths do not cross. We meet many people among the way, and each of them contribute somehow to our building our path. The Japanese folk believe that with every person we meet in our life, no matter how briefly the encounter, we create a bond, in their words Enishi, and this leaves a mark, the influence of their passage through our own path. This bond can become a good bond, Goen, or it can be such from the start, otherwise it just a bond, never a bad bond, for everyone we meet in our lives has something to teach us, be it something good, or how not to be bad.

My path is no different, and throughout the way I have met many people to whom I owe being who I am, who accompanied me along the way and helped me reach my goals and conquer my dreams, and to those persons words aren't enough to express my gratitude, but I will try my best to convey it. The first of all the encounters one has in their life is with their mother, and for me a goen is not word enough to express this bond. It is someone to whom I owe my life, literally and figuratively, and to whom I have the utmost gratitude. She stood for me, encouraged me to pursue my path, whatever path it may be, and gives me strength to go on despite all the difficulties the Path may have along the way. To this woman, all my love and gratitude is reserved, and I dedicated this conquest to her, may this make you proud and happy, and may you know that the later is my utmost desire. They say that the utmost desire of a parent is to see their offspring happy, but it is selfish of a child not to wish the same back for those who gave their most so that they could be where they are.

Next, it is not everyone that has the chance of meeting a Master, in the original and true meaning of the word, along their Path. Someone that will shine a light on their way, and act like a guiding beacon, the compass we need when we lose our path. My Master, to whom I have an infinite amount of respect, is Jorge Kishikawa Sensei, and to whom I have the utmost gratitude for showing me the Path of the old, and teaching me how to obtain the strength to pursue my dreams, and be happy. My deepest and more sincere Thank You. On this note, I can't forget the one that allowed Sensei's teachings to reach me, Senpai Ivan Santos, who sought to learn the teachings of the Samurai with Kishikawa Sensei, and decided to employ his strength in spreading this knowledge and employing the sword that spreads life, in abundance. My most sincere thanks. And to my brothers in arms, and all those from Niten Institute, who follow this path with me and help me learn and grow as a human being at every instant , Thank You.

Our genes still play a great part in our decisions (together with our gut microbiota, but that's a story for another time), and we partake most of our lives with those who partake of the same genes than us, our family. To those we owe most of the teachings and examples we have early in life, those that help define who we will be later on, and this is of much value.

But family isn't just those we are bond by blood, but also those we are bond by soul. Many people we meet in life go beyond what chemistry and biology can explain, and those people are very important for us in life, they make us company through most of our Path, even if they are not there from the beginning, sometimes they can be there until the end. For those, there are many reasons I have to be thankful for, for putting up with me during the hard times, the breakdowns. For lifting me up and helping me keep strong in the Path, not losing my ways to myself and my mind. To showing me amazing things in life, memories I'll never forget, moments of happiness and anguish. For being there when I needed, for the discussions in Physics, science, chemistry, computation. For the late night studies in the library, formulation of theories, works, simulations of zombie outbreaks, other cities kidnaps and RPG times. Thank you very much for being part of my life, and may you stay longer in this Path with me.

To my supervisors, Fernando Sato and Hiroshi Kobayashi, who accepted promptly to advise me in this research, even though it was a sudden and somewhat selfish request, believed in my potential and guided me through this portion of the Path into another milestone, one that will mark my life and career for long into the future. In special, to Hiroshi Kobayashi who, even being at the other side of the world, took in the effort to guide me through the studying of Astrophysics and the computational methods applied in such studies, and taught me all the necessary to reach this goal. My most sincere thanks.

For all those that crossed my Path, may you still be here or not, for you left a mark, and impression on the way, and however small it may have seemed, it represents a

learning that will be with me until the end of my days.

And last, but very important, to the funding agencies, CAPES, Cnpq, Fapemig, Finep, UFJF and the Physics Department from UFJF, without whom I wouldn't be able to pursue my research and achieve this conquest and milestone.

And as Kishikawa Sensei says, life is a war and after each battle won, one must prepares for the next. May the next combat come!

“Persuade thyself that imperfection and inconvenience are the natural lot of mortals, and there will be no room for discontent, nor despair.”

Tokugawa Ieyasu

Abstract

In the past century, computational methods have been being applied more and more to physical systems, in special to systems which are chaotic in nature or don't have an analytical solution, or both, such as is the case for systems that obey the N-body problem. To solve such systems, it is necessary to select the most suitable numerical method, one that takes into account both the necessary time and computational resources available to the researcher, and in order to be able to do so, one must have a good set of tools available. In this work we present a numerical method known as the Hermite Scheme, a fourth-order predictor-corrector integrator which makes use of an Individual Time Step structure, making it capable of processing multi-scale systems. We test its accuracy and study its applicability to the N-body problem, extending the result to chaotic systems in general. We then proceed to check its performance for a N-body system, and compare it to the performance of another fourth-order integrator, the Runge-Kutta. Lastly we verify its performance to multi-scale systems by reproducing some real-life results. Our results show that the Hermite Scheme has a good applicability to N-body systems, with an overall performance better than the fourth order Runge-Kutta. It also shows a good performance when applied to multi-scale systems, with no harm to its overall time performance, which can be applied to other multi-scale systems other than the N-body problem. With this verification, we intend to further apply this method to collision processes and apply the final result on the study of planet formation. The method also shows great potential applicability to Condensed Matter Physics, and we intend to test-apply to known systems in the future.

Keywords: Computational Physics, Hermite Scheme, N-body, Condensed Matter Physics

Resumo

No último século, métodos computacionais vem sendo aplicados mais e mais a problemas físicos, em especial àqueles que ou são caóticos ou não possuem solução analítica, ou ambos, como é o caso de sistemas que obedecem ao problema de N-corpos. Para resolver tais problemas, é necessário selecionar o método numérico mais adequado, um que leve em consideração ambos o tempo necessário e os recursos computacionais disponíveis ao pesquisador responsável; e para que ele seja capaz de fazê-lo, é necessário que ele tenha uma ampla gama de ferramentas disponíveis. Neste trabalho, mostraremos um método numérico conhecido como o Esquema de Hermite, um integrador de quarta ordem preditor-corretor que faz uso de uma estrutura de Passo de Tempo Individual, tornando-o capaz de processar sistemas em multiescalas. Nós testamos sua precisão e estudamos sua aplicação ao problema de N-corpos, estendendo o resultado a sistemas caóticos em geral. Em seguida, verificamos seu desempenho para um sistema de N-corpos e comparamos o resultado com o desempenho de outro integrador de quarta ordem, o Runge-Kutta. Por último nós reproduzimos resultados reais para verificamos seu desempenho em sistemas multiescala. Nossos resultados mostram que o Esquema de Hermite possui uma boa aplicabilidade para sistemas de N-corpos, com um desempenho ao todo melhor do que um Runge-Kutta de quarta ordem. Ele também apresenta um bom desempenho quando aplicado a sistemas multiescala, com nenhum prejuízo à sua performance temporal total, demonstrando que pode ser aplicado a sistemas multiescala que não somente o problema de N-corpos. Com estas verificações, pretendemos no futuro aplicar este método a sistemas com processos de colisão, e aplicar o resultado final no estudo de formação planetária. O método também apresenta grande potencial para aplicação em sistemas de Física da Matéria Condensada, nos quais pretendemos testar a aplicação do método em sistemas conhecidos no futuro.

Palavras-Chave: Física Computacional, Esquema de Hermite, N-corpos, Matéria Condensada

List of Figures

Contents

Complementary Literature

1. NITADORI, K.; MAKINO, J. Sixth-and eighth-order Hermite integrator for N-body simulations. New Astronomy, Elsevier, v. 13, n. 7, p. 498–507, 2008. [1]

2. MAKING, J. Comparison of two different tree algorithms. Journal of Computational Physics, Elsevier, v. 88, n. 2, p. 393–408, 1990. [2]

3. MAKINO, J. A modified Aarseth code for grape and vector processors. Publications of the Astronomical Society of Japan, v. 43, p. 859–876, 1991. [3]

4. MURRAY, C. D.; DERMOTT, S. F. Solar System Dynamics. [S.l.]: Cambridge University Press, 2000. [4]

5. PRESS, W. H. et al. Numerical recipes 3rd edition: The art of scientific computing.[S.l.]: Cambridge university press, 2007. [5]

6. DEVANEY, R. An introduction to chaotic dynamical systems. [S.l.]: CRC Press, 2018. [6]

7. DANFORTH, C. M. Chaos in an atmosphere hanging on a wall. Mathematics of Planet Earth, v. 17, 2013. [7]

Introduction

The past century was a very turbulent period for the planet. The world was devastated by many different wars, of which the two World Wars left their scars, showing us the worst part of humanity. It was the beginning of the nuclear crisis, among two major economical crises that shook and changed the social structure of the world. Even among this seeming chaos, the sciences in general thrived and saw a period of extremely rapid growth, with many important discoveries and developments that shook the bases of our understandings of the universe. Physics saw its biggest revolution since Isaac Newton published his work in the 17th century, with the appearance of two theories that tested our capacity of acceptance, and changed completely the way which we observe the universe around us: General Relativity [8] and Quantum Mechanics [9]. Those two revolutionary theories opened the doors to a series of new areas and phenomena, many of which still puzzle physicists nowadays. The development of new equipment for both observation and experimentation allowed the growth of many areas, among which we can cite Astrophysics, Condensed Matter Physics and Optics.

But another area also saw an astounding growth: Computer Science. The world saw an increase on the number of computers, with the development new hardware and software, and they started becoming part of the ordinary everyday life, until it became an inseparable part of today's society. Computers started to be developed not only for everyday uses or industrial applications, but the academia also saw a great opportunity, and scientists joined the challenge to apply those computational methods to develop fundamental research. This lead to many discoveries in all areas of scientific knowledge, and contributed to the creation of many new areas, among them Computational Physics. Those called computational physicists concern themselves with the study and development of numerical methods which can be applied to solve physical problems, and allow us to find solution which otherwise would take us years, decades or even centuries to solve. These methods started being applied to many different areas of Physics, from experimental, to observational up to theoretical Physics. The only limitations were that imposed by the technology limitations of the given period.

However,in order to overcome these limitations of the hardware, more efficient numerical methods were being developed, allowing for the solution of new and more complex systems by using computer simulations. Nowadays computer simulations are a fundamental part of the scientific development, with many areas highly dependent on them for the production of results. Computational methods gained a lot of space within Physics, with simulations being applied to a series of problems in many areas, such as in the study and synthesis of materials [10], the understanding and prediction of the weather

[11, 12, 13], fluid mechanics and its role in the human body [14, 15] and so on. Many of the solutions offered by numerical methods are that of system whose analytical solutions are very hard, or even impossible, to obtain. This is the case of many systems in Condensed Matter, and the N-body system.

The N-body problem was first proposed by Isaac Newton in the 17th century, refers to a system of N bodies interacting with one-another through his Law of Universal Gravitation. It was discovered by Poincaré that, for $N \geq 3$, the N-body problem is a chaotic system [16, 17], and its known analytical solutions are in the form of power series with very slow convergence [18, 19]. Albeit classical, it provides a good enough representation of the behavior of a great number of structures and systems in the universe; e.g, the orbital dynamics of planets or planet formation, and the dynamics of comets and asteroids. The dynamics of the N-body problem weren't well understood until the end of the 20th century, when numerical methods started to be applied in the study that obeyed its conditions [20], and since then it has thrived, with many new information being obtained from those simulations. In Modern days, a lot of systems are characterized as N-body systems. As an example it can be cited an atomistic N-body system is one in which N particles interact with each other through a mutual force that act at a distance. This force sometimes have a similar formulation as that of the celestial N-body problem, but nonetheless remains a chaotic system which relies on numerical methods for a solution. Numerical methods are now the preferred when looking for solutions to those types of systems. They became also the main tool used in the study and classification of new materials, or in the study of the comprehension of the molecular properties of materials, as those methods represented a faster, more economic way of reaching a result and understanding the feasibility of a given material.

However, as much as computer simulations contribute to scientific development, it possess its limitations. When working with a numerical simulation, one needs to take into account not only the characteristics and necessities of the physical system to be studied, but they also need to consider the availability of computational resources and human time necessary for such simulations to take place. When opting to use a simulation to solve a certain problem, it is important the researcher ponders carefully on which is the most suitable method available for the given problem. One which can balance both the exigence on accuracy and efficiency for the physical system being studied, but as well allows the simulation to be performed in the computational resources they have available in a viable amount of time. In order to guarantee that they are using the most suitable method, it is necessary that one have a sufficient knowledge of the available methods, that is, that he has enough tool with which to work. There are many famous such tools, that are widely used in all areas of physics; e.g., the Runge-Kutta methods, Velocity Verlet, Euler method. However, there are still many tools that are highly used in a given area or field, with potential applications to other fields but it is not widely spread. In this work, our aim is to present and discuss one of those methods, called the Hermite Scheme, which allows one to simulate physical systems with a good accuracy, as well as being useful when dealing with multi-scale systems.

The Hermite Scheme was developed in 1991 by Junichiro Makino as a means for solving N-body simulations [21]. We wish to study the method and its application to the N-body problem. We perform tests concerning its overall accuracy to analyse its applicability to chaotic systems, as well as a verification of its performance when compared to methods of similar accuracy, that are commonly known. We finish our study testing the capacity of reproduction of known results from the literature by using the method. During the work, we analyse its potential applications to problems in Condensed Matter Physics, and area in which N-body simulations are extensively used, and which could benefit from the performance and features of the Hermite Scheme. This work was born from a collaboration between the Computational Simulation Laboratory from the Universidade Federal de Juiz de Fora, and professor Hiroshi Kobayashi, from the Theoretical Astrophysics Laboratory from Nagoya University. Its final aim is to study the formation of planets of our solar system through the use of computational simulations that use the Hermite Scheme as the integrator, and to further investigate its potential applicability to systems in Condensed Matter Physics.

Part I

Methodology

1 Fourth-Order Runge-Kutta Method

When a physicist creates a numerical method to be applied in Physics, they usually do focusing on solving the problem at hand, save exceptions. When faced with a particular differential equation in his studies of atomic spectra, Carl Runge sought a way to extend Euler's method for solving Ordinary Differential Equations (ODE) and in 1895 he published a paper proposing three different schemes [22], which later would come to be known as Runge-Kutta methods. The method was later improved by Martin Kutta in a paper in 1901, who completed the method's name, who formulated the widely used fourth-order formulation, and attempted at a fifth order scheme [23].

The Runge-Kutta methods have been since thoroughly studied and developed, with contributions from both mathematicians and physicists in a similar way. Currently the most used method is the fourth-order Runge-Kutta, but there are also lower orders, from the second order, up to tenth order algorithms [24, 22]. In recent years, implicit and adaptive time-step methods have also been developed for the Runge-Kutta methods, which increase the applicability and performance of those methods. Overall, when one thinks about the numerical methods used in solving dynamical equations for systems in Physics, few methods are as famous as the Runge-Kutta. Many high-end softwares like the Mathematica and MatLab use the Runge-Kutta methods as the main integrator for the solution of ODEs.

