



Instituto de Física
Programa de Pós-Graduação em Física
Universidade Federal do Rio Grande do Sul

NOVOS MAPAS SIMPLÉTICOS PARA INTEGRAÇÃO DE SISTEMAS
HAMILTONIANOS COM MÚLTIPLAS ESCALAS DE TEMPO

Enfoque em Sistemas Gravitacionais de N -corpos

GUILHERME GONÇALVES FERRARI

Tese realizada sob orientação do Dr. Horacio Alberto Dottori e apresentada ao Instituto de Física da UFRGS em preenchimento parcial dos requisitos para a obtenção do título de Doutor em Ciências.

07 de Agosto de 2015

Guilherme Gonçalves Ferrari: *Novos Mapas Simpléticos para Integração de Sistemas Hamiltonianos com Múltiplas Escalas de Tempo*¹, © 07 de Agosto de 2015

ORIENTADOR:

Dr. Horacio Alberto Dottori

LOCAL:

Porto Alegre, RS

DATA:

07 de Agosto de 2015

¹ Trabalho parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e pelo *Netherlands Research Council* (NWO).

Dedicado à minha mãe, Graça
Ao meu pai, Julival (*in memoriam*)
Aos meus irmãos, Gustavo e George
(e aos meus amiguinhos com 4 patas, Luck, Jujuba e Frederico 😊)

RESUMO

Mapas simpléticos são bem conhecidos por preservarem o volume do espaço de fase em dinâmica Hamiltoniana e são particularmente apropriados para problemas que requerem longos tempos de integração. Nesta tese nós desenvolvemos abordagens baseadas em mapas simpléticos para o acoplamento de multi sub-sistemas/domínios astrofísicos/códigos de simulação, para integração eficiente de sistemas de N -corpos auto-gravitantes com grandes variações nas escalas de tempo características. Nós estabelecemos uma família de 48 novos mapas simpléticos baseados numa separação Hamiltoniana recursiva, que permite que o acoplamento ocorra de uma maneira hierárquica, contemplando assim todas as escalas de tempo das interações envolvidas. Nossa formulação é geral o suficiente para permitir que tal método seja utilizado como receita para combinar diferentes fenômenos físicos, que podem ser modelados independentemente por códigos especializados. Nós introduzimos também uma separação Hamiltoniana baseada em Hamiltonianos de Kepler, para resolver o problema gravitacional geral de N -corpos como uma composição de N^2 problemas de 2-corpos. O método resultante é exato para cada problema de 2-corpos individual e produz resultados rápidos e precisos para sistemas de N -corpos quase-Keplerianos, como sistemas planetários ou um aglomerado de estrelas que orbita um buraco-negro supermassivo. O método é também apropriado para integração de sistemas de N -corpos com hierarquias intrínsecas, como um aglomerados de estrelas com binárias compactas. Nós apresentamos a implementação dos algoritmos mencionados e descrevemos o nosso código tupan, que está publicamente disponível na seguinte url: <https://github.com/ggf84/tupan>.

Palavras-chave: Dinâmica estelar; Simulações de N -corpos; Mapas simpléticos; Integração numérica; GPGPU.

ABSTRACT

Symplectic maps are well known for preserving the phase space volume in Hamiltonian dynamics and are particularly suited for problems that require long integration times. In this thesis we develop approaches based on symplectic maps for the coupling of multi sub-systems/astrophysics domains/simulation codes for efficient integration of self-gravitating N -body systems with large variation in characteristic time-scales. We establish a family of 48 new symplectic maps based on a recursive Hamiltonian splitting, which allow the coupling to occur in a hierarchical manner, thus contemplating all time-scales of the involved interactions. Our formulation is general enough to allow that such method be used as a recipe to combine different physical phenomena which can be modeled independently by specialized simulation codes. We also introduce a Keplerian-based Hamiltonian splitting for solving the general gravitational N -body problem as a composition of N^2 2-body problems. The resulting method is precise for each individual 2-body solution and produces quick and accurate results for near-Keplerian N -body systems, like planetary systems or a cluster of stars that orbit a supermassive black-hole. The method is also suitable for integration of N -body systems with intrinsic hierarchies, like a star cluster with compact binaries. We present the implementation of the mentioned algorithms and describe our code `tupan`, which is publicly available on the following url: <https://github.com/ggf84/tupan>.

Keywords: Stellar dynamics; N -body simulations; Symplectic maps; Numerical integration; GPGPU.

PUBLICAÇÕES E PRÊMIOS

Algumas das ideias e figuras apresentadas no Cap. 4 apareceram previamente na seguinte publicação:

Gonçalves Ferrari, Boekholt, and Portegies Zwart (2014). *A Keplerian-based Hamiltonian splitting for gravitational N-body simulations*.

Em 13 de Março de 2014 nós fomos agraciados com o prêmio Win Nieuwpoort², oferecido pelo SURFsara, centro de supercomputação Holandês, por termos alcançado alta eficiência paralela com o método descrito na publicação acima.

Outros trabalhos pelo autor incluem:

Gonçalves Ferrari, Pelupessy, Martínez-Barbosa and Portegies Zwart (2015, to be submitted to MNRAS). *Generalizations of the Bridge Integrator*.

Ferrari, Dottori, and Díaz (2013). *Modeling the Black Hole Recoil from the Nucleus of M83*.

² Anúncio oficial disponível na seguinte url:

<https://www.surf.nl/en/agenda/2014/03/surfsara-data--computing-infrastructure-event/surfsara-data-and-computing-infrastructure-event.html>.

ACKNOWLEDGMENTS

I suppose this is the only part of this thesis people will be interested in reading. And, since the rest of the thesis is written in Portuguese, I decided to write at least the acknowledgments part in English so I can hopefully reach a wider audience here.

First, I'd like to thank Horacio Dottori, my supervisor, for the guidance, support, incentive, friendship, patience, understanding and all the opportunities he has provided for my professional growth. A special thanks goes to his wife, Isa Beatriz Noll, for the support and incentive.

I spent half of my Ph.D. working with Simon Portegies Zwart, in Leiden, the Netherlands. It has been a very enjoyable and fruitful experience, which resulted in the work presented in Cap. 4 of this thesis. His enthusiasm, exuberance and inexhaustible energy has always made it a pleasure to work with him. I'm very grateful for the opportunity to have worked with him.

My thanks also goes to Tjarda Boekholt and Noel Lopez. They were not only good officemates. Tjarda's collaboration was indispensable for the development of the ideas and implementation of the sakura integrator. And Noel, always worried about me and making sure that I was also enjoying life. We became indeed very good friends over the two years of my stay in the Netherlands. Their friendship was paramount to make me feel at home in a little country replete of nice crazy people ☺. Thank you Noel and Tjarda, for everything!

A special thanks goes to Inti Pelupessy. It has been a pleasure to work with him. I learned a lot with Inti, and his work, in many ways, has become an inspiration to me. Inti, I hope we can continue collaborating on many different topics related to N -body simulations.

My stay in the Netherlands wouldn't be so awesome if wasn't for the help, support, guidance and friendship from several people. I'm going to list their names here and I'd like to express my gratitude for everything they've done for me. Here we go: Simon Portegies Zwart, Evelijn Gerstel, Yvonne Kluijt, Carmen Martinez Barbosa, Tjarda Boekholt, Noel Lopez, Susy Blacio, Inti Pelupessy, Michiko Fujii, Nathan de Vries, Lucie Jilkova, Ainil Syalvianty, Daniel Caputo, Edwin van der Helm, Silvia Toonen, Paula Ortiz Otalvaro, Liliana Rivera, Adrian Hamers, Steven Rieder, Jeroen Bédorf, Alex Rimoldi, Arjen van Elteren, Anita van der Tang, Liesbeth van der Veld, Mrs P.E.C. Wieland and probably many others. Thank you all! My thanks also goes to all the people that make the Leiden Sterrewacht a very pleasant and fruitful environment for scientific research.

Here on the southern side of the world there are so many people I must thank that I'll not try to list all their names, otherwise I'd end up forgetting someone.

So I just want to thank everyone for everything! Some of those people notwithstanding deserve a special thanks: to Horacio Dottori, Rogério Riffel, Tarcísio Nunes Teles, Tibério Borges Vale, Jose Andres Hernandez Jimenez, Saulo Cordeiro Lima, Daniel Alf Drehmer, Dinalva Aires de Sales and Alejandra Daniela Romero (Ale, thanks for your hugs, they've helped me more than you could have imagined!). Thanks to all the colleagues and professors of the Astronomy Department and all my past professors. Thanks to all the people behind the PPGFis and the direction of the Physics Institute at UFRGS.

A special thanks goes to Rainer Karl Madejsky, my first supervisor during my Bachelor at UEFS, who first introduced me to the world of N -body simulations.

Many thanks to Guido van Rossum and the Python community, specially those involved with the creation and maintenance of the `numpy`, `pyopencl`, `cffl`, `matplotlib` and `vispy` packages.

I thank the members of my Ph.D. committee for agreeing to read my thesis.

Finally, I thank CNPq, CAPES and NWO, for the financial support.