However, the Runge-Kutta methods are excessively robust, and can consume a great amount of computational power, specially when dealing with N-body simulations, a case in which many other numerical methods fare better. However, given its applicability in many physical systems, one cannot talk about numerical methods in Physics without mentioning it. In this chapter, we present a brief introduction of the most used fourth-order formulation of the Runge-Kutta methods, and show a small development of the method for its application to second order ODEs and to N-body simulations.

1.1 Fourth-order Formulation

When Carl Runge published his paper on the Runge-Kutta methods, he desired to solve an initial value problem of the type

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

As canbe seen the Runge-Kutta method's original formulations were made for the solution of first order ODEs. He first proposed three second order methods, based on the midpoint rule applied to differential equations and different forms of the trapezoidal rule [22]. The fourth-order method is based on the trapezoidal rule quadrature formula, and uses a 4-stages calculation. The Runge-Kutta fourth-order method is a fixed time-step integrator, and throughout this chapter we shall consider the time step to be called h. Let y_n be the value of the function $y(t)$ at the time t_n. We wish to obtain the value of $y(t)$ at a time t_{n+1}. In order to do so, the fourth-order Runge-Kutta method makes use of four intermediary steps, which are given by

$$k_1 = hf(y_n, t_n) \tag{1.1}$$

$$k_2 = hf(y_n + \frac{k_1}{2}, t_n + \frac{h}{2}) \tag{1.2}$$

$$k_3 = hf(y_n + \frac{k_2}{2}, t_n + \frac{h}{2}) \tag{1.3}$$

$$k_4 = hf(y_n + k_3, t_n + h) \tag{1.4}$$

those values are then replaced in the trapezoidal rule's quadrature formula

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{1.5}$$

in which the coefficients of the k_i depend on the approximation formula for the quadrature. A more concise form of obtaining these coefficients can be found on J.C.Butcher's book [23]. The latter equation gives the value of the function $y(t)$ at the time t_{n+1}. The entire process needs only 5 main steps, being relatively simple to formulate for scalar systems. For vector systems, the adaptation is straightforward.

1.2 Second Order Differential Equations and Coupled Equations

As we saw, the Runge-Kutta methods were initially proposed for first-order ordinary differential equations, but adaptations to higher order differential equations are straightforward. The first such adaptation for second order differential equations was by Nyström in 1925, in a simple method given as follows.

Considering a second order differential equation of the form

$$\frac{d^2x}{dt^2} = f(x, t) \tag{1.6}$$

in which we wish to obtain $x(t)$. To do so, we observe that equation 1.6 can be written on the format

$$\frac{d}{dt}\left(\frac{dx}{dt}\right) = f(x,t) \tag{1.7}$$

From which we define the function $v(x,t)$ by

$$\frac{dx}{dt} = v(x,t) \tag{1.8}$$

By replacing equation 1.8 in equation 1.7, we arrive at the relation

$$\frac{d^2x}{dt^2} = \frac{dv}{dt} \tag{1.9}$$

From which we find that

$$\frac{dv}{dt} = f(x,t) \tag{1.10}$$

I other words, to apply the Runge-Kutta methods to second order differential equations of the form 1.6, one needs to instead solve the system of coupled two first order differential equations

$$\begin{cases} \frac{dx}{dt} = v(x,t) \\ \frac{dv}{dt} = f(x,t) \end{cases} \tag{1.11}$$

for which one needs the initial conditions $x_0 = X(t_0)$ and $v_0 = v(t_0)$. Solving this kind of system follows the same procedure as solving a general system of coupled first-order differential equations using the method. Given one such system

$$\begin{cases} \frac{dx}{dt} = g(x,y,t) \\ \frac{dy}{dt} = f(x,y,t) \end{cases} \tag{1.12}$$

with the initial conditions y_0 and x_0, one needs to apply the procedure from last section to both equations at the same step of the algorithm, as it must update all variables at the same time. That is, the procedure must be carried in the order

$$\begin{aligned} k_{x1} &= hg(x_n, y_n, t_n) \\ k_{y1} &= hf(x_n, y_n, t_n) \end{aligned} \tag{1.13}$$

$$
\begin{aligned}
k_{x2} &= hg(x_n + \frac{k_{x1}}{2}, y_n + \frac{k_{y1}}{2}, t_n + \frac{h}{2}) \\
k_{y2} &= hf(x_n + \frac{k_{x1}}{2}, y_n + \frac{k_{y1}}{2}, t_n + \frac{h}{2})
\end{aligned}
\tag{1.14}
$$

$$
\begin{aligned}
k_{x3} &= hg(x_n + \frac{k_{x2}}{2}, y_n + \frac{k_{y2}}{2}, t_n + \frac{h}{2}) \\
k_{y3} &= hf(x_n + \frac{k_{x2}}{2}, y_n + \frac{k_{y2}}{2}, t_n + \frac{h}{2})
\end{aligned}
\tag{1.15}
$$

$$
\begin{aligned}
k_{x4} &= hg(x_n + k_{x3}, y_n + k_{y3}, t_n + h) \\
k_{y4} &= hf(x_n + k_{x3}, y_n + k_{y3}, t_n + h)
\end{aligned}
\tag{1.16}
$$

$$
\begin{aligned}
x_{n+1} &= x_n + \frac{1}{6}(k_{x1} + 2k_{x2} + 2k_{x3} + k_{x4}) \\
y_{n+1} &= y_n + \frac{1}{6}(k_{y1} + 2k_{y2} + 2k_{y3} + k_{y4})
\end{aligned}
\tag{1.17}
$$

in which the lower indexes x and y represent that quantity calculated for the functions x and y respectively. For a second order differential equation, one just replaces in the above procedure $g(x, y, t) \rightarrow y(x, t)$ and $y(x, t) \rightarrow v(x, t)$. By formulating the method for second order differential equations, one is now capable of applying the Runge-Kutta method to solve the dynamical equations for physical systems, in which x is the position and v is the velocity of the particle being studied. Again, the generalization to the vectorial case is straightforward.

1.3 N-body Case

A N-body system of particle is composed of a system of N vectorial dynamic equations, which resume into $3N$ coupled second order differential equations to be solved. We use the method from the previous section to split these into $6N$ coupled first order differential equations

$$
\begin{aligned}
\frac{d\mathbf{r}_1}{dt} &= \mathbf{v}_1(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N, t) \\
\frac{d\mathbf{r}_2}{dt} &= \mathbf{v}_2(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N, t) \\
&\vdots \\
\frac{d\mathbf{r}_N}{dt} &= \mathbf{v}_N(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N, t)
\end{aligned}
\tag{1.18}
$$

$$\begin{aligned}
\frac{d\mathbf{v}_1}{dt} &= \mathbf{a}_1(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N, t) \\
\frac{d\mathbf{v}_2}{dt} &= \mathbf{a}_2(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N, t) \\
&\vdots \\
\frac{d\mathbf{v}_N}{dt} &= \mathbf{a}_N(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N, t)
\end{aligned} \tag{1.19}$$

in which the $\mathbf{r}_i$, $\mathbf{v}_i$ and $\mathbf{a}_i$ are the position, velocity and acceleration of particle i, respectively. We need $6N$ initial conditions, and solve using the same procedure applied for the solution of coupled differential equations in the previous section.

This means that, for a N-body system, the fourth-order Runge-Kutta algorithm must perform $5 \times 6N$ calculations per step, which is an unreasonable number of calculations. This shows that while the Runge-Kutta is a good integrator for single-particle systems, it is not well adapted to N-body systems.

2 Hermite Scheme

For many areas in Science, experimentation is the main resource one turns to when one wishes to test models and theories about observations. However, some areas, specially in Physics, work with phenomena on such scales that it is impossible to perform experiments to verify certain models or theories, and therefore strongly rely on observations. This is the case of Astronomy and Astrophysics. This scenario changed with the advent of numerical and computational methods, which became the "laboratory" for the astrophysicist to experiment with his models and compare the results to all the observations available. Not only in computer simulation, but advances in image processing softwares contributed to a better quality and efficiency on processing the data obtained through all kinds of telescopes.

There is a large variety of systems that can be observed in the universe that have been studied by astrophysicists, but a great number of them behave like N-body systems, whose studies faced great difficulties given the mathematical limitations associated to the N-body problem. The first computer simulations of the N-body problem have begun in 1960 with von Hoerner [20], albeit the existence of numerical methods carried by hand for N-body problems with a small value of N weren't unheard of. The computational limitations of the period forced physicists to limit themselves to low N values, and many different numerical methods started being developed as an effort to reduce computational costs, and computational time, in order to allow more complete and efficient simulations.

Nowadays, many areas in the sciences rely on computational resources on their studies, with the goal to reduce cost, to increase the productivity or simply because they have no choice. It is imperative then that the researcher have access to a broad range of computational methods and resources that would allow them to simulate and work with evermore complicated systems. In this chapter we present a numerical method developed at the end of the 20th century for the use on N-body simulations, which incorporate a characteristic that makes the method useful even for systems beyond those it was first intended for.

2.1 Overview

In the end of the 20th century, efforts in developing new computational methods and resources for the applications in the sciences in general were at full speed. Computers started appearing everywhere, and more and more they were being used to process data and perform experiments about nature. Not only computer scientists, but physicists also started seeking computational methods to apply in solving the problems they couldn't solve by hand, or to increase the efficiency of already-existing methods to improve the productivity of the scientific community. In 1991, Junichiro Makino in an attempt to increase the performance and accuracy of the algorithms created by Sverre J. Aarseth for application in N-body simulations, proposed a method that made use of an Hermite polynomial interpolation for the obtaining of the solution, and relied on an independent time-step for each particle in the system, a feature that was proving itself indispensable for this kind of integration [21, 20]. The method would later come to be called the Hermite Scheme, it is a predictor-corrector type integrator. A predictor-corrector integrator is one that performs the integration of a given differential equation in two major steps

1. A "prediction" step, called the 'predictor'. Given the value of the function or functions at a given set of points, the algorithm predicts their values at a subsequent set of points.
2. A "correction" step, called the 'corrector'. The previous prediction is then corrected using another method, to a desired precision for the value of the functions at the same set of points.

On top of that, the method showed great accuracy, with the minimum accuracy it could achieve being fourth-order. Capable of solving any order differential equations of the format

$$\frac{d^n y}{dt^n} = f(y, t) \tag{2.1}$$

it requires a set initial conditions which depend directly on the order of the differential equation one wants to solve. It was developed for applications to problems in Physics, in special to N-body algorithms, and therefore its intended use was primarily in solving second order ordinary differential equations. For equations with this order, one needs as initial conditions the initial time t_0, and the values of the function $y_0 = y(t_0)$ and its first derivative (here considered a time-derivative for simplicity purposes) $\dot{y}_0 = \dot{y}(t_0)$ for all the particles one wish to integrate. Bear in mind that, when we say the initial time t_0 for each particle must be given as an initial condition, it isn't necessary that they be the same, and the initial conditions for the particles must be with respect to their initial time. This is a feature of individual time-step integrators, and differ from that observed on integrators such as the Runge-Kutta, on which the initial conditions for all the particles must be given at the same instant of time. Differently from the Runge-Kutta, the Hermite Scheme processes and integrates each particle individually, and has a different value of time-step for each one of them.

Another feature of the method is that, aside from inputting the function $f(y, t)$, it is necessary to calculate its time derivative $\dot{f}(y, t)$ by hand and input in the algorithm , for all the particles. The values of these quantities at time t_0 are calculated by the algorithm itself. Considering the integration of a N-body system using the Hermite Scheme considering dt_j to be the time-step associated to the j-th particle, the integration process can be summarized in the following set of steps:

a) Choose particle j with the smallest $t_j + dt_j$

b) Define the global time as $t = t_j + dt_j$

c) Calculation of the 'predictor' for all the particles

d) Calculation of $f(y,t)$ and $\dot{f}(y,t)$ for particle j at instant t, using the predictors

e) Calculation of $f^{(2)}(y,t)$ and $f^{(3)}(y,t)$ for particle j, using an Hermite Interpolation based on $f(y,t)$ and $\dot{f}(y,t)$

f) Calculation of the 'corrector' for particle j alone, using its predictor and both $f^{(2)}(y,t)$ and $f^{(3)}(y,t)$

g) Repeat processes 'd', 'e' and 'f' a sufficient amount of times

h) Update $t_j = t$

i) Calculation of the new time-step dt_j for particle j

j) Return to step 'a'

2.2 Individual Time-Step

When one stops to investigate physical phenomena, one striking difference that they find is the time scale which is needed for each to occur. This difference becomes the more prevalent when we look at extremes: some quantum phenomena take time intervals of the order picoseconds to occur, whereas astronomical events may take over millennia to happen entirely. This feature is very important for the computational physicist, and the differential equation one needs to solve depend on time, and the numerical solution depends on a finite number of points, the number of which depend on the physicist's choice of a time-step, which in turn depends on the time scale of the phenomenon being studied.

It is easy to understand this difference in scale when comparing different systems, but this feat is not always so easy when we are comparing phenomena within the same system. Many systems in the universe present processes and phenomena that are dependent on each other, but which occur at completely different time scales. Those systems are called multi-scale, and some examples are planetary systems with collisions, the movement of the subatomic particles in an atom, the weather, among others. If one stops to analyse, it would come to the conclusion that multi-scale systems might in fact be the rule in the universe.

Physicists are mostly interested in a given phenomenon within a system, and oftentimes neglect the interaction among multi-scale phenomena. However, these interactions can prove important in one's study, and the development of a proper way to process them is necessary for those cases. When one thinks in terms of computer simulations, most of the commonly known numerical methods make use of a single, fixed value time-step for all the system, which means fixing the scale at which perform the study. This forces the researcher to choose which phenomenon he wants to study, and precludes them from studying the effect that the interaction of these multi-scale phenomena have in the final dynamics of the system, or when they allow for this study, they do so at a great computational time and resource cost.

Nonetheless, there are algorithms that are capable of using different time-steps for different portions of the system it is studying, at different moments of the simulation. This allows the algorithm to process these multi-scale phenomena successfully when necessary, and allows the study of their mutual interactions on the final dynamics. It is even possible to allow for the time-step to regulate itself based on the intensity of the interactions between the elements of the system. Such auto-regulatory time-step procedures are called *Individual Time-Step* [25, 26] procedures, and they have become indispensable for N-body algorithms, as they collaborate greatly to the reduction in computational time for a simulation. Numerical methods which use individual time-step procedures usually integrates every particle or portion of the system individually, allowing for an increase in performance over algorithms that perform the integration of all the particles at the same time, in part due to the number of calculations per step becoming significantly smaller in the former when compared to the later, according to the number of particles or partitions in the system.

The Hermite Scheme integrator makes use of an individual time-step procedure, having a different time-step for each particle, and recalculating it after processing every particle individually. Since it integrates the particles one-by-one, it allows the user to offer as inputs the particles and their initial conditions at different moments of time, without harming the final result. The time-step for each particle is self-adaptive, and regulates according to the intensity of the interactions on the particle. It is not necessary for the user to select a time-step for the algorithm, but instead they choose an accuracy parameter η and the algorithm calculates the time-step of each particle. This accuracy parameter is important to guarantee that the values of the time-steps are compatible with the range of values for time-steps within which the integrator can function.

In order to guarantee that the time-step is coherent with the intensity of the interactions for each particle, we must calculate it such that it depends on the acceleration and its derivatives for said particle. At the beginning of the simulation, the only information we have available is the acceleration and its first time-derivatives. Then, to calculate the initial time-step we use the equation [25]

$$dt_j = \eta \frac{|\mathbf{a}_j|}{|\dot{\mathbf{a}}_j|} \tag{2.2}$$

some prefer to use another similar equation, based on general considerations on the convergence of the associated Taylor series [20]

$$dt_j = sqrt\left(\eta \frac{|\mathbf{a}_j|}{|\dot{\mathbf{a}}_j|}\right) \tag{2.3}$$

both of which which give a suitable accuracy for small enough η, given that the velocities of the particles aren't too small [25]. This initial choice is fine, since the algorithm evolves fast enough allowing for the calculation of a more suitable time-step. Observe that the dependence on the acceleration sets a connection of the time-step to the intensity of the physical interactions on the particle.

This is a simple formula, used due to our lack of data. As each particle is selected and integrated the first time, we acquire more information on the interactions of the particle, namely the higher order derivatives of the acceleration. We are then capable of employing a more robust formula for the time-step, based on experimentation with the method [20]

$$dt_j = \eta \sqrt{\frac{|\mathbf{a}_{1,j}||\mathbf{a}^{(2)}_{1,j}| + |\dot{\mathbf{a}}_{1,j}|^2}{|\dot{\mathbf{a}}_{1,j}||\mathbf{a}^{(3)}_{1,j}| + |\mathbf{a}^{(2)}_{1,j}|^2}} \tag{2.4}$$

this formula is more sensitive than the previous one, and ensures that all the derivatives of the acceleration play a role in the definition of the time-step. It is also well define for some special cases; e.g., $|\mathbf{a}| = 0$ [20, 26]. It also scales with the scale of the interactions faced by each particle, such that the particles that interact the most strongly will have shorter time-steps than those who interact weakly with its peers.

This adaptive time-step allows for a 'smarter' integration. When we use a fixed time-step integrator, all particles must be integrated, even if the scale of the time-step being used is much smaller than that of the phenomenon being faced by the particle. As for the adaptive time-step, each particle is processed just a sufficient amount of times, compatible with the phenomenon it is facing, thus saving on computational time and avoiding unwanted computational errors.

2.2.1 Cancellation Error

When performing numerical simulations, one must be careful with a handful of numerical errors that may arise, many of which present in every algorithm to some degree, and which escape the control of the person who programmed the code, namely the round-off and truncation errors. There is a more subtle error, which is mostly associated with a lack of caution on the programmer part as to the choice of a suitable time-step and the scale of the system being studied, which can be common in multi-scale systems. This error is called *Cancellation Error* or *Loss of Significance.*

A cancellation error is an error that occurs when the accuracy of the numerical calculation performed is smaller than the accuracy of the computational variable being used for a given quantity, rendering the computer unable to "see" the new result, and thus it keeps the old one. For example, a floating point variable has an accuracy of 10^{-8}, if a calculation would cause a floating point variable associated to the position of a particle to change by an amount, say 10^{-9}, then the computer would plainly ignore this change, "cancelling" the evolution of the given particle. For a physical system, it might mean that the increase in the position of a particle is not computed, and it will remain static, and therefore the ensuing result would have no physical meaning.

When dealing with multiple scale systems, when one wishes to observe the fastest-paced phenomenon and they are using a fixed time-step algorithm, then he must use a time-step compatible with the scale of the desired phenomenon, but since all particles are integrated together, this could imply an impediment to the integration of the slowest-paced particle. Therefore, when using fixed time-step algorithms, one must consider carefully which time-step to choose, in order not to harm the physical nature of the system. Nonetheless, by processing the particles individually, and using an individual time-step, one eliminates the possibility of a cancellation error to occur, as the time-step of the given particle will be balanced with its physical interactions.

We call to the attention of the reader that this isn't the only adaptive time-step method in existence. Many methods allow for fixed block time-steps, which processes parts of the system at pre-determined values of time-steps, such as the Ahmad-Cohen procedure [25], which would prove useful for closely packed systems in Condensed Matter Physics, for example. Other methods integrates all the particles at once, but the time-step is calculated by the algorithm, and the smallest is chosen. We recommend a reading of the literature when choosing a suitable method for your application.

2.3 The Algorithm

Now that we have set the foundation, we proceed to a more in-depth explanation of the Hermite Scheme integration method. Given a N-body simulation, the position $(\mathbf{x}_j)$, velocity ($\mathbf{v}_j$) and initial time (t_j) of each individual particle must be supplied as an input. With these data, the acceleration ($\mathbf{a}_{0,j}$) and its first time-derivative ($\dot{\mathbf{a}}_{0,j}$) at time t_j is calculated for all of the particles in the system.We then proceed to the calculation of the initial time-step of the particles, using either formula 2.2 or formula 2.3. We then perform the following steps before beginning the integration procedure

1. Calculate the time $t_i + dt_i$ for all of the particles, and sort the values from smallest to biggest
2. Select particle j with the smallest $t_j + dt_j$; i.e., the most delayed particle, to be the one to be integrated
3. Define the global time as $t = t_j + dt_j$

this procedure ensures that the algorithm will always choose the most delayed in time, and most strongly interacting particle to integrate. This grants the proper time evolution of the physical system, avoiding a particle being left behind, or that the same particle is the only one to be integrated.

However, remember that when we inserted the inputs on the algorithm, we didn't need to guarantee that all the particles were situated at the same instant in time. In physical terms, since we are considering a classical N-body simulation, we have a huge problem. In Classical Mechanics, the long-distance interactions between particles are instantaneous, and thus in order to study the dynamics of the system we must ensure that all the particles are at the same moment in time.

2.3.1 Predictor

We can't proceed the integration before all the particles are at the same moment in time. In order to do so, the first step of the integration is the calculation of that which we call the *predictor* for all the particles. The predictor is a prediction of the position and velocities of all the particles in the system at the global time t. This prediction is performed by considering a Taylor Series expansion of both the position and velocity for all the particles, each around their initial time t_i up to third order for the position, and second order for the velocity

$$\mathbf{x}_{p,i} = \mathbf{x}_{0,i} + \Delta t_i \mathbf{v}_{0,i} + \frac{\Delta t_i^2}{2!}\mathbf{a}_{0,i} + \frac{\Delta t_i^3}{3!}\dot{\mathbf{a}}_{0,i} \tag{2.5}$$

$$\mathbf{v}_{p,i} = \mathbf{v}_{0,i} + \Delta t_i \mathbf{a}_{0,i} + \frac{\Delta t_i^2}{2!}\dot{\mathbf{a}}_{0,i} \tag{2.6}$$

In this expansion, the time-steps Δt_i are not the ones calculated before, but are given by the expression

$$\Delta t_i = t - t_i \tag{2.7}$$

by using this expression to calculate the time-step of the series, the only ensuing positive time-step will be that of the selected particle j, while all the other Δt_i will have a negative value. In terms of the prediction, while the selected particle advances in time, all the remaining particles are "moved" back in time, being placed at the positions they were at the global time t, allowing for a study of the dynamics of the system, since now all the particles are at the same instant of time, and thus instantaneous interactions are possible.

Notice that at this point, the overall accuracy order of the predictor is already second order in dt_i. In the above expansion, the predictor of the velocity is second order in dt_i, but that of position is third order. We do this because, in order to make the predictor of the velocities be second order, we need $\dot{\mathbf{a}}_{0,i}$, and since we already have it calculated, we can insert it into the predictor of the position without harming the algorithm. It is also important to point that in the above calculation, none of the initial $\mathbf{x}_{0,i}$ and $\mathbf{v}_{0,i}$ values are replaced by their respective predictors, since we are effectively integrating only a single particle's trajectory.

Using the predictors, we calculate the acceleration $\mathbf{a}_{1,j}$ and its first time-derivative $\dot{\mathbf{a}}_{1,j}$ at the global time t for particle j alone.

2.3.2 Obtaining $\mathbf{a}^{(2)}_{0,j}$ and $\mathbf{a}^{(3)}_{0,j}$

To perform the correction of the predictor, we need higher order derivatives of the acceleration, more specifically, we need its second and third order time-derivatives. This time, we do not need to provide the formulas derived by hand, but instead we will make use of a Hermite polynomial interpolation based on $\mathbf{a}$ and $\dot{\mathbf{a}}$, at both times t_j and $t = t_j + dt_j$, for particle j alone.

In this case, the Hermite interpolation polynomial for $\mathbf{a}$ and $\dot{\mathbf{a}}$ is equivalent to the Taylor Series expansion of both quantities up to $\mathbf{a}^{(3)}$, which are given by

$$\mathbf{a}_{1,j} = \mathbf{a}_{0,j} + dt_j\dot{\mathbf{a}}_{0,j} + \frac{dt_j^2}{2!}\mathbf{a}^{(2)}_{0,j} + \frac{dt_j^3}{3!}\mathbf{a}^{(3)}_{0,j} \tag{2.8}$$

$$\dot{\mathbf{a}}_{1,j} = \dot{\mathbf{a}}_{0,j} + dt_j\mathbf{a}^{(2)}_{0,j} + \frac{dt_j^2}{2!}\mathbf{a}^{(3)}_{0,j} \tag{2.9}$$

Solving these equations for $\mathbf{a}_{0,j}^{(2)}$ and $\mathbf{a}_{0,j}^{(3)}$, we obtain the results

$$\mathbf{a}_{0,j}^{(2)} = \frac{-6(\mathbf{a}_{0,j} - \mathbf{a}_{1,j}) - dt_j(\dot{\mathbf{a}}_{0,j} + 2\dot{\mathbf{a}}_{1,j})}{dt_j^2} \tag{2.10}$$

$$\mathbf{a}_{0,j}^{(3)} = \frac{12(\mathbf{a}_{0,j} - \mathbf{a}_{1,j}) + 6dt_j(\dot{\mathbf{a}}_{0,j} + \dot{\mathbf{a}}_{1,j})}{dt_j^3} \tag{2.11}$$

2.3.3 Corrector

We have shown that if we choose to stop the integration process here, we would already have a second-order accuracy. But in order to increase the accuracy, we need to perform a correction of the predictor we calculated previously for particle j. We call this correction the *corrector*. It is basically a continuation of the Taylor Series we made for the predictor, in which we add the next two terms in each expansion, making use of boith $\mathbf{a}_{0,j}^{(2)}$ and $\mathbf{a}_{0,j}^{(3)}$

$$\mathbf{x}_{c,j}(t_j + dt_j) = \mathbf{x}_{p,j} + \frac{dt_j^4}{4!}\mathbf{a}_{0,j}^{(2)} + \frac{dt_j^5}{5!}\mathbf{a}_{0,j}^{(3)} \tag{2.12}$$

$$\mathbf{v}_{c,j}(t_j + dt_j) = \mathbf{v}_{p,j} + \frac{dt_j^3}{3!}\mathbf{a}_{0,j}^{(2)} + \frac{dt_j^4}{4!}\mathbf{a}_{0,j}^{(3)} \tag{2.13}$$

since the $\mathbf{a}_{0,j}^{(2)}$ and $\mathbf{a}_{0,j}^{(3)}$ are order 0, the corrector for the position becomes order 5, and for the velocity order 4, achieving the desired overall order of 4 for the integrator. Again, we perform the one order higher correction for the position for convenience, since no extra work is necessary and we already have all the tools.

Here the solution has a fourth-order accuracy, which is that desired when applying numerical methods to the N-body problem, and if one wishes they can stop here. However, for some systems, specially those chaotic in nature, it is desired that the results be as accurate as possible. One can increase the accuracy of the result obtained this far by taking the following extra steps

1. Use the corrector to calculate again $\mathbf{a}_{1,j}$ and $\dot{\mathbf{a}}_{1,j}$
2. Use the new values to calculate $\mathbf{a}_{0,j}^{(2)}$ and $\mathbf{a}_{0,j}^{(3)}$
3. Replace these new values in the equation for corrector

and repeating them a sufficient amount of times. It is important that the value of the predictor remains unchanged during the whole process. At the last step, we recalculate the acceleration of the particle using the last obtained corrector. Bear in mind that, by choosing to do this, the total number of calculations are being increased with each iteration, and thus increasing the computational time of the simulation, so caution is advised when choosing a number of iterations in the correction process as to repeat just the strictly necessary amount of times.

We need to prepare for the next integration step. First we set the new initial conditions of particle j as the last calculated corrector

$$\begin{aligned} t_j &= t \\ \mathbf{x}_{0,j} &= \mathbf{x}_{c,j} \\ \mathbf{v}_{0,j} &= \mathbf{v}_{c,j} \end{aligned}$$

then we need to calculate the particle's new time-step. For this, we need the values of $\mathbf{a}^{(2)}$ and $\mathbf{a}^{(3)}$ at time t. Since we are using a third-order interpolation, given that $\mathbf{a}^{(3)}_{0,j}$ is already third order, and $\mathbf{a}^{(2)}_{1,j}$ is second order, at time t they will be

$$\mathbf{a}^{(3)}_{1,j} = \mathbf{a}^{(3)}_{0,j} \tag{2.14}$$

$$\mathbf{a}^{(2)}_{1,j} = \mathbf{a}^{(2)}_{0,j} + dt_j \mathbf{a}^{(3)}_{0,j} \tag{2.15}$$

we replace these values, together with $\mathbf{a}_{1,j}$ and $\dot{\mathbf{a}}_{1,j}$ in equation 2.4, and calculate its new time-step. At last, we calculate the new $t_j + dt_j$, reorder all the values, and repeat the entire process until necessary.

3 N-Body Systems

3.1 The Original N-Body Problem

The Renaissance was a thriving period for Astronomy, with many advances made in both observational and theoretical aspects being made by various scientists. Tycho Brahe would perform a series of intricate observations of the movements of the bodies in the night sky, specially that of the planets in the Solar System, and would leave his annotations to his assistant, Johannes Kepler, who would then derive from them his three laws governing the movements of the planets. The period would also see Galileo Galilei propose its heliocentric model, and his discoveries of four moons of Jupiter, Io, Callisto, Europa and Ganymede. But it was in 1687 that the epoch would see its greatest revolution in the sciences, as Issac Newton published his Philosophiæ Naturalis Principia Mathematica, in which he presented the world with his three Laws of Motion, that gave birth to what is known today as Classical Physics.

Newton also proposed a revolutionary force to explain the interaction and movement of the celestial bodies. That force would not require the bodies to be in direct contact with each other, but that would act among them from a distance, that would come to be known as the Law of Universal Gravitation. Given two isolated bodies of masses M_1 and M_2, respectively, separated by a distance r, they will interact through an attractive force whose modulus is given by the inverse square law

$$F = \frac{GM_1M_2}{r^2} \tag{3.1}$$

in which G is an universal constant, whose value is $G = 6.67 \times 10^{-11}\frac{Nm^2}{kg^2}$. He was able to derive the elliptical motion of a system of two bodies interacting through this force, and derived Kepler's Third Law of planetary motion with success. He would then take the next step in his study of gravitation, using his Law of Universal Gravitation Newton would try to derive the motion of the Moon under the gravitational influence of both the Earth and the Sun, the first attempt at a 3-body problem, but this would be his nemesis, as we wasn't able to find a complete solution.