Guilherme Gonçalves Ferrari

Porto Alegre, RS, 07 de Agosto de 2015

SUMÁRIO

Lista de Figuras	xv
Lista de Tabelas	xvii
Listagens	xix
Acrônimos	xxi
i INTRODUÇÃO	1
1 INTRODUÇÃO	3
1.1 Contextualização e Justificativa	3
1.2 Organização da Tese	6
ii METODOLOGIA E RESULTADOS	7
2 FORMULAÇÃO TEÓRICA	9
2.1 Conceitos Básicos	10
2.2 Formulação Hamiltoniana	13
2.3 Correções Relativísticas para Interações entre SMBHs	16
3 IMPLEMENTAÇÃO NUMÉRICA	21
3.1 Por que um Novo Código de N -corpos?	22
3.2 tupan: Uma Visão Geral do Código	24
3.3 Detalhes de Implementação	25
3.3.1 Módulo io	25
3.3.2 Módulo particles	25
3.3.3 Módulo ics	26
3.3.4 Módulo lib	27
3.3.5 Módulo integrator	28
3.3.6 Módulo simulation	28
3.3.7 Módulo animation	28
3.4 Exemplos de Uso Prático com tupan	28
3.4.1 Modelo de Plummer virializado com uma IMF tipo SALPETER1955	29
3.4.2 Executando uma simulação com visualização em tempo real	29
3.5 Medidas de Performance	31
4 NOVAS TÉCNICAS DE INTEGRAÇÃO NUMÉRICA	33
4.1 Separação Hierárquica do Hamiltoniano: <i>the slow-fast split</i>	33
4.1.1 Critério para o passo de tempo	34
4.1.2 Generalização à uma família de integradores	36
4.1.3 Operadores <i>drift</i> e <i>kick</i> com correções relativísticas	38
4.2 Separação Kepleriana do Hamiltoniano: <i>sakura</i>	41
4.3 Validação Numérica	44
4.3.1 Sistemas com poucas partículas	45
4.3.2 Sistemas com muitas partículas	47
4.4 Paralelização: <i>Win Nieuwpoort prize</i>	53

iii	CONCLUSÕES	55
5	CONCLUSÕES E PERSPECTIVAS	57
iv	APÊNDICES	61
A	FORMA EXPLÍCITA DOS TERMOS PÓS-NEWTONIANOS	63
B	WIN NIEUWPOORT AWARD	67
	REFERÊNCIAS BIBLIOGRÁFICAS	69

LISTA DE FIGURAS

Figura 3.1	Alguns dos módulos que compõem tupan.	24
Figura 3.2	Fluxo de execução nas diferentes partes de tupan.	27
Figura 3.3	Modelo construído com o trecho de código na List. 3.1. O painel à esquerda mostra a distribuição das massas das partículas seguindo uma IMF tipo SALPETER1955. O painel à direita mostra um modelo de Plummer com 4096 partículas em equilíbrio virial. O tamanho dos símbolos é proporcional à massa das partículas, enquanto o círculo laranja demarca, para comparação, o raio virial do sistema.	29
Figura 3.4	Janela de visualização 3D lançada por tupan quando a opção --view é ativada durante sua execução.	30
Figura 3.5	Evolução temporal das energias do sistema mostrado na Fig. 3.4.	30
Figura 3.6	Medidas de desempenho para o cálculo da aceleração gravitacional com tupan, para diferentes números de partículas. As barras dão a performance em Gflops/s, correspondente ao eixo à esquerda. As linhas com pontos dão o tempo de execução em segundos, correspondente ao eixo à direita.	32
Figura 4.1	A figura mostra o erro relativo da energia total como função do tamanho médio do passo de tempo (painéis à esquerda) e o tempo de CPU como função do erro da energia (painéis à direita) para três sistemas: figure8 (painéis superiores), pythagorean (painéis do meio) e solarsystem (painéis inferiores). Resultados com sakura são mostrados em linhas cheias com círculos pretos, sia21a.dkd (Leapfrog) em linhas pontilhadas com triângulos azuis e Hermite em linhas tracejadas com quadrados vermelhos. As grandezas estão em N -body units e o tempo de CPU é dado em segundos.	46
Figura 4.2	O mesmo que Fig. 4.1, mas para o sistema hierárquico de binárias com as seguintes razões de semi-eixo maior: $a_{\text{outer}}/a_{\text{inner}} = 10$ (painéis superiores), $a_{\text{outer}}/a_{\text{inner}} = 100$ (painéis do meio) e $a_{\text{outer}}/a_{\text{inner}} = 1000$ (painéis inferiores).	48
Figura 4.3	A figura mostra o erro relativo da energia total como função da razão de massas entre SMBH e estrelas, para três tamanhos do passo de tempo $\langle \tau \rangle = [10^{-3}, 10^{-4}, 10^{-5}]$ (linhas superiores, do meio e inferiores, respectivamente). sakura é comparado com sia21a.dkd (Leapfrog) (painel à esquerda) e Hermite (painel à direita). O esquema de cores é o mesmo da Fig. 4.1.	49

Figura 4.4	Para o mesmo sistema da Fig. 4.3, os painéis mostram o tempo de CPU como função do erro da energia para as seguintes razões de massa: $q = 10^3$ (superior esquerdo), $q = 10^6$ (superior direito), $q = 10^9$ (inferior esquerdo) and $q = 10^{12}$ (inferior direito).	50
Figura 4.5	A figura mostra a evolução temporal do raio do <i>core</i> de um aglomerado com 1024-corpos inicialmente em equilíbrio virial. Nós comparamos sakura usando diferentes tamanhos do passo de tempo $\tau = [10^0, 10^{-1}, 10^{-2}, 10^{-4}]$ (linhas pretas de cima para baixo), com sia21a.dkd (Leapfrog) (linha pontilhada azul) e Hermite (linha tracejada vermelha), ambos usando passo de tempo variável. As grandezas estão em <i>N-body units</i>	51
Figura 4.6	Painel superior: <i>strong scaling</i> para valores de $N \geq 64k$ com p até 6144 <i>cores</i> de processamento. Painéis inferiores: <i>strong scaling</i> (esquerda) e eficiência paralela (direita) para valores de $N \leq 64k$ com p até 128 <i>cores</i>	53
Figura B.1	A figura mostra o diploma referente ao prêmio Win Nieuwpoort, recebido na cerimonia de premiação em 13 de Março de 2014 em Amsterdam.	67

LISTA DE TABELAS

Tabela 4.1	Família de 48 mapas simpléticos implementados em tupan, classificados por nome, ordem de integração (Y), números de estágios (Z), esquema de passos de tempo e as devidas referências para os coeficientes a_p e b_p	39
------------	--	----

LISTAGENS

Listagem 3.1	(Python shell): Usando tupan para gerar um modelo de Plummer virializado com uma IMF tipo SALPETER1955. . .	29
Listagem 3.2	(Unix shell): Usando tupan para executar uma simulação com visualização em tempo real.	29

ACRÔNIMOS

ADM	Arnowitt-Deser-Misner
BCH	Baker-Campbell-Hausdorff
BTS	<i>Block-Time-Step</i>
CPU	<i>Central Processing Unit</i>
GPU	<i>Graphics Processing Unit</i>
GRAPE	<i>GRAvit PipE</i>
HPC	<i>High-Performance Computing</i>
IMF	<i>Initial Mass Function</i>
KISS	<i>Keep It Simple, Stupid</i>
LISA	<i>Laser Interferometer Space Antenna</i>
MPI	<i>Message Passing Interface</i>
MVS	<i>Mixed Variable Symplectic</i>
PBaSS	<i>Particle-Based Sampling Scheme</i>
PN	<i>post-Newtonian</i>
GR	<i>General Relativity</i>
SMBH	<i>Supermassive Black-Hole</i>
SPH	<i>Smoothed Particle Hydrodynamics</i>

Parte I

INTRODUÇÃO

One reason that life is complex is that it has a real part and an imaginary part.

— Andrew Koenig



INTRODUÇÃO

Nas últimas cinco décadas sistemas computacionais têm mostrado um crescimento expressivo em seu poder de computação, permeando e influenciando quase todos os aspectos de nossa sociedade. O impulso inicial para a busca por computadores cada vez mais rápidos foi dado pela comunidade científica, que desde então tem-se beneficiado grandemente desse aumento contínuo em performance computacional. Igualmente importante para o desenvolvimento científico atual é a produção de pacotes de *software* cada vez mais rápidos e sofisticados. Avanços nestas duas fronteiras tem gradualmente expandido a importância da computação na ciência, de uma mera ferramenta para análise numérica, à um ambiente plenamente desenvolvido para realização de experimentos virtuais.

Tradicionalmente, a investigação científica tem-se baseado em dois pilares: *teoria* e *experimento*. A disponibilidade dos assim chamados *laboratórios virtuais* (Hut, 2010), tem permitido a reprodução de fenômenos naturais complexos, e vem resultando na ascensão da *simulação* como um terceiro pilar da pesquisa científica. Simulações tem a grande vantagem de permitir a investigação de fenômenos muito difíceis, ou mesmo impossíveis, de serem reproduzidos em um laboratório real. Por outro lado, estudos teóricos são muitas vezes limitados à descrição de fenômenos cujas equações que o descrevem são analiticamente solúveis. Quando isto não é possível ou quando métodos perturbativos já não são suficientes para o seu completo entendimento, experimentos virtuais são a única alternativa viável.

A modelagem de fenômenos naturais, o desenvolvimento de pacotes de *software*, juntamente com a tarefa de realizar e analisar os dados de simulações, caracterizam o trabalho de um cientista computacional.

1.1 CONTEXTUALIZAÇÃO E JUSTIFICATIVA

Historicamente, simulações astrofísicas tem evoluído de maneira direta com o desenvolvimento de sistemas computacionais, tornando-se uma ferramenta indispensável para o entendimento de grandes bases de dados observacionais e para o ganho de conhecimento sobre a formação e evolução de sistemas astrofísicos. A demanda por tais simulações tem crescido em conformidade

com o aumento e a qualidade dos dados observacionais, especialmente com a recente introdução de arquiteturas multi-core tais como em *Central Processing Units (CPUs)* e *Graphics Processing Units (GPUs)*, que atuam como aceleradores de desempenho de propósito geral.