The general extension to the 3-body problem problem was then natural. Given N bodies with masses m_i and given initial positions $\mathbf{r}_i$ and velocities $\mathbf{v}_i$, for $i = 1, 2, ..., N$ with respect to a given inertial frame, such that $\mathbf{r}_i \neq \mathbf{r}_j$, for $i \neq j$, interacting with each other via Newton's Law of Universal Gravitation, the total force on the i-th particle is given by the relation

$$\mathbf{F}_i = -\sum_{j\neq i}^{N} G M_i M_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} \tag{3.2}$$

Given this setting, what would be the resulting motion of all the bodies?

This system went on to be known as the N-body problem, and would elude physicists and mathematicians alike for centuries. The result of the 2-body problem was completely characterized by Johann Bernoulli in 1701 [16], but the solution for $N \geq 3$ would take scientists many years of work still to come.

3.1.1 Chaos

Solving the N-body problem would revolutionize the understanding of celestial mechanics, and the search for a solution intrigued scientists and sympathizers alike, so much that in 1889 it was proposed a prize, to be awarded at King's Oscar II of Sweden and Norway's 60th birthday, to that who could present a power series solution to the N-body problem, known as King Oscar's Prize [16]. Unfortunately for King Oscar, none of the participants were capable of finding the desired solution, but still there was a winner. The prize would go to Henri Poincaré, for his work on the restricted 3-body problem, and for showing that a N-body system with $N \geq 3$ is a chaotic system, a result he came to publish later [17].

A chaotic system is one which is highly sensitive on its initial conditions, that is, if you change the initial conditions ever so slightly, this represents a completely change in the final result. In the words of Edward Lorenz, 'chaos is when the present determines the future, but the approximate present does not approximately determines the future'. This is the reason why the N-body problem sometimes is regarded by some people as not possessing a solution, which we'll shortly see otherwise.

3.1.2 Series Solutions

The search for a power series solution for the N-body problem raged on, and it was not long after the handing of King Oscar's prize to Poincaré that a power series solution to the 3-body problem has been found. In 1913, Karl Sundmann published in the *Acta Mathematica* a paper containing a review of his works on the 3-body problem, in which he presented a converging series solution in terms of $t^{1/3}$, given that a collision does not take place within the system [19]. His method, however, was not applicable for values of N greater than 3 [16]. The global solution would appear only almost 80 years later, in the end of the 20th century. A chinese student by the name Quidong Wang published a convergent power series solution to the global N-body problem [18], regarded that no collisions take place.

Notice, however, that althought both Sundmann and Quidong Wang have provided a solution, that is all they are. They solve the mathematical initial value problem, but their convergence are so slow (Wang's solutions starts to converge at the 10^{126}-th term) that they are completely non-practical in terms of studying the behavior of celestial N-body systems. That is where numerical methods come into action, as they show very acceptable results, with a much faster convergence and accuracy. Modern astrophysical studies rely greatly in the use of computer simulation in the attaining of solutions to numerical models, which show a great compatibility to the vast number of observations available.

3.2 Other Formulations

Newton's N-body problem can be regarded as the first time the term has been mentioned in the field. However, as Physics developed, many other types of forces were discovered, with characteristics similar to that of classical gravitation, which also depend on a mutual interaction between two or more particles; e.g., Coulomb's force. The definition of a N-body system was then broadened to any system composed of N-bodies mutually interacting with each other, and the number of such systems that can be found in nature is astounding. However, not all of these systems present the same difficulty as that placed by Newton's N-body problem, however most of them remain being chaotic. We can cite as examples of N-body systems a group of charged point particles, particles in a gas or in a molecule or even spins in a bi-dimensional arrangement.

Given the different characteristics of N-body systems and the forces that act on each one, different approaches to solutions than that of the N-body problem have been used, some even with an appreciable success. In general, physicist try to apply approximation methods in solving N-body systems analytically, which for a group of systems show a satisfactory result. An example of such methods is the Mean Field Theory [27], or Group Renormalization Theory [27], which is adaptable to spin systems. However, a general analytical solution is not available for a majority of these systems, and just as in the gravitational case, researchers rely mainly on the use of computational simulations in attaining their results, as they prove a faster and more efficient method.

Not only in the discrete aspects of Nature one can find N-body systems. When using a computer to describe a physical system, in many situations it is necessary to discretize a continuum system in order for the computer to be able to process the calculations, and many times as a means to save on computer time. These are called N-body simulations, as the bodies in the simulation can be representations of real elements, or just be used as a computational tool in the calculation. Whatever is the case, they are one of the main methods used in simulations of physical systems, and is the method to which we are most interested in applying in this work.

3.3 Central Body Systems

When we look at the variety of N-body systems in Nature, we see that many of them share a common pattern, one that usually is of great curiosity and interest for scientists in the respective area of study. As mentioned, N-body forces are those that depend on mutual interactions among two or more particles, and generally are proportional to a given intrinsic characteristic of those particles; e.g., mass. Using the mass in our example, we observe a large quantity of systems in the universe in which a single body possess most of the mass in the entire system, and the other particles seem to move about this central particle. These are called Central Body Systems, and examples of such systems are the Solar System and atoms in general, in which the central bodies are the Sun and the nucleus of the atom, respectively. Such systems are of great interest to researchers because they allow for a great simplification in its formulation. In a Central Body System, one usually is more interested in the orbits of the smaller bodies around the central one, rather than the movement of the entire system (sometimes the researcher is also interested on the later).

Therefore, when studying such systems, one can consider the central body to be static, and look exclusively at the movement of the remaining bodies. This approximation reduces the total number of equations by 1, and the remaining N-1 equations can be simplified by considering the movement in the reference frame of the central body. We are interested in the N-body Problem, so for a Central Body System, the general N-body equation 3.2 will become

$$\mathbf{a}_i = -GM_0\frac{\mathbf{r}_i}{|\mathbf{r}_i|^3} - \sum_{i\neq j} GM_j\frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} - \sum_j GM_j\frac{\mathbf{r}_j}{|\mathbf{r}_j|^3} \tag{3.3}$$

in which M_0 represents the mass of the central body, r_i the position of the i-th body in the reference frame of the central body and M_i the mass of the remaining $N-1$ bodies. In this equation, the first term represents the acceleration of the i-th body with respect to the attraction of the central body, the second term is the gravitational pull of all the other $N-2$ smaller bodies on the selected i-th body. The third term accounts for the fact that the reference frame of the central body is a non-inertial frame, which accelerates due to the gravitational pull of the other $N-1$ bodies; i.e., the third term is the acceleration of the reference frame.

If we were to solve the equation analytically, the above equation would be enough. However, as we have mentioned before, in solving N-body system's equation, it is way more efficient to use numerical simulations. When solving numerically a given differential equation, one generally wishes to solve it in the most general possible fashion, such as that the result can be applied to a series of systems, not just a particular one. Equation 3.3 is still highly dependent on the scale of the system being studied, that is, on the units of mass, position and on the units of G being used. Notice that, for the N-body problem, it isn't the actual value of the positions and velocities that are important, but instead theirs relative values to each other. Then, in order to transform the above equation into its more general formula, it suffices that we work with its dimensionless form. Representing the dimensionless quantities by a tilde, we define them as

$$\tilde{\mathbf{a}}_i = \frac{\mathbf{a}_i}{a_0} \tag{3.4}$$

$$\tilde{\mathbf{r}}_i = \frac{\mathbf{r}_i}{r_0} \tag{3.5}$$

$$\tilde{M}_i = \frac{M_i}{M_0} \tag{3.6}$$

in this definition, we choose a given reference value for the position r_0, and M_0 is the mass of the central body. The reference value for the position depends on the system being studied. We also define

$$a_0 = \frac{GM_0}{r_0^2} \tag{3.7}$$

which is the reference acceleration associated to the reference position. If we replace these quantities in equation 3.3, we arrive at its dimensionless form

$$\tilde{\mathbf{a}}_i = -\frac{\tilde{\mathbf{r}}_i}{|\tilde{\mathbf{r}}_i|^3} - \sum_{i \neq j} \tilde{M}_j \frac{\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_j}{|\tilde{\mathbf{r}}_i - \tilde{\mathbf{r}}_j|^3} - \sum_{j} \tilde{M}_j \frac{\tilde{\mathbf{r}}_j}{|\tilde{\mathbf{r}}_j|^3} \tag{3.8}$$

which is the equation we wish to work with, as it gives the most general solutions for the central body N-body problem with the proportions as stipulated. A solution to this equation could represent, for example, either the movement of planets around a star or of satellites around the Earth, and to go from the solution from this equation back to a result for an specific scale, one needs only to invert equations 3.4, 3.5 and 3.6, with values for M_0 and r_0 suitable for the system being studied.

Part II

Results

Algorithm Setting

In order to demonstrate the potential for application of the Hermite Scheme in physical systems such as the N-body problem or atomistic systems, in this chapter we present a study of the accuracy and time performance of the method. In order to do so, we developed two serial Hermite Scheme algorithms and a serial N-body fourth-order Runge-Kutta algorithm, both using the Fortran 90 language. For both algorithms we allow the number of bodies to be defined by the user, and in the Hermite algorithms, as means of comparison, we consider two possible numbers of iterations for the corrector: 1 and 3. The Runge-Kutta method was chosen because it is one of the most widely used numerical methods in physics. As our test system, we will consider a classical planetary N-body problem, in special we consider a Central Star System, formed by point-like particles allowed only translational degrees of freedom (no rotation), in which the dynamical equations that we desire to solve are those given by equation 3.8. We chose this system because it is a chaotic physical N-body system, formed by coupled differential equations, proving a challenge to be solved.

The computer used in performing the simulations had an Intel® Core i7-7700 processor, with 8 cores of 3.60GHz each, 32GB RAM and a GeForce GTX 1050Ti graphics board. The operational system was the ubuntu 16.04 LTS.

In this chapter we will present an analysis of the results we obtained from our tests, each section representing a characteristic of the method. We begin with a study of the accuracy of the Hermite Scheme and its applicability to chaotic systems. We then compare its time performance to that of the Runge-Kutta algorithm, taking into account the dependence of the computational time on both the simulation time and on the number of bodies being simulated. We finish presenting the reproduction of two known results, the orbits of the Solar System and an excerpt from Chambers et al(1996) [28] on the stability of 3-planets systems, as a validation to the method.

4 Analysis of Accuracy

The first step when choosing a numerical method which to apply in obtaining the solution of the dynamical equations for a given physical problem is to understand the accuracy needed for said problem, and to find a method with the appropriate accuracy. For many physical systems, a simple second-order integrator (such as the Leapfrog or the Velocity Verlet) suffices. For chaotic systems, it's high sensitivity on the initial conditions makes it such that all the accuracy one can get is welcome. Commonly, however, an increase in accuracy means an increase in computational resources and computational time, which are things that laboratories can't spare. It is then necessary a very good understanding of the physical problem being studied, as well as a study of the computational setting available in order to define the precision of the variables and the time-step size to be used on the simulation, and therefore more robust methods are necessary.

Given the sensibility of chaotic systems on the initial conditions, the use of the highest precision variables is desirable. However, considering the computational resources and human time restrictions (a simulation which needs 5 years to be performed is unreasonable), double precision variables usually offer a good enough option, and are commonly used. With a suitable choice of the time-step size, a fourth-order integrator can cause the accuracy to go beyond that of the double precision variables, making them the most sought for choices in performing the simulations. A fourth-order integrator, as the name suggests, is one which has an accuracy equal to the fourth power of the time-step used. Given the accuracy at which the integrator works, it has a restriction to the size of the time-step that can be used, in other words, if the time-step used is greater than a critical value, the results of the integration diverge. The Hermite Scheme is one such fourth-order integrator, making it a very suitable choice for the application to chaotic systems, in special we are concerned with the classic N-body problem. In order to show this fourth-order accuracy, and to study the time-step range within which the Hermite Scheme is able to properly work, we performed an analysis using a set of simulations for a sample 3-body central body problem as our test system.

In order to perform the accuracy analysis, we would need to compare the values obtained from the simulation to that from the actual solution, that is, compare with the analytical solution. However, in the absence of the analytical solution, we need some reference quantity, dependent on the variables being integrated, who provides a reference value with which to compare the error of the integrator. Since the system we are studying is chaotic, and do not possess an analytic solution, we need another quantity to use in our tests. This is not a problem. It is known that a physical system such as the one being studied must obey a series of physical laws, and for the numerical solution obtained through a computer simulation to be considered physical, it must obey the laws of Physics. Two laws in special are more interesting for us at this moment, the conservation of momentum and conservation of energy, as both momentum and energy depend on the position and velocity (the variables we are integrating), and we can easily calculate them from the initial conditions, making the initial energy and momentum suitable reference values for our study. Given that we are dealing with a central body problem, we opt in special for the angular momentum of the system. In order to verify the accuracy of a given integrator, one studies the dependence of those variables on the time-step used, and this dependence shall give the order of the integrator.

However, two things should be observed in this development. First, we aren't dealing with the position and velocity directly, but with quantities that depend on them, and as thus we need to keep in mind that the dependence of the relative error of those quantities shall be one order higher than that of the variables being integrated, that is, one order higher than the integrator itself. Secondly, the dependence on the time-step is a very useful for integrator which make use of a fixed time-step for all the particles throughout the simulation, which is not our case. The Hermite Scheme makes use of a different time-step for each particle, which vary during the simulation, making it impossible to choose a given one in order to obtain a proper analysis. To deal with this issue, remember that in order to guarantee that the size of the time-steps for each particle remains within the working range of the integrator, we made use of the accuracy parameter η, which is a fixed quantity associated to the time-steps. So instead of analysing the dependence of the relative errors on the time-steps, we look at their dependence on the accuracy parameter η, which gives an analogous result to that which we desire.

We begin our study with one of the simplest known solutions. We simulated a system in which the two orbiting bodies perform circular orbits around the central body, over a total period of 2π, or 10 years, using a 1 iteration corrector Hermite Scheme. We vary the accuracy parameter from 10^{-5} until 1. In order to better illustrate the bottom limit dependence, we used a quadruple precision variable in the simulations. The resulting dependencies of the relative errors of both the energy and the angular momentum are shown in Figure 1. Close analysis of those graphs show a fifth-order dependence on the accuracy parameter, for both quantities. Since this dependence is one order higher than that of the integrator, we confirm that it is indeed fourth-order.