Simulações gravitacionais de N -corpos requer ambos, uma forma rápida para calcular as forças e um método de integração suficientemente preciso para evoluir as partículas no tempo. Uma série de metodologias tem sido desenvolvidas nos anos passados e uma escolha tem que ser feita com base no sistema que pretende-se modelar. Por exemplo, métodos de soma direta das forças são mais precisos, porém mais caros computacionalmente $\mathcal{O}(N^2)$, e requerem esquemas de integração de altas ordens a fim de manterem a precisão da simulação por longos períodos de tempo. Estes são tipicamente mais apropriados para simulações de sistemas colisionais tais como sistemas planetários, aglomerados estelares, núcleos galácticos e dinâmica de buracos-negros. No outro extremo encontra-se métodos de implementação muito mais complexa, baseados numa estrutura de dados em árvore (*tree-codes*, Barnes and Hut, 1986) ou numa expansão multi-polar do potencial gravitacional (*fast-multi-pole method*, Greengard and Rokhlin, 1987), que reduzem a complexidade computacional para $\mathcal{O}(N \log(N))$ e $\mathcal{O}(N \log(\log(N))) \approx N$, respectivamente, às custas de uma estimativa menos precisa para as forças gravitacionais. Tais métodos tipicamente adotam um simples esquema *leap-frog* para integração temporal das partículas e são apropriados para estudos de sistemas não-colisionais tais como dinâmica de galáxias e simulações cosmológicas.

Simulações astrofísicas *realísticas* e completamente auto-consistentes, entretanto, tem sido praticamente impossíveis nos anos passados, tanto pela falta de suficiente poder computacional, quanto pela ausência de códigos implementando múltiplos domínios astrofísicos, como *dinâmica gravitacional*, (*magneto*-) *hidrodinâmica*, *evolução estelar*, *cosmologia*, *transferência radiativa*, *evolução química*, dentre outros. Nas décadas de 1970 e 1980 os primeiros códigos mono-domínio foram estabelecidos juntamente com métodos para simulação de cada domínio separadamente. Os domínios da *dinâmica gravitacional* e *cosmologia* foram combinados já na década de 1980 (Efsthathiou et al., 1985) e, uma à duas décadas depois, estudos combinando *dinâmica gravitacional* e *evolução estelar* foram realizados (Portegies Zwart et al., 1997b,a, 1999, 2001, 2004, 2006, 2007). Com a introdução de arquiteturas paralelas, placas dedicadas GRAvit PipE (GRAPE, Sugimoto et al., 1990), e mais recentemente, GPUs, um grande aumento em performance computacional foi alcançado (tipicamente 2 ordens de magnitude) e os códigos atuais tem melhorado grandemente em termos de sua faixa de aplicabilidade dinâmica e precisão numérica. Contudo, um importante *bottle-neck* na área de astrofísica computacional continua sendo o desenvolvimento de *software* capaz de modelar um sistema *realisticamente*.

No contexto de simulações empregando múltiplos domínios astrofísicos, existe atualmente duas vertentes sobre como tais domínios são combinados/acoplados. A primeira vertente baseia-se na construção de códigos seguindo uma estrutura monolítica, onde, partindo de um código gravitacional bem fundamentado, mais fenômenos físicos são gradativamente adicionados à estrutura existente. Um exemplo de tal filosofia está concretizada no famoso código Gadget (Springel, 2005), que em sua versão original combina os domínios da *dinâmica*

gravitacional, cosmologia e hidrodinâmica, e tem sido amplamente adotado pela comunidade científica como base para diversos estudos e implementação de outros fenômenos igualmente importantes. Uma segunda vertente baseia-se na construção de códigos seguindo uma estrutura modular, onde cada domínio astrofísico é implementado de maneira autônoma e independente dos demais. Neste caso, um código adicional deve ser escrito para cada aplicação em específico, no qual os diferentes domínios necessários são acoplados da maneira desejada. Um exemplo de tal filosofia está concretizada no *framework* AMUSE (Pelupessy et al., 2013), que implementa uma interface em Python para diversos códigos existentes.

Experiência tem mostrado que ambas as abordagens acima possuem limitações. A maior parte dos códigos computacionais em uso foram desenvolvidos com um problema particular em mente, e pode não ser utilizável fora de seu domínio de aplicação. Por exemplo, um código de N -corpos resolve a dinâmica gravitacional de milhares de estrelas, mas pode não incluir algoritmos para evolução estelar ou dinâmica do gás entre as estrelas. O posterior desenvolvimento dessas características necessita um conjunto completamente diferente de *expertises* que o desenvolvedor do código gravitacional original pode não possuir. Isto representa uma barreira para o contínuo desenvolvimento de códigos monolíticos, dada a tendência atual em que a modelagem numérica de cenários realísticos envolvendo interações físicas complexas está cada vez mais suplantando as soluções de problemas idealizados de décadas atrás. A filosofia utilizada, p. ex., no *framework* AMUSE tem a vantagem de que cada desenvolvedor pode focar na construção de códigos cada vez mais robustos e eficientes com base na sua própria área de *expertise*. O problema aqui aparece quando do momento em que necessita-se o acoplamento dos diferentes domínios astrofísicos para o estudo de um sistema em particular. Por exemplo, certos fenômenos como i) gravitação com correções relativísticas, ii) gravitação e hidrodinâmica em regiões de alta densidade, ou iii) magneto-hidrodinâmica, etc., necessitam ser acoplados à um nível *fine-grained*¹, ou seja, o acoplamento deve existir à cada passo de tempo da solução a fim de evitar certos artefatos numéricos que apareceriam de outro modo. Em outros casos tais como, i) gravitação e evolução estelar, ii) gravitação e hidrodinâmica em regiões de baixa densidade, ou iii) gravitação e cosmologia, o acoplamento pode ser relaxado à um nível *coarse-grained*, ou seja umas poucas vezes durante a escala de tempo típica do sistema já contempla o comportamento esperado para sua evolução dinâmica. Assim, o problema com a estrutura modular do *framework* AMUSE é que este se presta à modelagem de sistemas cujos diferentes domínios astrofísicos podem ser acoplados apenas à um nível *coarse-grained*.

Em última instância, o problema do acoplamento auto-consistente de diferentes fenômenos astrofísicos nos códigos computacionais é uma questão de como as escalas de tempo envolvidas estão relacionadas. Um primeiro avanço no sentido de se modelar sistemas com fenômenos envolvendo diferentes escalas de tempo seguindo a filosofia mais tarde adotada pelo *framework* AMUSE surgiu com a criação de um método híbrido nomeado BRIDGE e primeiramente

¹ Não confundir com as terminologias *fine-* e *coarse-grained* utilizadas no contexto da Mecânica Estatística.

introduzido por [Fujii et al. \(2007\)](#). BRIDGE é baseado numa extensão de segunda ordem do esquema *Mixed Variable Symplectic* (MVS), desenvolvido no contexto de integrações à longo prazo de sistemas planetários (ver, p. ex., [Wisdom and Holman, 1991](#)). Em sua versão clássica, o método BRIDGE tornou possível o acoplamento de um código direto $\mathcal{O}(N^2)$ com um código em árvore $\mathcal{O}(N \log(N))$, resultando em um único código composto capaz de lidar com a coevolução de sistemas colisionais e não-colisionais auto-consistentemente. A ideia por trás do método BRIDGE é bastante poderosa, pois permite o acoplamento de diferentes códigos especializados sem a necessidade de modificar cada um deles individualmente. Variações do método BRIDGE tem sido usadas extensivamente para o acoplamento de códigos no *framework* AMUSE. Contudo, isto tem sido feito apenas à um nível *coarse-grained*.

A inclusão de outros fenômenos que necessitem de um acoplamento a um nível *fine-grained*, a modelagem de sistemas mais complexos e potencialmente mais interessantes do ponto de vista astrofísico, necessita assim o que nós chamamos de BRIDGE-hierárquico, onde o método BRIDGE original (ou alguma variação deste), possa ser aplicado de maneira hierárquica de modo a contemplar as diferentes escalas de tempo envolvidas nos diferentes domínios astrofísicos. Neste trabalho, nós atacamos este problema sob a ótica da evolução de sistemas gravitacionais. Contudo, nós mantemos nossos desenvolvimentos teóricos gerais o suficiente para que o método construído aqui seja aplicável à qualquer sistema físico cuja evolução temporal seja governada pela eq. 2.23. Assim, baseado na ideia por trás do método BRIDGE, nós estabelecemos uma família de 48 novos mapas simpléticos para evolução temporal de sistemas Hamiltonianos, além de um mapa específico à sistemas gravitacionais.

1.2 ORGANIZAÇÃO DA TESE

Esta tese é dividida em três partes: I - Introdução, II - Metodologia e Resultados e III - Conclusões.

A parte I contém este Capítulo de Introdução em que nós apresentamos uma breve contextualização e justificativa ao tópico abordado neste trabalho. A parte II é composta por três Capítulos: no Cap. 2 nós descrevemos alguns conceitos básicos e apresentamos a formulação teórica que fundamenta os desenvolvimentos numéricos e teóricos dos Capítulos seguintes; o Cap. 3 é devotado à descrição do nosso código tupan, oferecendo vários detalhes de implementação bem como alguns exemplos de uso prático; no Cap. 4 nós apresentamos nossos principais desenvolvimentos teóricos, estabelecendo uma família de 48 novos mapas simpléticos para evolução temporal de sistemas Hamiltonianos, além de um mapa específico à sistemas gravitacionais de N -corpos, baseado numa decomposição Hamiltoniana em termos de N^2 problemas de Kepler, e discutimos alguns testes de validação numérica. Finalmente na parte III nós apresentamos no Cap. 5 as nossas conclusões e perspectivas.

Parte II

METODOLOGIA E RESULTADOS

It is impossible to explain honestly the beauties of the laws of nature without some deep understanding of mathematics.

— Richard Feynman

2

FORMULAÇÃO TEÓRICA

O problema gravitacional de N -corpos consiste em determinar o movimento de um sistema de massas puntuais sob influência da força gravitacional atuando entre cada par de partículas. Sistemas astrofísicos típicos estudados por meio de simulações de N -corpos incluem o Sol e os oito planetas ($\sim 10^1$ corpos), aglomerados de estrelas ($\sim 10^3 - 10^6$ corpos), galáxias ($\sim 10^{11}$ corpos), ou mesmo o universo em escalas cosmológicas ($\sim 10^{22}$ corpos).