This test concerned a very specific group of solutions, one of the simplest ones. In order to fully analyse the accuracy, we need to study the behavior of the relative errors of more general solutions. We then repeat a simulation with the same configuration for the system, but this time we consider a system in which the two external bodies move in more general, non-circular, eccentric orbits around the central body, the results shown in Figure 2. The new curves show a slightly different inclination for the smaller values of η, but as the

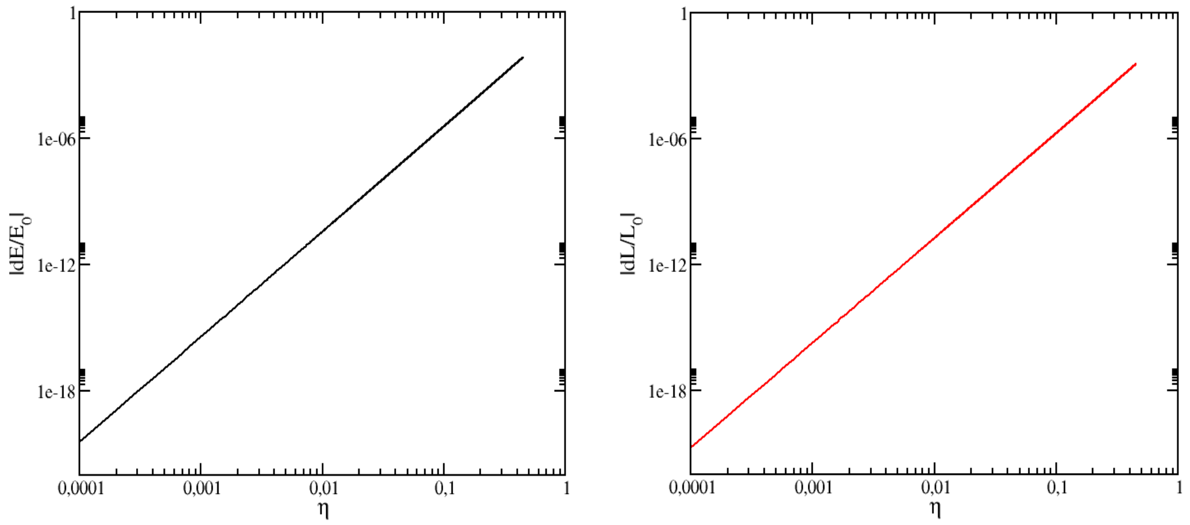

Figure 1 – Dependence of the relative errors for the energy (left) and the angular momentum (right) on the accuracy parameter η, for a system with a fixed central body and two smaller bodies moving in circular orbits, showing a fifth-order dependence.

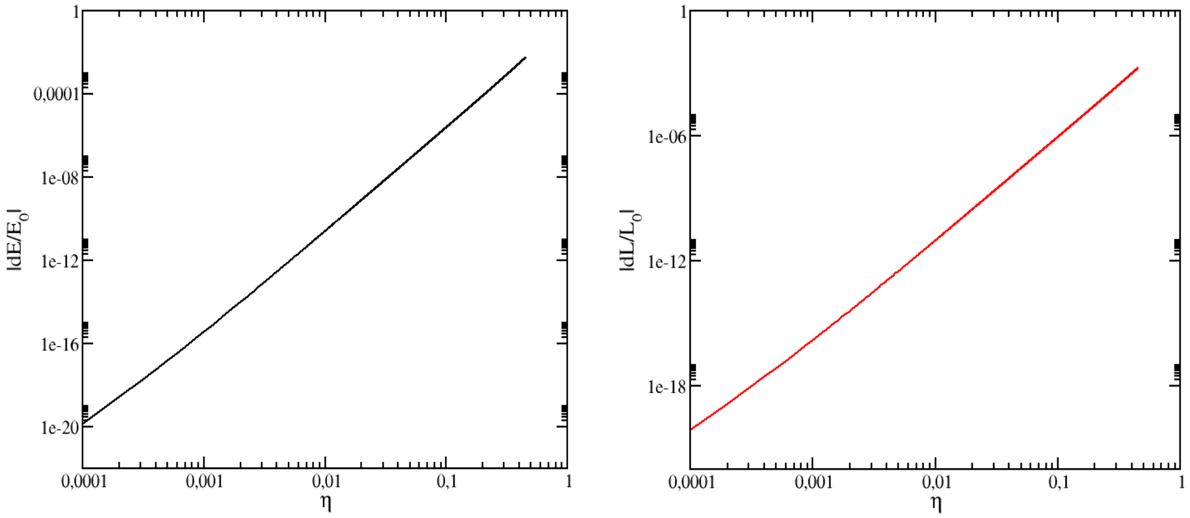

Figure 2 – Dependence of the relative errors for the energy (left) and the angular momentum (right) on the accuracy parameter η, for a system with a fixed central body and two smaller bodies moving in a general non-circular orbits, showing the same fifth-order dependence.

value increases the behavior of the curve approaches that of the previous ones, until it finally becomes the same. The transition occurs for a value of $10^{-4} < \eta < 10^{-3}$. This slight difference does not change the overall dependence of the relative errors, which remain fifth-order, assuring the fourth-order dependence of the integrator.

These results not only provide us with the accuracy of the method, but it also gives us information on the range of values for η within which the integrator can operate. The upper value for η in the graphs is around 0.4, and we found this to be the upper operational value for the integrator. Above this value, the integrator can still operate, but the final results become completely unreliable and non-physical. These graphs also work as a great tool to be used in deciding which accuracy-parameter is more adequate to the accuracy necessary for the given physical system. For a double precision variable, its precision is of the order 10^{-16}, and thus a value of $\eta = 0.01$ is enough for systems in which they are needed.

5 Analysis of Performance

The accuracy of a numerical method is very important, as many times it is what will guarantee if the ensuing solutions will be physical or not. However, it must not be the only element considered in the decision of which method to use. The researcher also needs to weight the computational resources and the time they have available to perform the simulation. The researcher should seek a balance between all those factors, and choose the method that suits most this balance. Sometimes it is worth a sacrifice of the accuracy for a gain in computational time, so long it does not harm the physical result.

Among the factors mentioned above, computational time is perhaps one of the most important, since it translates directly into the human time needed to perform the simulation. The scientist must be constantly show their results to the financing parties, scientific meetings and other circles in order to be able to continue its work, and a method which takes months, or even years to produce a single result isn't appealing, much less practical in scientific works, provided there are other options. With this in mind, we dedicate this chapter to a study of the time performance of the Hermite Scheme for a N-body problem in two situations: considering the total simulation time, and considering a change in the number of bodies. In order to have a sense of the value of its performance, we compare the results to that obtained through a N-body fourth-order Runge-Kutta integrator for the same configurations.

5.1 Computational Time versus Simulation Time

Physics concerns itself with the study of a broad number of natural systems, each containing its own characteristics and particularities. A very important characteristics of all physical systems which can be used to characterize them into groups is the time scale at which the interactions in that system takes place. This can vary greatly from system to system, and for one studying the given system, it reflects on the amount of time needed for the significant phenomenon we wish to study to be observes; e.g., for a quantum system, intervals of the order of pico-seconds and micro-seconds are very significant, whereas for planetary systems a time interval of this size is insignificant. The time scale of the interactions is useful not only to classify different physical systems, but they are also very useful in classifying phenomena within the same system. Oftentimes a system may present itself with a broad number of phenomena taking place within different time scales, the multi-scale systems, and one needs to pay careful attention to the time scales of the phenomena in the system they wish to study. An example of such system is that of a planetary system in which collisions are observed: the collision process takes place within days or months, whereas a study of the orbits of the planets around their star takes years, or even centuries.

For many system which possess an analytical solution, oftentimes this solution provides information about the behavior of the system at any instant of time, but the same can't be said when one uses computer simulations. A numerical solution is a set of calculated points, spaced by a distance determined by the time-step, which gives the behavior of the system being studied within a given time interval determined by the computer physicist. Generally, nothing can be said of the behavior of the system outside of this time interval (this is always true for chaotic systems). This time interval at which the simulation takes place, which represent the real time within which the phenomena being studied happens, is called the *Simulation Time*. The choice of which size of the simulation time to use is directly associated to the time scale of the phenomenon, or phenomena, being studied. For multi-scale systems it is dictated by the scale of the slowest paced phenomenon.

In computational terms, the simulation time is directly associated, and can even dictate, the computational time needed for the simulation to be performed. An influence in this relation is the number and complexity of calculations performed at each step of the integrator: if the algorithm is simple, like that of lower order integrator, the calculations take place faster than that of more robust, higher-order integrator. Since the points in a numerical solution are separated by a time-step, this also means that the simulation time is directly associated to the number of steps that the integrator need to perform in order to provide the desired solution. This is true for methods which make use of a fixed time-step; i.e., the number of steps and the simulation time can be regarded as having the same weight, but both quantities have completely different interpretations when one look ad adaptive time-step algorithms.

This line of thought brings an interesting insight to this work. For methods which make use of a fixed time-step, the number of steps performed will be the same when it processes two different systems with the same time-step and the same simulation time. However, for methods like the Hermite Scheme, different systems making use of the same accuracy parameter and simulation times, most certainly will perform a different number of steps. No direct conclusion can be said of the computational time, however, as the complexity and number of calculations within a step depend on the system being studied. Nonetheless, this opens the question as to the performance of both kinds of methods when applied to the same system, for a fixed simulation time.

In order to compare these performances, it is important to consider the same system, as to guarantee that the complexity and number of calculations will be a characteristic of the method, and not originating from the system being simulated. For this, we choose a planetary central body, 10-body system. We wish to compare a serial 3 iterations corrector

Hermite Scheme and a serial N-body fourth-order Runge-Kutta methods. We used a time-step of 0.001 for the Runge-Kutta algorithm, and an accuracy parameter of 0.01 for the Hermite Scheme. We verified that this choice of accuracy parameter ensures that the smallest time-step of a body in the Hermite Scheme is of about 0.001, thus maintaining the same accuracy as that of the Runge-Kutta. We perform 10 simulations, each with a fixed simulation time, but vary the value between different simulations.

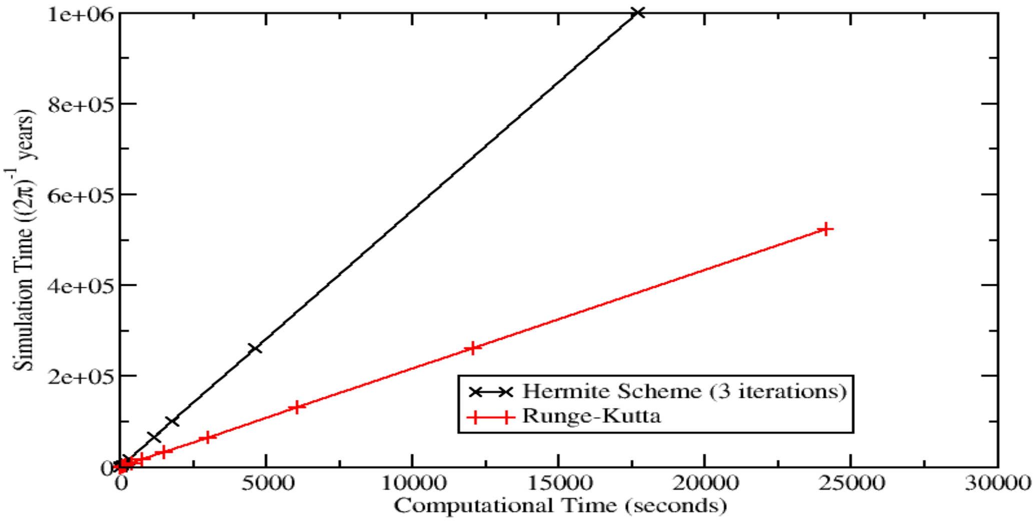

Figure 3 – Comparison of the computational time needed to perform a simulation with a given simulation time, for both Hermite integrator with a 3 iterations corrector and a N-body fourth-order Runge-Kutta.

The results of the study appear in Figure 3. The increase in computational time with the increase in the simulation time is linear, for both methods. The angular coefficient of both lines however is different. We observe that, for a given fixed simulation time, the computational time needed for the Hermite Scheme to perform the simulation is around 2.6 times faster than the Runge-Kutta of same order. These results can be explained through some observed factors. The N-body Runge-Kutta needs to update the coordinates of all the particles at each step, thus increasing the number of calculations per steps performed when compared to the Hermite Scheme, which updates the coordinates of a single particle per step. This means that, while we have observed that the total number of steps taken by the Hermite Scheme were higher than that taken by the Runge-Kutta, the number of outputs of the later were significantly higher than that of the former, which contributes to the increase in computational time.

Also, by doing so, the Runge-Kutta processes all the phenomena based on the time scale of the fastest-paced phenomenon, which many times it means integrating a given particle more times than it is strictly needed. The individual time-step routine in the

Hermite Scheme, coupled with its processing of individual particles, allow for the time-step of each particle to scale with the scale of its interactions, and in this way performing a smarter integration, that is, it integrates each particle just the sufficient amount of times needed for the phenomenon associated to that particular particle to appear in the solution. This allows for a huge economy in computational time, earning the Hermite Scheme the best performance in this situation.

5.2 Computational Time versus Number of Bodies

The result from the previous section shows that the complexity and the number of calculations performed per step is one of the biggest contributors to the difference in computational needed for a given simulation to be performed. This difference in complexity and number of calculations is highly dependent on the model used for the chosen integrator, but also depends on the type of system being considered. For dynamical systems in Physics, this characteristic is directly associated to the number of particles being considered, as for each new particle added to the system, a new interaction arises for every of the remaining $N - 1$ particles. For this reason, when applying N-body simulations to solve physical problems, one of the key factors the researcher needs to pay attention to is a choice of N which permits a good characterization of the system, and allows for a reduced computational cost.

This consideration must take into account the system being studied. For the use of a N-body simulation to describe a continuous system, the N particles corresponds to the discretization choice for the system being studied. For such systems, one needs to pay careful attention to choose a number of discretizations which will not harm the behavior of the physical system, but which also doesn't require an absurd computational cost. One would think that an absurd value for N would be necessary, but there are many methods used by computational physicists; e.g., the application of semi-analytical processes, simplifications in the method or mean field methods, which permit for a good description of the system and allows the use of small values of N. For systems in which its components already are discrete elements, such as the case of N-body systems, one does not have a choice as to the value of N to be used, and needs to rely on possible simplifications of the model, or seek the most efficient method to apply in the given situation.

Then, for a given numerical method we wish to apply in a N-body simulation, it is important to understand its performance not only based on the total simulation time, but also the performance based on the number of bodies being simulated. This is true for the Hermite Scheme we are presenting in this work. We perform the study of the performance of the Hermite Scheme abased on a change of the number of bodies as a set of N-body simulations using a 1 and 3 iterations corrector Hermite Schemes and using a N-body fourth-order Runge-Kutta integrator for comparison. The time-step for the Runge-Kutta is set at 0.001, and the accuracy parameter for both Hermite Schemes is set as 0.01. We vary the number of bodies from 1 up until 8, for time limitation purposes. We considered first the algorithms with a fixed total amount of 1-body outputs, and the results can be seen in Figure 4.

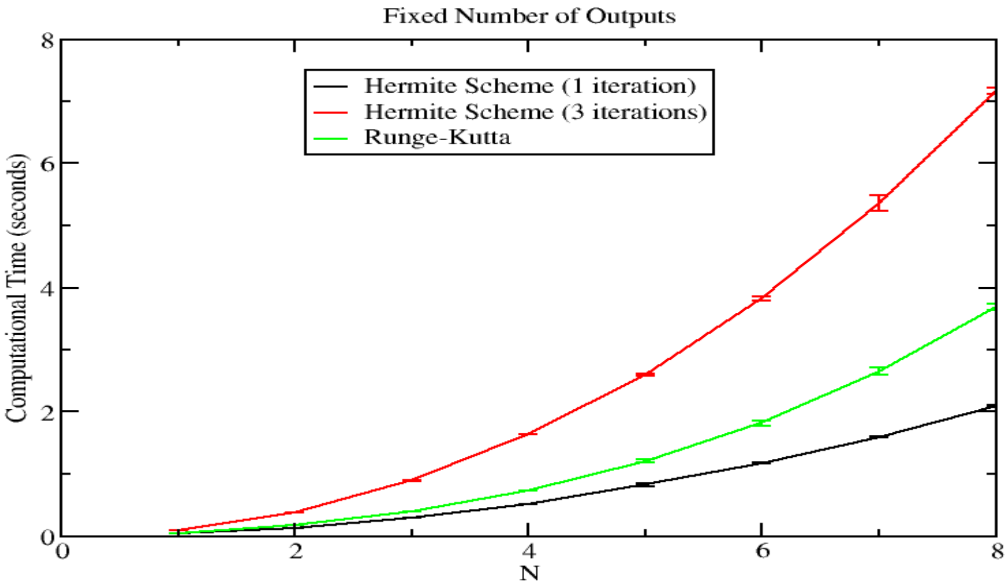

Figure 4 – Dependence of computational time with the increase in the number of bodies in the N-body algorithm for a Hermite Scheme with 1 and 3 iterations, and a fourth order N-body Runge-Kutta algorithms, for a fixed number of outputs.

We observe that when we fix the number of outputs, the 3 iterations Hermite Scheme becomes slower than the Runge-Kutta, contrary to that observed in the previous section. As we noticed in the previous section, the Hermite Scheme integrates the equations in a smarter fashion, producing the same result with a smaller number of 1-body outputs, whereas the Runge-Kutta produces 9 1-body outputs each step. This means that to produce the same amount of outputs, the Runge-Kutta takes 9 times less integration steps than the 3 iteration Hermite Scheme, and therefore making lesser calculations than the Hermite Scheme. This extra calculation, however, isn't due to the structure of the integrator, as surprisingly the 1 iteration Hermite Scheme still fares better than the Runge-Kutta for all values of N but $N = 1$, showing the same accuracy. The discrepancy between the computational time of the two Hermite Scheme is due to the extra corrections of the corrector, which adds a significant amount of extra calculations to the integrator, but adding an extra accuracy as well. We saw that the 1 iteration Hermite Scheme is fourth-order, but the extra corrections add extra accuracy to the method, resulting in a more precise solution, which are preferable in chaotic systems. This simulations aren't entirely fair, however, since the solutions of the Hermite Scheme algorithms aren't the same as that of the Runge-Kutta. Since the Runge-Kutta performs 9 times less integrator steps than the Hermite Scheme, and we choose the accuracy parameter such as the smallest time-step in the Hermite Scheme would be of the same order of that in the Runge-Kutta, this means that the solution given by the Hermite Scheme algorithms covers a greater time range than that of the Runge-Kutta, which mean they did "extra work" and give a more "complete" solution. This makes the fact that the 1 iteration Hermite Scheme was faster even more impressive.

In order to even the odds for all the integrators, we performed a second set of simulations, but now keeping the total simulation time fixed. We already know that, for N = 9, the 3 iteration Hermite Scheme is faster than the Runge-Kutta, but we need to check if that is true for all N. The results for this second set of simulations can be seen in Figure 5.

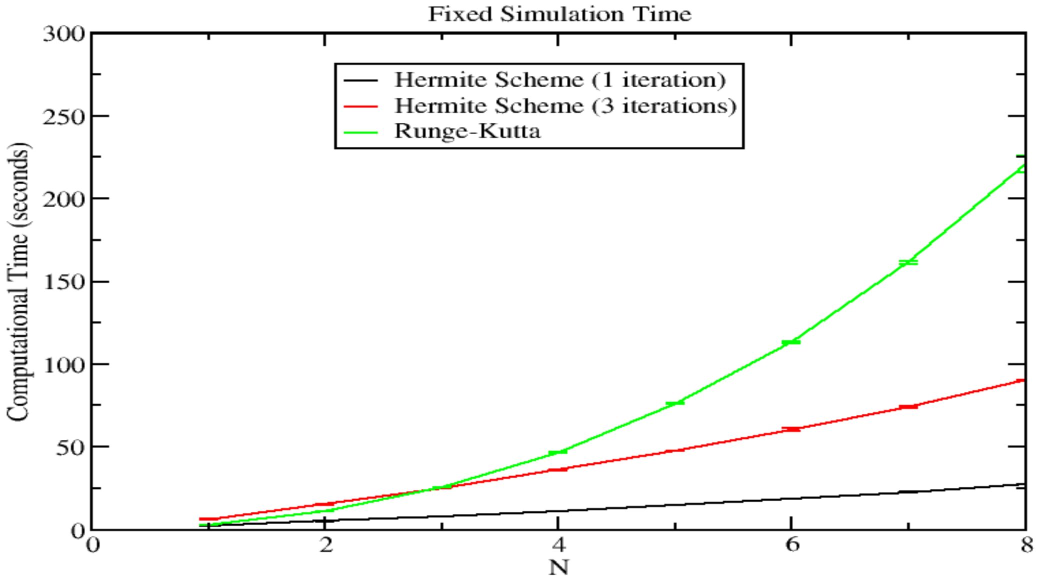

Figure 5 – Computational time dependence with the increase in the number of bodies in the N-body algorithm for a Hermite Scheme with 1 and 3 iterations, and a fourth order N-body Runge-Kutta algorithms, for a fixed simulation time.

Again, for $N = 1$ both Runge-Kutta and the 1 iteration Hermite Scheme perform the same, but for any other value of N the later perform better than any of its peers, a difference in performance that increases as N increases. For the 3 iterations Hermite Scheme and the Runge-Kutta, we see that for $N < 3$, the later performs better than the former, and at $N = 3$, both have the same performance, and for $N > 3$, the performance of the Hermite Scheme becomes increasingly better than that of the Ruunge-Kutta. To understand this result, we need to keep two factors in mind. First the fact we mentioned that oftentimes the Hermite Scheme takes more integration steps than the Runge-Kutta, which for $N < 3$ this means taking more steps overall, therefore performing more calculations than the later, with $N = 3$ being the threshold. Second, we remember that a planetary N-body system is a multi-scale system, and the number of different scale phenomena increase with the number of bodies in the system. In the system we used as test system, up until $N = 3$, the scale of the phenomena wouldn't change much, thus the difference appearing only for $N = 4$. The individual time-step Hermite Scheme is significantly better than the Runge-Kutta when dealing with multi-scale systems, and it can also be that the threshold in the above result may be a consequence of the particular system we used as our test system. Further studies are necessary to confirm this supposition, but overall we see that the Hermite Scheme has a better performance than the Runge-Kutta when dealing with the N-body problem, and we can expect the result to any other N-body systems with similar formulations.

6 Reproduction of Real-life Results

Since the foundation of Computational Physics, its main objectives are the search of new numerical methods to solve physical problems which didn't, or don't, have analytical solutions, or as a means to solve physical problems in general in a faster, more effective fashion. Nowadays, numerical methods are broadly used in a great range of areas in Physics, contributing to experimental and theoretical physicists alike in the development and verification of theories, with many of those areas highly dependent on computational methods for a proper development, such as its the case of both Condensed Matter Physics and Astrophysics.

Throughout history, many new numerical methods have been discovered or created by physicist who needed some way to find a solution to the problem they had at hand. These methods have been thoroughly tested in a great range of problems, and those that fared good started to be applied in areas beyond that to which they were intended for. This reflects the main characteristic of the numerical methods developed for Physics, they need to be able to solve physical dynamical equations, offering a physical solution to the desired problems, with a good efficiency. Methods like the Runge-Kutta's were so successful that until today they are thoroughly studied and developed, with new formulations arising for an enhancement in performance and applicability.

In order to understand if a numerical method is suitable to be applied in Physics, one must first check if it is capable of reproducing already known results in the field, and compare the precision of the numerical solution to the result itself. In this chapter, we aim to verify the applicability of Hermite Scheme to physical systems, in special to N-body systems, by reproducing some known results in Astrophysics, and area in which given its limitations, Computational Physics is broadly used as a means to verify models and theories and provide means of comparison of these with observation.

6.1 Orbits in the Solar System

The night sky has always fascinated human kind with its beauty and complexity. Almost every civilisation that has existed on Earth has proposed a different explanation as to what those little shiny spots were, or what were their function. Some believed them to be the souls of their ancestors that watched them from above, others like the Polynesians used them as guiding compasses in their travels [29], but there were those like the Mayan [30] and the Greek [31], who set to investigate and study these beings, and this allowed them great findings. They would come to discover that some of these shiny dots would maintain their relative position fixed in the night sky, but a few of them would move about during the year. Thus were discovered the planets, of which 5 are visible to the naked eye, namely Mercury, Venus, Mars, Jupiter and Saturn. The extraneous movement of these bodies fascinated scientists from the past, and a search for an explanation to their movements fueled research into astronomy and astrophysics, and was one of the motivations for the studies of Johannes Kepler, and for the formulation of Newton's law of gravitation.

Newton came very close to finding a solution that he so much sought to explain their movement, but couldn't offer an analytical solution to his theory. A full analysis came only centuries later, when numerical methods began being used in Physics, and computer simulations would offer us a glimpse on the movement of these celestial bodies, showing that Newton's theory provided a good solution for the movement of the planets in the Solar System (now known to be 8), but fell short of the perfect match by a difference in the precession in Mercury's perihelion [32]. This later detail would then be explained in the beginning of 20th century by Einstein's theory of General Relativity, one of its first successes. Nowadays, the movement of the 8 planets of our Solar System is well understood, with space agencies like NASA and the ESA keeping track of their movement, as well as that many other celestial bodies that exist in out Solar System.

Although Newton's N-body problem does not offer the fully correct solution for the movement of the planets in our Solar System, it represents a good approximation not only to their orbits, but to the movements of many other celestial bodies throughout our universe which are not subject to an extremely strong gravitational field, and for this reason is highly used as a model for simulations in Astrophysics. In truth, the main reason for it to be used is that classical gravity is much easier to implement in an algorithm, and the computational cost associated is much smaller than when we apply General Relativity, and the overall results are as satisfactory as the later.

The Hermite Scheme was initially developed for an application in astrophysics [25] for an application in N-body systems. To show this applicability we choose one of the most known N-body systems, which has fascinated humanity for milenia: the Solar System. The Sun contains approximately 99.8% of the mass in our Solar System, making the later a central body N-body system. The number of bodies in the Solar System is overwhelming, and for simplicity we will concern our attention to the main bodies, the planets (and Pluto, for hommage and nostalgia). Notice that the choice to simulate the Solar System goes beyond a simple tribute to humanity's history, but the later is a good example of a multi-scale system, for the orbits of the planets occur in very different scales (for instance Mercury's orbital period is of 88 days, whereas Pluto takes approx 248 years to perform a complete revolution), allowing us to see the application of the method to multi-scale systems as well.

For the simulation, we used a 3 iterations Hermite Scheme, with the equations of the planets given by equation 3.8. We used as the initial conditions real data on the planets of the Solar System made available by NASA [33], with a simulation time of around 2500 years, or approximately 10 Pluto orbital periods. The simulation is performed in three dimensions, taking into account the inclination of the orbits of each planet with respect to the plane of the ecliptic (the plane at which the Earth's orbit takes place).

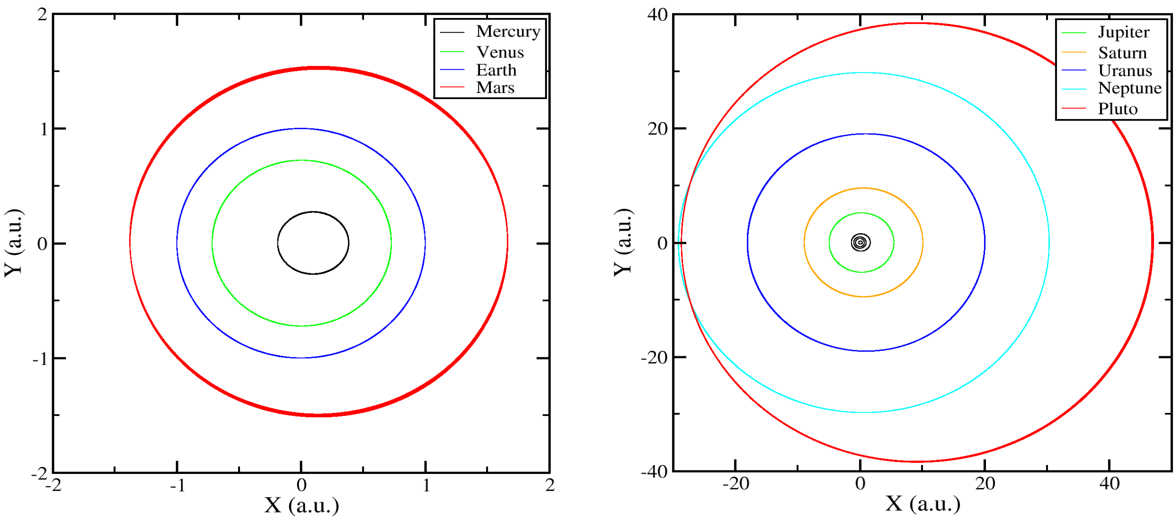

Figure 6 – Projection on the plane of the ecliptic of the orbits of all the planets of the Solar System, plus Pluto, as obtained from a 3 iteration Hermite Scheme algorithm.

Figure 6 shows the orbits obtained from the simulation projected into the plane of the ecliptic. The spatial scales of the orbits differ greatly, so in order to allow for a better visualization of them all, they were separated into two different graphics. Some important characteristics of the planets' orbits can be observed in the solutions. First, the orbital period of the planets were observed to match that found in literature. We can point out some other aspects which we can observe directly from the images.

Earth's orbital eccentricity's value is approximately 0.017, meaning that it's orbit is approximately circular, which is what we observed in our simulation, as well as the highly eccentric orbit of Mercury. One striking point can be observed on the orbits of Neptune and Pluto. In 2006, the International Atronomical Union defined that a planet is "a celestial body that (a) is in orbit round the Sun, (b) has sufficient mass for its self-gravity to overcome rigid body forces so that it assumes a hydrostatic equilibrium (nearly round) shape, and (c) has cleared the neighbourhood around its orbit." [34]. Pluto fails to fulfill the last exigence, as it crosses Neptune's orbit in two points, which we can clearly observe in the results above, helping to validate the obtained result.

6.2 Stability of Circular Orbits

The secrets of the universe are not limited to the movement of the celestial bodies of which it is composed. Even before the planets were discovered, questions as to the nature of these beings, as well as to the formation of the universe itself were asked, and answered by scholars. The first cosmological models can be tracked as back as the Babylonians [35], and further developed until modern day Cosmology. New observational tools and methods were being developed, which allowed us to discover the existence of other types of celestial bodies throughout the universe, showing us that stars and planets aren't exclusive to our solar system, and this served to further fuel our interest in the Cosmos. The further development of the available knowledge led us to understand what is a star, a galaxy and to characterize planetary systems outside our Solar System. This opened space to questions that we still seek to understand, and one of them is of special interest for the project, as it is a theme that gave rise to it: How are planets formed?