Como é sabido desde os primórdios da teoria da gravitação Newtoniana o problema de N -corpos admite uma solução exata somente nos casos em que $N = 1$ (partícula livre) ou $N = 2$ (duas partículas interagentes). Para $N \geq 3$ o que se sabe é que, na ausência de forças externas, existem dez integrais de movimento: a energia (E), o centro de massa (\mathbf{R}), o momentum linear (\mathbf{P}) e o momentum angular (\mathbf{J}) do sistema. Matematicamente, tais grandezas são definidas como segue:

$$E = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} - \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N \frac{Gm_i m_j}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (2.1)$$

$$\mathbf{R} = \frac{1}{M} \sum_{i=1}^N m_i \mathbf{r}_i, \quad (2.2)$$

$$\mathbf{P} = \sum_{i=1}^N m_i \mathbf{v}_i, \quad (2.3)$$

$$\mathbf{J} = \sum_{i=1}^N \mathbf{r}_i \times \mathbf{p}_i, \quad (2.4)$$

onde \mathbf{r}_i , \mathbf{v}_i e $\mathbf{p}_i = m_i \mathbf{v}_i$ são, respectivamente, a posição, a velocidade e o momentum linear da partícula de massa m_i , $M = \sum_{i=1}^N m_i$ é a massa total do sistema e G é a constante gravitacional Newtoniana. Os dois termos na eq. 2.1 represen-

tam, respectivamente, a energia cinética (T) e a energia potencial (U) do sistema. Naturalmente, na presença de uma força externa, \mathbf{f}_{ext} , termos de correção,

$$\dot{E}_{corr} = - \sum_{i=1}^N \mathbf{v}_i \cdot \mathbf{f}_{ext,i}, \quad (2.5)$$

$$\dot{\mathbf{R}}_{corr} = + \frac{1}{M} \mathbf{P}_{corr}, \quad (2.6)$$

$$\dot{\mathbf{P}}_{corr} = - \sum_{i=1}^N \mathbf{f}_{ext,i}, \quad (2.7)$$

$$\dot{\mathbf{J}}_{corr} = - \sum_{i=1}^N \mathbf{r}_i \times \mathbf{f}_{ext,i}, \quad (2.8)$$

devem ser levados em conta de modo a preservar as integrais de movimento:

$$E \rightarrow E + E_{corr}, \quad (2.9)$$

$$\mathbf{R} \rightarrow \mathbf{R} + \mathbf{R}_{corr}, \quad (2.10)$$

$$\mathbf{P} \rightarrow \mathbf{P} + \mathbf{P}_{corr}, \quad (2.11)$$

$$\mathbf{J} \rightarrow \mathbf{J} + \mathbf{J}_{corr}. \quad (2.12)$$

Para $N = 3$ é possível obter algumas soluções em situações bastante restritas, ou onde há algum tipo de simetria que permita uma redução do problema à casos solúveis. No caso geral com $N \geq 3$ somente soluções numéricas são possíveis através de simulações de N -corpos. No presente trabalho nós consideramos simulações diretas de N -corpos, o que significa que todos corpos no sistema são tratados como partículas adimensionais descritas por massa, posição e velocidade. Com isso, o objetivo do método de N -corpos resume-se ao cálculo da trajetória de cada partícula no sistema sob influência da interação gravitacional.

Simulações de N -corpos têm sido uma ferramenta essencial para o entendimento teórico de sistemas astrofísicos auto-gravitantes desde os trabalhos pioneiros de [von Hoerner \(1960\)](#), [Aarseth \(1963\)](#) e [van Albada \(1968\)](#). Antes de apresentar na seção 2.2 uma descrição formal para o método, introduziremos a seguir alguns conceitos preliminares. Sua implementação numérica será discutida no Cap. 3 e os novos métodos e algoritmos desenvolvidos pelo autor serão apresentados no Cap. 4.

2.1 CONCEITOS BÁSICOS

Em um sistema auto-gravitante existem duas escalas de tempo fundamentais: o *crossing time* e o *half-mass relaxation time*. O *crossing time*, t_{cr} , é a mais intuitiva delas, e é também a escala no qual o sistema estabelece o equilíbrio virial, $2T + U = 0$. Para um sistema em equilíbrio dinâmico t_{cr} é definido como

$$t_{cr} = \frac{2R_V}{\sigma}, \quad (2.13)$$

onde $R_V = -GM^2/2U$ é o raio virial e σ é a dispersão de velocidades. Em um estado de equilíbrio virial, $\sigma^2 \simeq GM/2R_V$, o que resulta em

$$\begin{aligned} t_{cr} &\simeq 2\sqrt{2} \left(\frac{R_V^3}{GM} \right)^{1/2}, \\ &\sim 4.7 \times 10^5 \text{ yr} \left(\frac{M}{10^6 M_\odot} \right)^{-1/2} \left(\frac{R_V}{10 \text{ pc}} \right)^{3/2}, \end{aligned} \quad (2.14)$$

onde, para referência, introduzimos algumas escalas astrofísicas relevantes. Alternativamente, uma definição formal do *crossing time* em termos de quantidades conservadas é dada por

$$t_{cr} = \frac{GM^{5/2}}{(-2E)^{3/2}}, \quad (2.15)$$

que pode ser derivada a partir da eq. 2.14, utilizando a definição do raio virial definido acima e notando-se que em equilíbrio virial, $E = U/2$.

Podemos agora definir as chamadas *N-body units*, que são unidades adimensionais tipicamente utilizadas em simulações de *N*-corpos. Seguindo [Heggie and Mathieu \(1986\)](#), nós definimos unidades de massa (u_m), comprimento (u_l) e tempo (u_t) pela condição de que a constante gravitacional, G , a massa total, M , e a energia total, E , do sistema em $t = 0$, expressa nessas unidades sejam respectivamente:

$$G = 1, \quad M = 1, \quad E = -\frac{1}{4}.$$

Isto resulta nas seguintes expressões para as unidades básicas:

$$u_m = M, \quad u_l = \frac{GM^2}{(-4E)}, \quad u_t = \frac{GM^{5/2}}{(-4E)^{3/2}}.$$

A escolha para E parece estranha, mas corresponde à um raio virial $R_V = 1$ para um sistema em equilíbrio virial. Nessas unidades temos ainda que $\sigma = \frac{1}{2}\sqrt{2}$ e $t_{cr} = 2\sqrt{2}$.

A segunda escala de tempo fundamental em um sistema auto-gravitante é o *half-mass relaxation time*, t_{rh} , sobre o qual encontros entre dois corpos transferem energia entre estrelas individuais, levando o sistema a uma evolução dinâmica em busca do equilíbrio térmico que, como se sabe, nunca é atingido por sistemas com interações de longo alcance. Dito de outro modo, é o tempo para que mudanças na velocidade quadrática média das estrelas, surgindo de deflexões por pequenos ângulos, tornem-se comparáveis a dispersão de velocidades inicial. Para propósitos práticos t_{rh} pode ser definido como

$$t_{rh} = n_{rh} t_{cr}, \quad (2.16)$$

onde t_{cr} é o *crossing time* definido anteriormente e n_{rh} é o número de vezes que uma estrela tem que cruzar o sistema, de modo que a mudança em sua velocidade seja $\Delta v^2 \sim v^2$. Embora não exista uma correspondente quantidade numericamente bem definida, uma aproximação comumente utilizada é (cfe. [Aarseth, 2003](#))

$$t_{rh} \simeq \frac{N}{22 \ln(\gamma N)} t_{cr}, \quad (2.17)$$

onde o fator γ foi determinado empiricamente como cerca de 0.11 – 0.15 (Hénon, 1975; Giersz and Heggie, 1994) para um sistema com partículas de massas iguais (ver também derivação analítica em Binney and Tremaine (2008), para o qual $\gamma = 1$).

A relaxação via encontros entre dois corpos é cada vez menos importante com o aumento do número de partículas, N , uma vez que o potencial torna-se cada vez mais suave. Isto está refletido na eq. 2.17 e permite-nos fazer uma distinção natural entre i) sistemas *não-colisionais*, cujo longo tempo de relaxação implica que eles não sofrem significativa evolução dinâmica interna durante a idade do universo, e ii) sistemas *colisionais*, que evoluem de forma significativa em menos que um tempo de Hubble. Galáxias enquadram-se na primeira categoria, uma vez que com $\sim 10^{11}$ estrelas distribuídas em um raio de ~ 10 kpc, possuem um tempo de relaxação de $\sim 10^{16}$ anos. Por outro lado, alguns núcleos galácticos, aglomerados globulares ou sistemas menos populosos caem na segunda categoria, com $t_{rh} \lesssim 10^{10}$ anos.