It is known that planets form within the so called protoplanetary disks, which are disks of gas and debris leftover from the star formation process, which orbit around a young star. Many observations of such disks can be found on literature [36, 37]. It is believed that dust and debris within the disk start to clump together forming objects of increasing mass, called the planetary embryos, until they reach a critical size at which they start to accrete the material around them until they become a fully formed planet. However, there are still many unanswered questions as to how this growth takes place and what are the conditions that guarantee the formation of a stable planetary system.

Much research has been made in the study of planet formation. Given the mass that is necessary for the formation of a planet, the estimated life-time of a protoplanetary disk, and the estimated time needed for a planetary embryo to accrete enough mass to become a planet, it is understood that collisions among planetary embryos must play an important part in planet growth and planet formation. But it is also known that planetary embryos that are embedded in a protoplanetary disk composed of much smaller objects evolve to move in nearly circular, non-overlapping orbits [28]. However, in 1993, Gladman showed that any system of two small planets around a central star, with orbits of low eccentricity and inclination, are stable if the difference in their semi-major axes measured in Hill radii, which we shall call Δ, is higher than a given critical value, namely $\Delta_{crit} \simeq 2\sqrt{3}$ [28]. The Hill radius of an astronomical body is the radius of a sphere within which smaller objects will tend to orbit it.

With this knowledge, it was not understood how these systems would evolve into one on which close encounters between those embryos are possible, allowing for the occurrence of collisions to occur, with them coalescing into full size planets [28]. This also led to a thought that systems of 3 or more small planets would behave like concatenations of such two-planets subsystems and have the same stability behavior, precluding the formation of planets altogether. However, in 1996 Chambers et al produced a paper [28] in which they studied the stability of multi-planet systems. They showed that any system with 3 or more planets with $\Delta < 10$ are always unstable, with the time of the first close encounter given approximately by the relation $log(t) = b\Delta + c$, in which b and c are constants [28]. They use computational methods in obtaining those results.

The results obtained by Chambers et al. are very important in the study of planet formation, and illustrate the use and importance of the application of computational methods in the study of physical systems, as they could not be obtained by other means [28]. The system considered by them was one in which multi-scale phenomena, with very different scales, were present. A close encounter usually occurs at a scale of days, whereas the orbits of the planets presented in their work would take months or years to complete. Given the huge difference in scale between the phenomena, and the importance of the result presented in the paper, we use this work as a second validation test for the method being presented in this work, limiting our study to the study of the 3-planets system presented in the first part of Chambers et al.

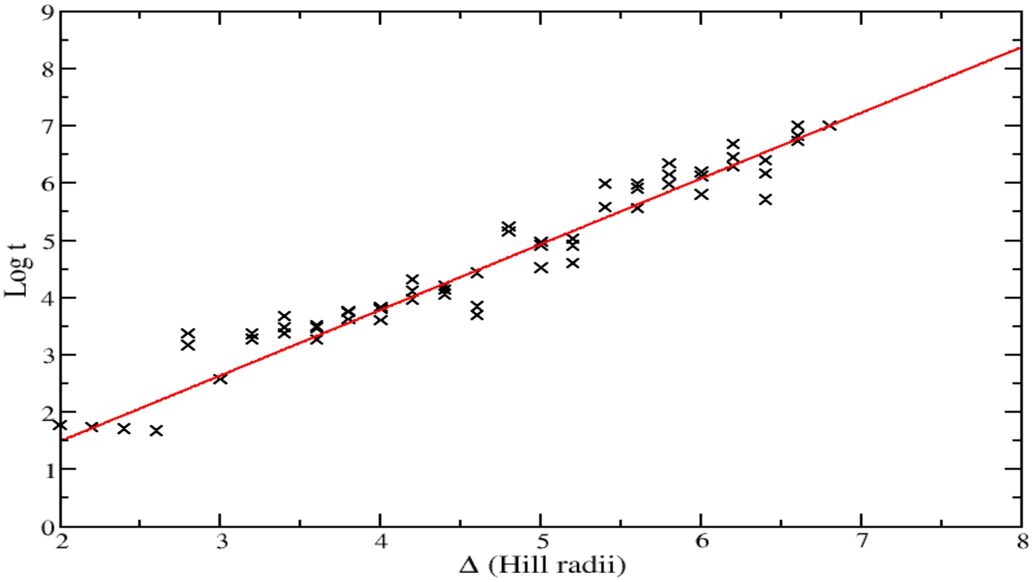

Figure 7 – Dependence of the time of the first close encounter between any two bodies on the value of Δ. We observe the dependence to be the same as that in Chambers et al.

Our simulation consider a 3-planet system with the same configurations as presented in Chambers et al, that is, masses of 10^{-7} times the mass of the central star, set into initially coplanar, circular orbits with semi-major axes given by $a_1 = 1$, $a_2 = a_1 + R_{H1,2}\Delta$ and $a_3 = a_2 + R_{H2,3}\Delta$ respectively, in which the $R_{hi,j}$ are the mutual Hill radius between planets i and j, which are given by the modified equation

$$R_{Hi,j} = \left[\frac{m_i + m_j}{3}\right]^{\frac{1}{3}} \left(\frac{a_i + a_j}{2}\right) \tag{6.1}$$

and Δ is the difference in the semi-major axes of their orbits, measured in units of Hill radii. We use the 3 iterations Hermite Scheme as the integrator, with an accuracy parameter of 0.01. Δ is used as a parameter, varying from 2.0 up to 10.0, in steps of size 0.2. We run 3 simulations for each value of Δ, with the initial positions of the planets decided randomly each run, subjected to the condition that they begin with an angular displacement of at least 20 degrees from each other. Each simulation is carried until a close encounter occur, or until a total simulation time of 10^7 years occur, whichever happens first. We consider a close encounter between planets i and j when $R_{Hi,j} = 1$, and when one occurs we record the time of the occurrence. The results we obtained are shown in figure 7

The line in the figure was obtained using a least squares regression. Since we used a different method than that used in the paper, the points weren't the same, but we observe the same behavior as shown in Figure 1 of Chambers et al. The logarithmic dependence of the time of the first close encounter on Δ is the same as proposed on the paper. For values below $2\sqrt{3}$, the results are that as given by Gladman, but we observe that for a system with $N = 3$, independently of the size of Δ, the system is always unstable if the planets begin at initially circular, coplanar orbits. The validation of this results shows that the Hermite Scheme indeed is capable of processing multi-scale systems, even when the scales in consideration differ by a great amount, allowing for its usage in a great variety of physical systems.

Conclusion

With the development of Computational Physics, many physical systems are studied using numerical methods, which most of the time are cheaper and easier methods to obtain the solution, and sometimes are the only way to obtain the solution, but in both being a faster method to reach the desired result. But when applying computational methods to physical systems, one have to bear in mind the computational resources available, the accuracy needed for the physical system being studied, and the computational time needed for the simulation, so it is of dire importance that the researcher employ numerical methods that can comply and show a balance between those conditions. A feature of many systems which many times pose a challenge to this balance is the presence of multi-scale phenomena, that is, phenomena that happen to the same set of elements in a system but which happen in different time scales, e.g. molecular dynamics with the presence of collisions. In such case, it is important to make use of a good numerical method which is able to process such multi-scale phenomena without sacrifice of time, or loss of performance.

We set to study a numerical method capable of processing such multi-scale phenomena, called the Hermite Scheme, a fourth-order predictor-corrector integrator based on an Hermite Interpolation. This integrator processes each particle of the system individually, and is capable of integrating their dynamical equation by making use of time-steps that are different for each particle. The value of this time-step is related to the intensity of the interactions to which each particle is subjected, and therefore is directly related to the time scale at which the given phenomenon is taking place. This should allow the method to efficiently manage multi-scale systems, without the necessity of changes to its original formulation. In special, we are interested in the performance of the method when applied to N-body systems, so in order to verify its accuracy and performance, we applied the system to a classical N-body problem, making a series of tests and compared the results to to that of an integrator with the same accuracy, the fourth-order N-body Runge-Kutta.

The first test was a verification of the overall accuracy of the Hermite Scheme when applied to chaotic physical systems, which is the case of the N-body problem. Such systems are highly dependent on their initial conditions, and therefore they need an integrator with a good accuracy in order to avoid numerical errors and non-physical solutions. The overall accuracy obtained was of fourth-order, as expected from the method, which reflects in a fifth-order accuracy for the energy and angular momentum. We also observed a scale parameter threshold of around 0.4 for this accuracy, above which we observe great numerical error of the method, which shows that the method has a good behavior even for time-steps of higher values. For a chaotic system such as the N-body problem, a scale parameter of the order 0.01 is enough, which falls well inside the threshold of the method, making the Hermite Scheme a suitable method for studying such systems with a good enough accuracy. A suitable choice of the scale parameter can even cause the error to fall below the accuracy of a double precision variable.

Nonetheless, many methods in the literature show a similar, or even the same, accuracy as observed, which leaves the question on the time performance of the Hermite Scheme when compared to other such methods. We analysed the time performance of the Hermite Scheme, and compared the results we obtained to that of another method with the same accuracy, which widely used in many areas of Physics, the fourth-order Runge-Kutta. We considered for the test both a 1 and 3 iteration corrector Hermite Schemes. Our results have shown that when we fix the total simulation time, the Hermite Scheme has a better overall time performance than the fourth order Runge-Kutta for classical N-body problems with $N \geq 3$, below which it performs better than the 3 iteration Hermite Scheme, but the 1 iteration Hermite Scheme still beats its performance. Nonetheless, when we compare their time performance considering a fixed number of outputs, the Runge-Kutta fares better than a 3 iterations Hermite Scheme for any value of N, but poorer than a 1 iteration Hermite Scheme, except for $N = 1$. However, in this case both Hermite Schemes solve the dynamical equations over a greater range of time than the Runge-Kutta, thus performing "extra work". Also, the 3 iteration Hermite Scheme has a slightly better precision than the Runge-Kutta due to its extra corrections. This that overall, when we consider the task that we want to solve in a physical problem, the Hermite Scheme has a better time performance than the Runge-Kutta for N-body systems, but for other continuous systems, it might show the same performance.

Finally, since we are interested in the application of the method to solve real-life systems, it is not enough to just test dummy systems, but it is necessary that the method can in fact reproduce these real-life phenomena. We then apply the Hermite Scheme algorithm to two known results of N-body systems, both which present multi-scale interactions, namely we simulated the three dimensional orbits of the planets of the Solar Systems and the now Dwarf Planet Pluto, and attempted to reproduce the results from Chambers et al(1996) on the stability of 3-planet systems. The results we obtained were satisfactory, with the orbits showing key characteristics of the celestial bodies in our solar system, such as the two points in which Pluto's orbit crosses that of Neptune, the low eccentricity orbit of the Earth and the orbit of the planets. For the Chambers et al(1996) paper, our result showed the same behavior as observed in the first figure, albeit using a different method. This demonstrates that the Hermite Scheme is indeed capable of processing multi-scale systems, even if the scale difference is that between an orbit step and a close encounter process.

Overall, we see that the Hermite Scheme has a good accuracy, on par with the most often used integrators in Physics, with a good applicability to chaotic dynamical systems. It also presents an overall better time performance when dealing with the N-body systems when compared to the widely used fourth-order Runge-Kutta, representing an economy in computational time without loss of accuracy. A fixed time-step method would need to either use a time-step adapted to the smallest time scale phenomena, or implement a routine to adapt the time-step when dealing with a multi-scale system. For the Hermite Scheme, however, the implementation of the Individual Time-Step procedure allows the algorithm to process multi-scale phenomena naturally, with no necessity for change in the algorithm, and thus no harm to its time performance or accuracy, making it a more efficient method to use when dealing with such types of systems. This procedure, coupled with the fact that the algorithm processes one particle at a time, grants that the system will be less susceptible to cancellation errors as each body will move according to the time scale of its interactions. This makes the method very useful when dealing with systems with both long-range and close-ranged interactions, in which both are important to the physical process being studied. For physical systems in general and when comparing serial methods, the Hermite Scheme can provide a great economy in computational time with the same accuracy as the commonly used methods, without the need for increase in the computational resources available in each laboratory. Although we used as a sample system a planetary N-body problem, the results shown in this work are extensible to any N-body systems, in special it may prove very useful to atomistic systems.

Perspectives

This works was proposed as part of a collaboration between the Simulation Laboratory from the Universidade Federal de Juiz de Fora and the Theoretical Astrophysics Laboratory from Nagoya University. Its initial goal was to produce an algorithm that is capable to study late stage planetary formation, in special the formation of the planet Uranus in our Solar System, by using a N-body simulation. The algorithm needs to be capable of processing close encounters and collisions, an important process in planetary formation, without an extra computational cost or sacrifice of accuracy. In the development of the algorithm, we saw the potential application of the Hermite Scheme for atomistic systems in Condensed Matter Physics and correlated area. The combination of both ideas gave birth to the work here presented, in which albeit we use as the sample system a planetary N-body problem, the results are extensible to any physical N-body system.

As of the writing of this text, the algorithm is capable of identifying a close encounter between any two bodies, but the procedure that processes the collision shall yet be implemented in the future. This procedure is important because collisions are an important mechanism of planetary growth, and it is also observed in atomistic confined systems. We aim to first implement a simple collision procedure, one which takes into account only completely inelastic, considering the coupling of the bodies, or elastic collisions, both without considering fragmentation of any of the bodies. This should allow for a good approximation of collisional events in planetary formation, and we aim to apply the algorithm to a study of the formation of the planet Uranus in our Solar System, and inspect if the presence of Jupiter and Saturn has played any role on its formation. This simplification also applies to the atomistic systems studied in the Simulation Laboratory from UFJF, in which fragmentation of the components isn't present. For these cases, we are yet to apply the method to a system in Condensed Matter Physics and to verify its overall performance.

Depending on the performance of the method for atomistic systems, the next natural step is to consider the parallelization of the algorithm, and to check if there is an improvement on the performance of the algorithm. For planetary formation, it is arguable if fragmentation in collisions plays a much important role in the formation process or not. We shall then further study the collision process, aiming to implement a procedure capable of taking fragmentation into account. If this is successful, we will be able to determine its importance in the formation process, as well as to assess its role in the formation of moons and planetary rings in planetary systems.

Appendix

APPENDIX A – Derivation of the Formula for a Central Body System

Like any other problem in Physics, N-body problem is formulated considering an unconstrained system of N mutually interacting bodies in an inertial frame. Its main applications are on the study of the movements of celestial bodies which we can observe in the night sky, such as in attempts to predict the motion of these bodies, or in order to test theoretical models in comparison to observational data. When we observe the Cosmos, however, we see a majority of systems composed of a set of smaller bodies orbiting a bigger, very massive central body. When studying such systems, most oftentimes one is more concerned about the movement of the smaller bodies around the central one than in the overall movement of the entire system about its surroundings.