A evolução dinâmica interna pode levar sistemas estelares a estágios de alta densidade central, tornando propícias frequentes interações entre estrelas, sistemas binários, buracos negros e outros objetos astrofísicos eminentemente observáveis. Um conceito útil em dinâmica estelar é a definição da fricção dinâmica, primeiramente introduzida por Chandrasekhar (1943), quem elucidou a tendência sistemática de que um objeto de massa m_* seja desacelerado na direção de seu movimento como consequência direta da flutuação da força atuando sobre este, devido a variação na distribuição de vizinhos próximos. Se o objeto é muito mais massivo que as estrelas de campo de massa m_i , $m_i \ll m_* \ll M$, com M a massa total do sistema, então quando m_* se move com velocidade \mathbf{v}_* através das estrelas com distribuição de velocidades $f(\mathbf{v}_i)$, a deflexão das estrelas por m_* aumenta a densidade estelar, ρ , em sua retaguarda e, conseqüentemente, a sua atração gravitacional, causando a fricção dinâmica. A aceleração friccional sobre m_* pode ser escrita de forma geral como (e.g. Binney and Tremaine, 2008)

$$\frac{d\mathbf{v}_*}{dt} = -4\pi G^2 m_* m_i \ln \Lambda \int d^3\mathbf{v}_i f(\mathbf{v}_i) \frac{\mathbf{v}_* - \mathbf{v}_i}{|\mathbf{v}_* - \mathbf{v}_i|^3}, \quad (2.18)$$

onde Λ é o logaritmo de Coulomb, tipicamente $\sim (M/m_*)(r_*/R)$, sendo r_* o seu raio orbital e R o tamanho do sistema. Se $f(\mathbf{v}_i) = \sqrt{\frac{2}{\pi}} \frac{v_i^2}{\sigma^3} e^{-v_i^2/2\sigma^2}$, uma distribuição Maxwelliana com dispersão de velocidades σ , então a eq. 2.18 torna-se

$$\frac{d\mathbf{v}_*}{dt} = -\frac{4\pi G^2 m_* \rho \ln \Lambda}{v_*^3} \left[\operatorname{erf}(X) - \frac{2X}{\sqrt{\pi}} e^{-X^2} \right] \mathbf{v}_*, \quad (2.19)$$

onde $X \equiv v_*/(\sqrt{2}\sigma)$, erf é a função erro e $\rho = nm_i$, com $n = \int d^3\mathbf{v}_i f(\mathbf{v}_i)$ sendo a densidade numérica de estrelas de campo.

Aproximando um núcleo galáctico por um perfil de densidade de Jaffe (1983) em $r \rightarrow 0$ ou, mais simplesmente, por uma esfera isotérmica singular (e.g. Binney and Tremaine, 2008), para o qual $\rho(r) = \frac{\sigma^2}{2\pi G r^2}$ e assumindo $X = v_c/(\sqrt{2}\sigma) = 1$, então a força friccional, $F = m_* |d\mathbf{v}_c/dt|$, sobre m_* , como dada pela eq. 2.19, resulta em

$$F \simeq 0.428 \ln \Lambda \frac{Gm_*^2}{r^2}, \quad (2.20)$$

que é uma força tangencial e oposta à velocidade de m_* , causando uma perda de momentum angular orbital dada por

$$\frac{dL}{dt} = -Fr \simeq -0.428 \ln \Lambda \frac{Gm_*^2}{r}, \quad (2.21)$$

o que leva o objeto a espiralar em direção ao centro da galáxia. Uma vez que $L = r m_* v_c$, substituindo na eq. 2.21 e resolvendo a equação diferencial resultante para t , é possível mostrar (para detalhes, veja [Binney and Tremaine, 2008](#)) que o tempo de espiralamento por fricção dinâmica, i. e., tempo necessário para que m_* chegue ao centro galáctico pode ser escrito como

$$t_{fric} = \frac{1.17}{\ln \Lambda} \frac{M(r)}{m_*} t_{cr}, \quad (2.22)$$

onde $M(r) = v_c^2 r / G$ é a massa interna ao raio r e t_{cr} é o *crossing time* definido anteriormente. Para valores típicos de $\ln \Lambda \simeq 6$ e $m_* = 10^8 M_\odot$, a partir de um raio orbital ~ 5 kpc com $\sigma \simeq 200 \text{ km s}^{-1}$, resulta em $t_{fric} \sim 3$ Gyr, mostrando que objetos massivos tais como *Supermassive Black-Holes (SMBHs)* normalmente deveriam ser encontrados no centro das galáxias.

2.2 FORMULAÇÃO HAMILTONIANA

De um ponto de vista bastante geral, muitas equações da física, desde mecânica clássica, eletrodinâmica, mecânica estatística e mecânica quântica, possuem a forma genérica,

$$\frac{\partial \psi}{\partial t} = \mathcal{H} \psi, \quad (2.23)$$

onde $\psi = \psi(\mathbf{r}, \mathbf{p}) \equiv \psi(\mathbf{r}_1, \dots, \mathbf{r}_N, \mathbf{p}_1, \dots, \mathbf{p}_N)$ é uma quantidade dinâmica dependendo do conjunto completo das variáveis de fase. O operador \mathcal{H} , associado ao Hamiltoniano H , é dado pelo parêntese de Poisson,

$$\begin{aligned} \mathcal{H} &= \sum_{i=1}^N \left(\frac{\partial}{\partial \mathbf{r}_i} \frac{d\mathbf{r}_i}{dt} - \frac{\partial}{\partial \mathbf{p}_i} \frac{d\mathbf{p}_i}{dt} \right), \\ &= \sum_{i=1}^N \left(\frac{\partial}{\partial \mathbf{r}_i} \frac{\partial H}{\partial \mathbf{p}_i} - \frac{\partial}{\partial \mathbf{p}_i} \frac{\partial H}{\partial \mathbf{r}_i} \right), \\ &\equiv \{ \cdot, H \}. \end{aligned} \quad (2.24)$$

No caso de um sistema gravitacional com N-corpos, H é definido por

$$H(\mathbf{r}, \mathbf{p}) = \sum_{i=1}^N T_i + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N U_{ij}, \quad (2.25)$$

com

$$T_i = \frac{\mathbf{p}_i^2}{2m_i}, \quad (2.26)$$

$$U_{ij} = -G \frac{m_i m_j}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (2.27)$$

onde notamos que na ausência de perturbações externas ou dependência temporal explícita, eq. 2.25 é idêntica à energia total do sistema, definida pela eq. 2.1.

A partir da eq. 2.24 é possível identificar as equações de movimento de Hamilton,

$$\dot{\mathbf{r}}_i = +\frac{\partial H}{\partial \mathbf{p}_i}, \quad (2.28)$$

$$\dot{\mathbf{p}}_i = -\frac{\partial H}{\partial \mathbf{r}_i}, \quad (2.29)$$

que descrevem o comportamento do sistema, dada as condições iniciais m_i , \mathbf{r}_i e \mathbf{p}_i , $i = 1 \dots N$, em um dado tempo inicial t_0 . Eqs. 2.28 e 2.29 são, de fato, casos especiais da eq. 2.23, que admite a seguinte solução formal,

$$\psi(t + \tau) = e^{\tau \mathcal{H}} \psi(t), \quad (2.30)$$

embora esta não seja analiticamente integrável. Em muitos casos, contudo, o Hamiltoniano H pode ser decomposto em duas ou mais partes analiticamente integráveis. Por exemplo, o Hamiltoniano 2.25, é naturalmente decomposto em uma energia cinética, T , e uma energia potencial, U , tal que $H = T + U$. Assim, definindo-se operadores \mathcal{T} e \mathcal{U} associados a T e U , respectivamente, eq. 2.30 torna-se

$$\psi(t + \tau) = e^{\tau(T+U)} \psi(t). \quad (2.31)$$

Dado que $T = T(\mathbf{p})$ e $U = U(\mathbf{r})$ (assumindo que as massas são mantidas constantes), a aplicação de \mathcal{T} e \mathcal{U} independentemente às variáveis de fase podem ser resolvidas analiticamente como segue:

$$e^{\tau \mathcal{T}} \psi(\mathbf{r}_0, \mathbf{p}_0) = \psi(\mathbf{r}_0 + \tau \mathbf{v}_0, \mathbf{p}_0) \equiv \psi(\mathbf{r}_1, \mathbf{p}_0), \quad (2.32)$$

$$e^{\tau \mathcal{U}} \psi(\mathbf{r}_0, \mathbf{p}_0) = \psi(\mathbf{r}_0, \mathbf{p}_0 + \tau \mathbf{F}_0) \equiv \psi(\mathbf{r}_0, \mathbf{p}_1), \quad (2.33)$$

onde $\mathbf{v} = \mathbf{p}/m$ e $\mathbf{F} = -\nabla U$ é a força atuando sobre a partícula de massa m e os subscritos 0 e 1 indicam os valores iniciais e finais das variáveis de fase. Na eq. 2.32 o operador $e^{\tau \mathcal{T}}$ evolui as posições mantendo os momenta constantes, enquanto na eq. 2.33 o operador $e^{\tau \mathcal{U}}$ evolui os momenta mantendo as posições constantes. Uma vez que \mathcal{T} e \mathcal{U} em geral não comutam, segue que $e^{\tau(T+U)} \neq e^{\tau \mathcal{T}} e^{\tau \mathcal{U}} \neq e^{\tau \mathcal{U}} e^{\tau \mathcal{T}}$. Consequentemente, podemos construir somente uma aproximação ao verdadeiro operador de evolução temporal, $e^{\tau(T+U)}$. Isto usualmente é feito através do produto (composição) de operadores mais simples (leia-se analiticamente integráveis) tais como $e^{\tau \mathcal{T}}$ e $e^{\tau \mathcal{U}}$, via a identidade de Baker-Campbell-Hausdorff (BCH, veja Hairer et al., 2006),

$$e^{\tau \mathcal{U}} e^{\tau \mathcal{T}} = e^{\tau(\mathcal{T}+\mathcal{U}+\frac{\tau}{2}[\mathcal{U},\mathcal{T}]+\mathcal{O}(\tau^2))} \equiv e^{\tau \tilde{\mathcal{H}}}, \quad (2.34)$$

a partir do qual nota-se que

$$\tilde{\mathcal{H}} = \mathcal{H} + \mathcal{H}^{err}, \quad (2.35)$$

com

$$\mathcal{H}^{err} = \frac{\tau}{2}[\mathcal{U}, \mathcal{T}] + \mathcal{O}(\tau^2), \quad (2.36)$$

onde $[\cdot, \cdot]$ é um comutador. Podemos assim definir o mapa,

$$\phi_\tau \equiv e^{\tau\mathcal{U}} e^{\tau\mathcal{T}}, \quad (2.37)$$

que é simplético e cujo erro é dado pela eq. 2.36. Assim, podemos notar que a aplicação de ϕ_τ às variáveis de fase (cfe. eqs. 2.32 e 2.33),

$$\phi_\tau \psi(\mathbf{r}_0, \mathbf{p}_0) = e^{\tau\mathcal{U}} e^{\tau\mathcal{T}} \psi(\mathbf{r}_0, \mathbf{p}_0) = e^{\tau\mathcal{U}} \psi(\mathbf{r}_1, \mathbf{p}_0) = \psi(\mathbf{r}_1, \mathbf{p}_1), \quad (2.38)$$

evolui exatamente o Hamiltoniano modificado \tilde{H} , que por sua vez constitui uma aproximação em primeira ordem ao Hamiltoniano original do sistema, H .