To focus on the movement of the smaller bodies, which we from now on will refer to as planets (note that the results here shown are applicable to any situation) about the central body, whom we will refer to as star, one needs only to look at the equations of motion of the planets in the reference frame of the star. One needs to be very careful when doing so, as they might be tempted to use the same equations of motion and integrate them directly. Remember that the equations of motion of a physical system are invariant only within a change from an inertial frame to another, however the reference frame of the star isn't an inertial frame, since the star suffers the gravitational pull of all the planets surrounding it, which results in an acceleration. Thus, when studying the motion of the planets in the reference frame of the star, the equations of motion we shall use will be slightly different than the ones we began with, as we have shown in the text. In this appendix, we will show step-by-step how to derive the equations of motion for a central body system that was shown in the text.

First, let's consider a N-body system with $N+1$ elements. Let the element 0 be the star, whose mass we consider to be M_0, which will be our central body. We define $\mathbf{r}_0$ as its initial position at a given inertial frame S. The remaining N planets have masses M_j (different or not) and initial positions $\mathbf{r}_j$, $j = 1, .., N$ in the frame S. The dynamical equation for body i is given by

$$\mathbf{F}_i = -\sum_{i \neq j}^{N} G M_i M_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} \tag{A.1}$$

in which we sum on the index j from 0 to N.

We want to change from the frame S to another frame S'. Denote origin of the S'

frame by O', and let its position in the S frame be $\mathbf{r}$. The positions of all the particles in the S' frame can be written as

$$\mathbf{r'}_i = \mathbf{r}_i - \mathbf{r} \tag{A.2}$$

the relative position between any two particles in the S' frame will be

$$\mathbf{r'}_{i,j} = \mathbf{r'}_i - \mathbf{r'}_j \tag{A.3}$$

Replacing equation A.2 in equation A.3

$$\begin{aligned} \mathbf{r'}_{i,j} &= \mathbf{r}_i - \mathbf{r} - \mathbf{r}_j + \mathbf{r} \\ \mathbf{r'}_{i,j} &= \mathbf{r}_i - \mathbf{r}_j \\ \mathbf{r}_{i,j} &= \mathbf{r'}_{i,j} \end{aligned} \tag{A.4}$$

that is, the relative position of any pair of bodies remains unchanged in the new reference frame. Now, we can write equation A.1 as

$$\mathbf{F}_i = -GM_iM_0\frac{\mathbf{r}_i - \mathbf{r}_0}{|\mathbf{r}_i - \mathbf{r}_0|^3} - \sum_{i\neq j}^{N} GM_iM_j\frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} \tag{A.5}$$

if the frame S' is an inertial frame, then the equations of motion for all the bodies will have the same form as equation A.10, only with a change from the non-primed variables to the primed ones, that is

$$\mathbf{F'}_i = -GM_iM_0\frac{\mathbf{r'}_i - \mathbf{r'}_0}{|\mathbf{r'}_i - \mathbf{r'}_0|^3} - \sum_{i\neq j}^{N} GM_iM_j\frac{\mathbf{r'}_i - \mathbf{r'}_j}{|\mathbf{r'}_i - \mathbf{r'}_j|^3} \tag{A.6}$$

We want to study the system in the frame of the star. In this frame, the star is located at the origin and is static, thus

$$\begin{aligned} \mathbf{r} &= \mathbf{r}_0 \\ \mathbf{r}'_0 &= \mathbf{0} \end{aligned} \tag{A.7}$$

also, for the planets

$$\mathbf{r}'_i = \mathbf{r}_i - \mathbf{r}_0 \tag{A.8}$$

Since this frame is a non-inertial frame, we can derive equation A.2 twice in order to obtain the equation of motion for particle i

$$\begin{aligned} \mathbf{v'}_i &= \mathbf{v}_i - \mathbf{v}_0 \\ \mathbf{a'}_i &= \mathbf{a}_i - \mathbf{a}_0 \end{aligned} \tag{A.9}$$

on which we have used equation A.7. Now, the acceleration of the star in frame S is

$$\mathbf{a}_0 = \sum_{i=1}^{N} GM_i \frac{\mathbf{r}_i - \mathbf{r}_0}{|\mathbf{r}_i - \mathbf{r}_0|^3} \tag{A.10}$$

Therefore, using equation A.9, the acceleration of planet i in the frame of the star will be

$$\mathbf{a'}_i = -GM_0 \frac{\mathbf{r}_i - \mathbf{r}_0}{|\mathbf{r}_i - \mathbf{r}_0|^3} - \sum_{i \neq j}^{N} GM_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^3} - \sum_{j=1}^{N} GM_j \frac{\mathbf{r}_j - \mathbf{r}_0}{|\mathbf{r}_j - \mathbf{r}_0|^3} \tag{A.11}$$

Finally we apply equations A.8 and A.3, arriving at

$$\mathbf{a'}_i = -GM_0 \frac{\mathbf{r'}_i}{|\mathbf{r'}_i|^3} - \sum_{i \neq j}^{N} GM_j \frac{\mathbf{r'}_i - \mathbf{r'}_j}{|\mathbf{r'}_i - \mathbf{r'}_j|^3} - \sum_{j=1}^{N} GM_j \frac{\mathbf{r'}_j}{|\mathbf{r'}_j|^3} \tag{A.12}$$

APPENDIX B – Solar System Data

Solar system data based on data obtained from NASA Goddard Space Flight Center, obtained on 1st of September of 2019[33].

Solar System Data					
Quantity	MERCURY	VENUS	EARTH	MOON	MARS
Mass ($10^{24} kg$)	0.330	4.87	5.97	0.073	0.642
Diameter (km)	4879	12,104	12,756	3475	6792
Density (kg/m^3)	5427	5243	5514	3340	3933
Gravity (m/s^2)	3.7	8.9	9.8	1.6	3.7
Escape Velocity (km/s)	4.3	10.4	11.2	2.4	5.0
Rotation Period (hours)	1407.6	-5832.5	23.9	655.7	24.6
Length of Day (hours)	4222.6	2802.0	24.0	708.7	24.7
Distance from Sun ($10^6 km$)	57.9	108.2	149.6	0.384*	227.9
Perihelion ($10^6 km$)	46.0	107.5	147.1	0.363*	206.6
Aphelion ($10^6 km$)	69.8	108.9	152.1	0.406*	249.2
Orbital Period (days)	88.0	224.7	365.2	27.3*	687.0
Orbital Velocity (km/s)	47.4	35.0	29.8	1.0*	24.1
Orbital Inclination (degrees)	7.0	3.4	0.0	5.1	1.9
Orbital Eccentricity	0.205	0.007	0.017	0.055	0.094
Obliquity to Orbit (degrees)	0.034	177.4	23.4	6.7	25.2
Mean Temperature (C)	167	464	15	-20	-65

Solar System Data					
Quantity	JUPITER	SATURN	URANUS	NEPTUNE	PLUTO
Mass ($10^{24}kg$)	1898	568	86.8	102	0.0146
Diameter (km)	142,984	120,536	51,118	49,528	2370
Density (kg/m^3)	1326	687	1271	1638	2095
Gravity (m/s^2)	23.1	9.0	8.7	11.0	0.7
Escape Velocity (km/s)	59.5	35.5	21.3	23.5	1.3
Rotation Period (hours)	9.9	10.7	-17.2	16.1	-153.3
Length of Day (hours)	9.9	10.7	17.2	16.1	153.3
Distance from Sun (10^6km)	778.6	1433.5	2872.5	4495.1	5906.4
Perihelion (10^6km)	740.5	1352.6	2741.3	4444.5	4436.8
Aphelion (10^6km)	816.6	1514.5	3003.6	4545.7	7375.9
Orbital Period (days)	4331	10,747	30,589	59,800	90,560
Orbital Velocity (km/s)	13.1	9.7	6.8	5.4	4.7
Orbital Inclination (degrees)	1.3	2.5	0.8	1.8	17.2
Orbital Eccentricity	0.049	0.057	0.046	0.011	0.244
Obliquity to Orbit (degrees)	3.1	26.7	97.8	28.3	122.5
Mean Temperature (C)	-110	-140	-195	-200	-225

*The distance from the sun, perihelion, aphelion, orbital period and orbital velocity are considered with reference to The Earth for The Moon.

Bibliography

1 NITADORI, K.; MAKINO, J. Sixth-and eighth-order hermite integrator for n-body simulations. *New Astronomy*, Elsevier, v. 13, n. 7, p. 498–507, 2008. Cited on page 19.

2 MAKING, J. Comparison of two different tree algorithms. *Journal of Computational Physics*, Elsevier, v. 88, n. 2, p. 393–408, 1990. Cited on page 19.

3 MAKINO, J. A modified aarseth code for grape and vector processors. *Publications of the Astronomical Society of Japan*, v. 43, p. 859–876, 1991. Cited on page 19.

4 MURRAY, C. D.; DERMOTT, S. F. *Solar System Dynamics*. [S.l.]: Cambridge University Press, 2000. Cited on page 19.

5 PRESS, W. H. et al. *Numerical recipes 3rd edition: The art of scientific computing*. [S.l.]: Cambridge university press, 2007. Cited on page 19.

6 DEVANEY, R. *An introduction to chaotic dynamical systems*. [S.l.]: CRC Press, 2018. Cited on page 19.

7 DANFORTH, C. M. Chaos in an atmosphere hanging on a wall. *Mathematics of Planet Earth*, v. 17, 2013. Cited on page 19.

8 EINSTEIN, A.; ROSEN, N. The particle problem in the general theory of relativity. *Physical Review*, APS, v. 48, n. 1, p. 73, 1935. Cited on page 20.

9 SAKURAI, J. J.; COMMINS, E. D. *Modern quantum mechanics, revised edition*. [S.l.]: AAPT, 1995. Cited on page 20.

10 MENDONÇA, J. P. A. de et al. The role of sulfate in the chemical synthesis of graphene oxide. *Materials Chemistry and Physics*, Elsevier, v. 215, p. 203–210, 2018. Cited on page 20.

11 BUIZZA, R. et al. Computation of optimal unstable structures for a numerical weather prediction model. *Tellus A*, Wiley Online Library, v. 45, n. 5, p. 388–407, 1993. Cited on page 21.

12 KRONFELD, K. M.; ROBERTSON, R. E.; KOENIGS, G. J. *Weather hazard threat level computation and display*. [S.l.]: Google Patents, 2015. US Patent 9,134,418. Cited on page 21.

13 THOMPSON, J.; COLLINS, G. A generalized study of precipitation forecasting: Part 1: Computation of precipitation from the fields of moisture and wind. *Monthly Weather Review*, v. 81, n. 4, p. 91–100, 1953. Cited on page 21.

14 TEZDUYAR, T. E. et al. Modelling of fluid–structure interactions with the space–time finite elements: Arterial fluid mechanics. *International Journal for Numerical Methods in Fluids*, Wiley Online Library, v. 54, n. 6-8, p. 901–922, 2007. Cited on page 21.

15 CADOU, J.-M.; POTIER-FERRY, M.; COCHELIN, B. A numerical method for the computation of bifurcation points in fluid mechanics. *European Journal of Mechanics-B/Fluids*, Elsevier, v. 25, n. 2, p. 234–254, 2006. Cited on page 21.

16 DIACU, F. *The solution of the n-body problem.* [S.l.], 1995. Cited 2 times on pages 21 and 40.

17 POINCARÉ, H. *New methods of celestial mechanics.* [S.l.]: Springer Science & Business Media, 1992. v. 13. Cited 2 times on pages 21 and 40.

18 QIU-DONG, W. The global solution of the n-body problem. *Celestial Mechanics and Dynamical Astronomy*, Springer, v. 50, n. 1, p. 73–88, 1990. Cited 2 times on pages 21 and 41.

19 SUNDMAN, K. F. *Recherches sur le problème des trois corps.* [S.l.]: Ex Officina Typographica Societatis Litterariae Fennicae, 1907. Cited 2 times on pages 21 and 40.

20 AARSETH, S. J. *Gravitational N-body simulations: tools and algorithms.* [S.l.]: Cambridge University Press, 2003. Cited 4 times on pages 21, 29, 30, and 33.

21 MAKINO, J. Optimal order and time-step criterion for aarseth-type n-body integrators. *The Astrophysical Journal*, v. 369, p. 200–212, 1991. Cited 2 times on pages 22 and 30.

22 BUTCHER, J. C. A history of runge-kutta methods. *Applied numerical mathematics*, Elsevier, v. 20, n. 3, p. 247–260, 1996. Cited 2 times on pages 24 and 25.

23 BUTCHER, J. C. *Numerical methods for ordinary differential equations.* [S.l.]: John Wiley & Sons, 2016. Cited 2 times on pages 24 and 25.

24 FEHLBERG, E. Classical fifth-, sixth-, seventh-, and eighth-order runge-kutta formulas with stepsize control. 1968. Cited on page 24.

25 MAKINO, J.; AARSETH, S. J. On a hermite integrator with ahmad-cohen scheme for gravitational many-body problems. *Publications of the Astronomical Society of Japan*, v. 44, p. 141–151, 1992. Cited 4 times on pages 32, 33, 34, and 58.

26 BRACKBILL, J. U.; COHEN, B. I. *Multiple time scales.* [S.l.]: Academic Press, 2014. v. 3. Cited 2 times on pages 32 and 33.

27 CHANDLER, D. Introduction to modern statistical mechanics. *Introduction to Modern Statistical Mechanics, by David Chandler, pp. 288. Foreword by David Chandler. Oxford University Press, Sep 1987. ISBN-10: 0195042778. ISBN-13: 9780195042771*, p. 288, 1987. Cited on page 41.

28 CHAMBERS, J.; WETHERILL, G.; BOSS, A. The stability of multi-planet systems. *Icarus*, Elsevier, v. 119, n. 2, p. 261–268, 1996. Cited 3 times on pages 45, 60, and 61.

29 SELIN, H. *Astronomy across cultures: the history of non-Western astronomy.* USA: Springer Science & Business Media, 2012. v. 1. Cited on page 57.

30 SMILEY, C. H. The antiquity and precision of mayan astronomy. *Journal of the Royal Astronomical Society of Canada*, v. 54, p. 222, 1960. Cited on page 57.

31 HEATH, T. L. *Greek astronomy.* New York: Courier Corporation, 1932. Cited on page 57.

32 WINBERG, S. Gravitation and cosmology. *ed. John Wiley and Sons, New York*, 1972. Cited on page 58.

33 WILLIAMS, D. D. R. *Planetary Fact Sheet - Metric, NASA Goddard Space Flight Center*. 2019. Disponível em: <https://nssdc.gsfc.nasa.gov/planetary/factsheet/>. Cited 2 times on pages 59 and 71.

34 UNION, I. A. *Planetary Fact Sheet - Metric, NASA Goddard Space Flight Center*. 2006. Disponível em: <https://www.iau.org/news/pressreleases/detail/iau0603/>. Cited on page 59.

35 SACHS, A. Babylonian observational astronomy. *Phil. Trans. R. Soc. Lond. A*, The Royal Society, v. 276, n. 1257, p. 43–50, 1974. Cited on page 60.

36 FUKAGAWA, M. et al. Near-infrared images of protoplanetary disk surrounding hd 142527. *The Astrophysical Journal Letters*, IOP Publishing, v. 636, n. 2, p. L153, 2006. Cited on page 60.

37 FLAHERTY, K. M. et al. Weak turbulence in the hd 163296 protoplanetary disk revealed by alma co observations. *The Astrophysical Journal*, IOP Publishing, v. 813, n. 2, p. 99, 2015. Cited on page 60.