Nós agora definimos o mapa auto-adjunto,

$$\phi_\tau^\dagger \equiv e^{\tau\mathcal{T}} e^{\tau\mathcal{U}}, \quad (2.39)$$

que possui os mesmos passos mas na ordem reversa que ϕ_τ , ou seja,

$$\phi_\tau^\dagger \psi(\mathbf{r}_0, \mathbf{p}_0) = e^{\tau\mathcal{T}} e^{\tau\mathcal{U}} \psi(\mathbf{r}_0, \mathbf{p}_0) = e^{\tau\mathcal{T}} \psi(\mathbf{r}_0, \mathbf{p}_1) = \psi(\mathbf{r}_1, \mathbf{p}_1). \quad (2.40)$$

ϕ_τ^\dagger possui propriedades similares à ϕ_τ , discutidas acima. Porém ambos constituem mapas simpléticos não reversíveis. Nós podemos garantir reversibilidade temporal com a construção de mapas simétricos. Dessa forma nós definimos

$$\phi_\tau^2 \equiv \phi_{\tau/2}^\dagger \circ \phi_{\tau/2} = e^{\frac{\tau}{2}\mathcal{T}} e^{\tau\mathcal{U}} e^{\frac{\tau}{2}\mathcal{T}}, \quad (2.41)$$

onde o símbolo \circ significa a composição do operador à sua esquerda com o operador à sua direita. A condição de reversibilidade para um mapa arbitrário φ_τ ,

$$\varphi_\tau \circ \varphi_{-\tau} = id, \quad (2.42)$$

é então satisfeita para $\varphi_\tau = \phi_\tau^2$, onde id é a matriz identidade. Note que o mesmo também vale para o mapa auto-adjunto de ϕ_τ^2 ,

$$(\phi_\tau^2)^\dagger \equiv \phi_{\tau/2} \circ \phi_{\tau/2}^\dagger = e^{\frac{\tau}{2}\mathcal{U}} e^{\tau\mathcal{T}} e^{\frac{\tau}{2}\mathcal{U}}. \quad (2.43)$$

A aplicação da identidade de BCH ao mapa ϕ_τ^2 resulta em $\tilde{\mathcal{H}}_2 = \mathcal{H} + \mathcal{H}_2^{err}$, com

$$\mathcal{H}_2^{err} = \tau^2 \left(\frac{1}{12} [\mathcal{U}, [\mathcal{U}, \mathcal{T}]] - \frac{1}{24} [\mathcal{T}, [\mathcal{T}, \mathcal{U}]] \right) + \mathcal{O}(\tau^4), \quad (2.44)$$

mostrando que ϕ_τ^2 constitui um mapa simplético aproximando o Hamiltoniano original em segunda ordem. Além disso, ϕ_τ^2 possui uma outra propriedade não compartilhada por ϕ_τ na forma de H . Por causa que ϕ_τ^2 é simétrico, $\tilde{\mathcal{H}}$ terá somente termos pares em τ (veja prova em Yoshida, 1990), especificamente,

$$\tilde{\mathcal{H}} = \mathcal{H} + \sum_{k=1}^{\infty} \mathcal{F}_k(\mathcal{T}, \mathcal{U}) \tau^{2k}, \quad (2.45)$$

onde \mathcal{F}_k são funções conhecidas dos operadores \mathcal{T} e \mathcal{U} , calculadas a partir da aplicação da identidade de BCH (veja Casas and Murua, 2009). Essa propriedade explica o fato de que mapas simpléticos e reversíveis podem somente ser de ordem par.

A construção de mapas simpléticos de ordem mais alta pode ser feita através do método de [Yoshida \(1990\)](#), que permite gerar um mapa de ordem, e. g., $2n + 2$, a partir da composição de mapas de ordem $2n$. Assim, um mapa de quarta ordem, ϕ_τ^4 , é obtido a partir de ϕ_τ^2 como segue,

$$\phi_\tau^4 = \phi_{z_0\tau}^2 \circ \phi_{z_1\tau}^2 \circ \phi_{z_0\tau}^2, \quad (2.46)$$

onde $z_0 = 1/(2 - 2^{1/3})$ e $z_1 = 1 - 2z_0$ ([Yoshida, 1990](#)). Em sexta ordem, temos

$$\begin{aligned} \phi_\tau^6 &= \phi_{z'_0\tau}^4 \circ \phi_{z'_1\tau}^4 \circ \phi_{z'_0\tau}^4, \\ &= \phi_{w_0\tau}^2 \circ \phi_{w_1\tau}^2 \circ \phi_{w_2\tau}^2 \circ \phi_{w_3\tau}^2 \circ \phi_{w_2\tau}^2 \circ \phi_{w_1\tau}^2 \circ \phi_{w_0\tau}^2, \end{aligned} \quad (2.47)$$

onde os w_i fornecidos por [Yoshida \(1990\)](#) com precisão de 15 casas decimais, são

$$w_0 \approx +0.784513610477557, \quad (2.48a)$$

$$w_1 \approx +0.235573213359358, \quad (2.48b)$$

$$w_2 \approx -1.177679984178871, \quad (2.48c)$$

$$w_3 = 1 - 2(w_0 + w_1 + w_2). \quad (2.48d)$$

Análogos das eqs. 2.46 e 2.47 também existem para os respectivos mapas auto-adjuntos $(\phi_\tau^4)^\dagger$ e $(\phi_\tau^6)^\dagger$. Uma compilação de todos os possíveis mapas até sexta ordem, resultando da composição direta dos operadores $e^{\tau\mathcal{T}}$ e $e^{\tau\mathcal{U}}$ pode ser encontrada em [Omelyan et al. \(2003\)](#), que utilizando uma generalização do método de [Yoshida \(1990\)](#), encontrou cerca de 50 novas composições, algumas delas com propriedades muito superiores aos mapas previamente conhecidos.

Como uma última observação, é importante salientar que, por construção, mapas simpléticos admitem apenas um intervalo de tempo, τ , constante. O motivo para tal restrição é que a simpleticidade do mapa só é mantida se \mathcal{H}^{err} for independente do tempo, ou seja, se $\tau \neq \tau(t)$, tal que a evolução do Hamiltoniano seja dada por, e. g., eq. 2.45. Quando isso não é possível, ou quando há necessidade de um intervalo de tempo, τ , variável com o tempo, \mathcal{H}^{err} não mais permanece limitado ao longo da transformação $\psi(\mathbf{r}_0, \mathbf{p}_0) \mapsto \psi(\mathbf{r}_1, \mathbf{p}_1)$, ocasionando erros seculares à evolução Hamiltoniana, ou, em outras palavras, tornando o mapa $\psi(\mathbf{r}_0, \mathbf{p}_0) \mapsto \psi(\mathbf{r}_1, \mathbf{p}_1)$ uma transformação não-simplética. Nós retornaremos a esse tópico no Cap. 4, quando discutirmos um algoritmo para superar tal restrição.

2.3 CORREÇÕES RELATIVÍSTICAS PARA INTERAÇÕES ENTRE SMBHs

SMBHs binários são uma das fontes de radiação gravitacional mais interessantes para os futuros detectores de ondas gravitacionais tais como o *Laser Interferometer Space Antenna* (LISA). Tais objetos, uma vez em regime orbital relativístico (distância relativa $d \lesssim 10^{-3}$ pc), coalescem sob forte emissão de ondas gravitacionais em menos que um tempo de Hubble. Como consequência das leis de conservação de energia e momentum, o SMBH resultante deste processo tipicamente recua na direção oposta àquela da emissão da radiação. As velocidades de recuo que podem alcançar valores tão altos quanto ~ 4000 km/s ([Campanelli](#)

et al., 2007), o que é suficiente para ejetar SMBHs do centro das galáxias. No contexto de simulações de N -corpos, tal fenômeno pode ser modelado utilizando correções relativísticas baseadas em uma aproximação *post-Newtonian* (PN) da *General Relativity* (GR). Outras abordagens se baseiam em métodos perturbativos, ou mesmo em relatividade numérica.

O problema do movimento e da radiação gravitacional na aproximação PN têm sido estudado por Lorentz and Droste (1917), e mais tarde por Einstein-Infeld-Hoffmann (Einstein et al., 1938), entre outros, e são baseados no limite de campo fraco no qual a métrica é próxima a métrica de Minkowski e na suposição de que a velocidade típica, v , é muito pequena comparada à velocidade da luz, c . O desvio da métrica plana pode ser caracterizada pelo potencial Newtoniano Φ , de modo que para um sistema binário, por exemplo, $\Phi/c^2 \sim v^2/c^2 \equiv \beta^2$, com a aproximação PN reduzindo-se às equações de Newton no limite apropriado, $\beta \rightarrow 0$. Posteriormente a estes primeiros trabalhos, numerosas investigações de ambas aproximações PNs e pós-Minkowskianas¹ tem aparecido, com expansões formais em ordens de $1/c$ (veja, p. ex., Hawking and Israel, 1987). Termos de ordem $1PN$ ($\sim 1/c^2$) são rotineiramente levados em conta para descrever a dinâmica do sistema solar (Newhall et al., 1983), enquanto termos de ordem $2.5PN$ ($\sim 1/c^5$), tem sido verificados experimentalmente, pela observação da aceleração secular no movimento orbital do pulsar binário PSR 1913+16 (Taylor et al., 1979; Taylor and Weisberg, 1982; Taylor, 1993).

A partir de 1990 dois grupos embarcaram na derivação das equações de movimento a um nível $3PN$ ($\sim 1/c^6$). Um grupo usou o formalismo Hamiltoniano de Arnowitt-Deser-Misner (ADM) da GR (Jaranowski and Schäfer, 1998, 1999, 2001; Damour et al., 2000, 2001a,b, 2002), trabalhando em um correspondente sistema de coordenadas do tipo-ADM. Um outro grupo usou diretamente uma iteração PN das equações de movimento em coordenadas harmônicas (equivalente a uma formulação Lagrangeana) (Blanchet and Faye, 2000a,b, 2001a,b; de Andrade et al., 2001; Blanchet and Iyer, 2003). O resultado dessas duas abordagens mostrou-se fisicamente equivalente (de Andrade et al., 2001; Damour et al., 2001b, 2002), à menos de um parâmetro de ambiguidade presente em ambas as formulações e relacionado ao método de regularização usado para tratar as divergências de auto-campo de partículas pontuais em ordem $3PN$. Este parâmetro de ambiguidade foi mais tarde univocamente fixado graças a uma regularização dimensional do auto-campo (Damour et al., 2001c; Blanchet et al., 2004). Isso está também em completa concordância com os trabalhos de Itoh and Futamase (2003); Itoh (2004a,b), que derivaram independentemente as equações de movimento em ordem $3PN$ sem ambiguidade, usando o calibre harmônico numa abordagem baseada em “integrais de superfície” e sem a necessidade de nenhuma espécie de regularização de auto-campo. Finalmente, os termos $3.5PN$, que constituem a correção de primeira ordem aos termos $2.5PN$ da força de reação a radiação, foram calculados por Iyer and Will (1993, 1995); Jaranowski and Schäfer (1997); Pati and Will (2002); Königsdörffer et al. (2003); Nissanke and Blanchet (2005); Itoh (2009).

¹ Aproximações pós-Minkowskianas diferenciam-se das PNs por não assumir $v \ll c$, de modo que no limite apropriado a aproximação pós-Minkowskiana reduz-se às equações da Relatividade Especial.

Os trabalhos acima derivam as equações de movimento PNs para o caso de dois corpos i e j ($i \neq j$) interagindo gravitacionalmente em regime relativístico. Tais equações podem ser escritas como:

$$m_i \dot{\mathbf{v}}_i = - \frac{Gm_i m_j}{r_{ij}^2} [(1 + \mathcal{A}) \mathbf{n}_{ij} + \mathcal{B} \mathbf{v}_{ij}] , \quad (2.49)$$

onde $\mathbf{n}_{ij} = \mathbf{r}_{ij}/r_{ij}$ e os outros símbolos tem o significado usual. Os termos \mathcal{A} e \mathcal{B} são funções de m_i, m_j, r_i, r_j, v_i e v_j , contendo as diferentes ordens PNs (veja Itoh, 2004a, 2009). \mathcal{A} e \mathcal{B} podem ainda ser decompostos nas diferentes ordens PNs como segue:

$$\begin{aligned} \mathcal{A} &= \frac{1}{c^2} \mathcal{A}_{1PN} + \frac{1}{c^4} \mathcal{A}_{2PN} + \frac{1}{c^5} \mathcal{A}_{2.5PN} + \frac{1}{c^6} \mathcal{A}_{3PN} + \frac{1}{c^7} \mathcal{A}_{3.5PN} \\ &+ \mathcal{O}\left(\frac{1}{c^8}\right) , \end{aligned} \quad (2.50a)$$

$$\begin{aligned} \mathcal{B} &= \frac{1}{c^2} \mathcal{B}_{1PN} + \frac{1}{c^4} \mathcal{B}_{2PN} + \frac{1}{c^5} \mathcal{B}_{2.5PN} + \frac{1}{c^6} \mathcal{B}_{3PN} + \frac{1}{c^7} \mathcal{B}_{3.5PN} \\ &+ \mathcal{O}\left(\frac{1}{c^8}\right) , \end{aligned} \quad (2.50b)$$

onde termos de ordens inteiras, tais como $1PN$, $2PN$ e $3PN$, são responsáveis por causar o avanço do periastro do sistema binário e são conservativos, no sentido de que não causam uma variação secular do semi-eixo maior orbital. Termos de ordens semi-inteiras, tais como $2.5PN$ e $3.5PN$, por outro lado, são responsáveis pela dissipação da energia orbital na forma de ondas gravitacionais, sendo portanto aqueles que geram a força de reação à radiação. Explicitamente, as eqs. 2.50 contêm um total de 213 termos (veja Ap. A). Efeitos de spin devem aparecer como um termo de acoplamento spin-orbita em ordem $1.5PN$ e acoplamento spin-spin em ordem $2PN$ e também nos termos de reação a radiação, ordens $2.5PN$ e $3.5PN$ (Blanchet, 2006). No que segue, efeitos de spin não são considerados no presente trabalho.

Uma importante classe problemas de N -corpos onde a GR deve ser levada em conta envolve a evolução de um sistema estelar em torno de um SMBH, ou a interação de múltiplos SMBHs. Em Gonçalves Ferrari (2010) nós propomos a seguinte aproximação para as equações de movimento PNs no problema relativístico de N -corpos:

$$m_i \dot{\mathbf{v}}_i = - \sum_{j \neq i}^N \frac{Gm_i m_j}{r_{ij}^2} [(1 + \mathcal{A}) \mathbf{n}_{ij} + \mathcal{B} \mathbf{v}_{ij}] . \quad (2.51)$$

Na eq. 2.51 a aceleração do objeto i é dada (como no caso puramente Newtoniano) pela superposição das contribuições de todos os outros objetos j no sistema. Aqui, vê-se que eq. 2.51 reduz-se ao caso Newtoniano quando $\beta \rightarrow 0$. Contudo, a validade do princípio de superposição num regime relativístico é questionável, visto que a GR é uma teoria essencialmente não-linear. Dessa forma, espera-se que a eq. 2.51 contenha ainda termos adicionais que levem em conta efeitos não-lineares devido à interação relativística entre múltiplos corpos quando $N > 2$.

Equações PN s de N -corpos foram primeiramente introduzidas por [Lorentz and Droste \(1917\)](#) e são comumente chamadas de equações de movimento de Einstein–Infeld–Hoffmann ([Einstein et al., 1938](#)). Estas equações são implícitas e são conhecidas somente em ordem 1PN ,

$$\begin{aligned}
\mathbf{a}_i = & - \sum_{j \neq i}^N \frac{Gm_j}{r_{ij}^2} \mathbf{n}_{ij} \\
& + \frac{1}{c^2} \sum_{j \neq i}^N \frac{Gm_j}{r_{ij}^2} \mathbf{n}_{ij} \left[-\mathbf{v}_i^2 - 2\mathbf{v}_j^2 + 4(\mathbf{v}_i \cdot \mathbf{v}_j) + \frac{3}{2}(\mathbf{n}_{ij} \cdot \mathbf{v}_j)^2 \right. \\
& \left. + 4 \sum_{k \neq i}^N \frac{Gm_k}{r_{ik}} + \sum_{k \neq j}^N \frac{Gm_k}{r_{jk}} + \frac{1}{2}(\mathbf{r}_{ij} \cdot \mathbf{a}_j) \right] \\
& + \frac{1}{c^2} \sum_{j \neq i}^N \frac{Gm_j}{r_{ij}^2} [\mathbf{n}_{ij} \cdot (4\mathbf{v}_i - 3\mathbf{v}_j)] \mathbf{v}_{ij} \\
& + \frac{7}{2c^2} \sum_{j \neq i}^N \frac{Gm_j}{r_{ij}} \mathbf{a}_j + O(c^{-4}), \tag{2.52}
\end{aligned}$$

onde nota-se que a aceleração do objeto i depende da aceleração do objeto j . Comparando-se eq. 2.52 com a forma explícita da eq. 2.51 nota-se que os termos não incluídos na eq. 2.51 são aqueles contendo somas em k e termos dependentes da aceleração. Para ordens acima de 1PN , o único trabalho de conhecimento do autor, tratando o problema de N -corpos na formulação PN é dado em [Chu \(2009\)](#), embora este não apresente uma forma explícita das equações PN .

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one— and preferably only one —obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea — let's do more of those!

— The Zen of Python, by Tim Peters

3

IMPLEMENTAÇÃO NUMÉRICA

Neste Capítulo nós apresentamos `tupan`¹, um novo código para simulações de N -corpos. `tupan` é especialmente desenvolvido com foco em sistemas colisionais, tais como aglomerados de estrelas e núcleos galácticos com `SMBHs`. `tupan` implementa uma família de métodos de integração simplética, com diferentes esquemas de passo de tempo, para evolucionar as partículas durante uma simulação. O código alcança alta performance por meio de uma interface em `OpenCL`², para fácil acesso ao poder computacional de `GPU`s ou multi-core `CPU`s. A estrutura do código é modular e permite agregar diferentes fenômenos físicos, tais como evolução estelar e hidrodinâmica, em adição a (atualmente implementadas) dinâmica gravitacional e equações de movimento `PNs` para interações entre `SMBHs`. `tupan` encontra-se em pleno desenvolvimento e atualmente pode ser usado para evoluir aglomerados de estrelas com uma *Initial Mass Function* (`IMF`) ou usando partículas com massas iguais, possivelmente na presença de múltiplos `SMBHs`, que são sistemas de particular interesse Astrofísico. Nas seguintes seções nós discutiremos os principais detalhes de sua implementação.

¹ <https://github.com/ggf84/tupan>.

² <http://www.khronos.org/opencl/>.

3.1 POR QUE UM NOVO CÓDIGO DE N -CORPOS?

Nos últimos 55 anos o desenvolvimento de códigos para simulações de N -corpos tem se deparado com sérios desafios no que diz respeito a faixa de escalas de tempo e comprimento inerentes em todos os sistemas estelares. As mais curtas escalas de tempo e comprimento de interesse na evolução de aglomerados globulares e núcleos galácticos são postas por seus subsistemas mais compactos. Por exemplo, um par de estrelas de neutrons sofrendo um encontro de raspão, possui uma escala de tempo de interação de tipicamente uns poucos mili-segundos que, quando comparado à idade de um aglomerado globular, $\sim 10^{10}$ anos, resulta em uma discrepância em escalas de tempo de cerca de 20 ordens de magnitude! Um problema similar ocorre quando comparamos escalas de comprimento. O diâmetro de uma estrela de neutrons, ~ 10 km, quando expresso em unidades do raio de maré de um aglomerado globular, tipicamente ~ 100 pc, é da ordem de 10^{-15} . Levando-se em conta que em computação, números de ponto flutuante possuem representação finita (64-bits, com mantissa de ~ 15 dígitos significantes) é fácil entender o motivo pelo qual uma simulação pode perder completamente seu significado físico, caso sistemas compactos estejam presentes ou sejam naturalmente criados como consequência de interações prévias entre as partículas. Nestas circunstâncias, o método de integração numérica utilizado deve ser robusto o suficiente para mitigar o efeito do *round-off error* causado pelas grandes diferenças em escalas de tempo e comprimento, além de ser capaz de auto adaptar-se, a fim de capturar a dinâmica correta de tais sub-sistemas e manter a conservação das integrais de movimento do sistema como um todo, *sem perder eficiência computacional*.

A modelagem numérica de aglomerados globulares ou núcleos galácticos — estrela por estrela — pareceria desesperançosa não fosse pela ingenuidade e persistência de Sverre Aarseth, que dedicou a sua vida como pesquisador para fornecer uma série métodos (e. g. [Aarseth, 2003](#)) e implementações eficientes com o qual esses e outros problemas semelhantemente intransponíveis puderam ser atacados. A resposta de Aarseth para os problemas descritos acima pode ser resumida em dois conceitos-chave: passos de tempo individuais e regularização de coordenadas. Estas melhorias algorítmicas, embora não representem uma solução definitiva, fizeram mais pela velocidade dos cálculos do que décadas de avanço tecnológico em *hardware*. Contudo, algumas das técnicas introduzidas por Aarseth são de difícil paralelização, e não há ainda uma “receita” universal de como sanar todas as dificuldades numéricas a que o problema de N -corpos está sujeito. Dentre as possíveis razões para isso está o fato de que, apesar de conceitualmente simples, na prática a construção de um código de N -corpos é algo bastante complexo, tendo em vista que exige do programador uma combinação de conhecimentos avançados sobre design de software, algoritmos sofisticados, detalhes do funcionamento do hardware, técnicas de paralelização, etc., além de possuir uma visão clara sobre o problema físico em questão. Talvez por isso, novos códigos de N -corpos continuam sendo desenvolvidos por alguns grupos como tentativa de implementar novas abordagens a fim de lidar com tais dificuldades de maneira mais robusta.

Tradicionalmente, tais códigos têm sido desenvolvidos em linguagens de programação como C, C++ ou Fortran, em que quase sempre é possível obter

performance computacional razoável utilizando desde máquinas relativamente modestas, até sistemas massivamente paralelos. Entretanto, o fato de que tais linguagens em geral não favorecem o programador, acaba obrigando-o a escrever muito mais linhas de código, tornando o processo de desenvolvimento lento e muito mais propenso à erros de programação (*bugs*). Como exemplo, Hut (2010) estima que um desenvolvedor experiente leva, tipicamente, cerca de 10–20 anos para construir um código de N -corpos considerado *estado-da-arte*. Além disso, tendo em vista que físicos em geral não possuem formação adequada em técnicas de programação, frequentemente são escritos códigos bastante confusos e mal documentados, o que muitas vezes acaba prevenindo sua adoção por usuários menos experientes, ou até mesmo impossibilitando sua manutenção a longo prazo e sua extensão à novas tecnologias em *software* e *hardware*.

Nosso objetivo com tupan é o de criar um ambiente de *software* totalmente integrado para realização e análise dos resultados de simulações Astrofísicas de N -corpos. A motivação original para o seu desenvolvimento surgiu da necessidade de modelar o recuo gravitacional de SMBHs em um núcleo galáctico devido à emissão anisotrópica de ondas gravitacionais, tópico que foi discutido em Gonçalves Ferrari (2010). Naquele trabalho, uma primeira versão do código foi escrita em linguagem C, implementando dinâmica gravitacional básica e equações de movimento PNs, eq. 2.51. Porém, por se tratar de uma implementação bastante simplificada, nós encontramos muitos dos problemas descritos nos parágrafos acima, o que acabou por limitar as simulações apresentadas em Gonçalves Ferrari (2010) à um número de partículas muito menor do que seria necessário para uma modelagem realística de um núcleo galáctico. Tendo em vista que, para o conhecimento do autor, tupan é ainda hoje (2015) um dos poucos códigos³ disponíveis atualmente que implementa equações PNs até ordem $3.5PN$, nós decidimos então investigar em mais detalhes meios para torná-lo mais robusto.

Nós começamos por reavaliar a maneira pela qual tais códigos têm sido tradicionalmente desenvolvidos e propomos algumas mudanças de paradigma, novas abordagens, novos algoritmos e formas para acelerar seu desenvolvimento. Como tal, nós projetamos tupan de modo que este seja de fácil utilização e/ou modificação, extensível à novas tecnologias em *software/hardware*, eficiente e cientificamente competitivo. Com isso, a fim de minimizar o tempo de desenvolvimento, nossa primeira decisão foi com relação a escolha da linguagem de programação. Nós pensamos que *o tempo de desenvolvimento do código é mais importante do que o tempo que este leva para ser executado pela máquina*. Assim, nós optamos por uma linguagem que favoreça o programador, a legibilidade e a facilidade de manutenção do código. Desta forma, tupan foi completamente reescrito em Python⁴, uma linguagem madura e que recentemente vem tendo grande aceitação dentro do campo da computação científica, tendo disponível um rico ecossistema de *software*. Uma suposta desvantagem neste caso deve-se ao fato de que linguagens de altíssimo nível como Python frequentemente são consideradas relativamente lentas para problemas que demandam grande po-

³ Existem alguns outros códigos que implementam equações PNs até ordem $2.5PN$, ou versões simplificadas dos termos de ordem $3PN$.

⁴ <http://www.python.org/>.

der computacional. Isso, porém, não representa um problema em nossa implementação, pois tupan possui suporte à GPUs e multi-core CPUs, através de uma interface em OpenCL, tornando-o essencialmente um código tão eficiente quanto um código similar escrito em linguagens como C, C++ ou Fortran. Dessa forma, nós superamos a enorme barreira em tempo de desenvolvimento, sem perder eficiência computacional por conta da escolha da linguagem de programação. A primeira linha de código desde sua completa reescrita em Python, data⁵ de 14 de outubro de 2010. O *design* inicial do código foi projetado e implementado dentro do primeiro ano de desenvolvimento. Durante o ano de 2012 o código sofreu uma reestruturação com a implementação de uma família de integradores simpléticos com suporte à diferentes esquemas de passos de tempo. Estes foram posteriormente generalizados até sexta ordem e reimplementados de maneira mais simples e robusta em 2013. Atualmente tupan implementa vários algoritmos e métodos considerados *estado-da-arte*, bem como métodos desenvolvidos pelo autor (Gonçalves Ferrari et al., 2014).

3.2 TUPAN: UMA VISÃO GERAL DO CÓDIGO

tupan foi projetado de maneira modular. Sua construção foi baseada no princípio *Keep It Simple, Stupid* (KISS)⁶, de modo a manter o código simples e facilitar seu uso e modificação pelo usuário final. Isto também facilita sua manutenção e extensibilidade à novas tecnologias em *software* e *hardware*. A Fig. 3.1 mostra alguns dos módulos que compõem tupan e que serão detalhados na seção 3.3.

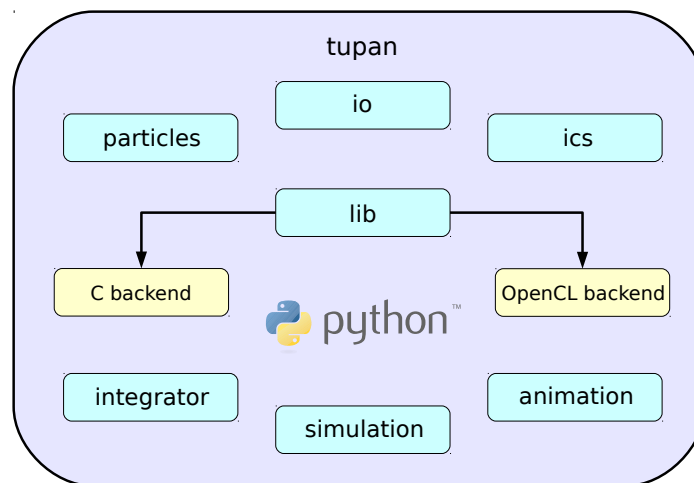


Figura 3.1: Alguns dos módulos que compõem tupan.

Atualmente tupan conta com pouco mais de 7600 linhas de código (descontando-se linhas em branco e comentários), distribuídas em 73 arquivos fonte e de configuração. Cerca de 48.4% desse total está escrito em Python, onde se fez uso extensivo do pacote para computação científica, Numpy⁷. Sub-rotinas

⁵ <https://github.com/ggf84/tupan>.

⁶ http://en.wikipedia.org/wiki/KISS_principle.

⁷ <http://www.numpy.org/>.

que escalam com $\mathcal{O}(N^2)$ (tipicamente, cálculo da aceleração ou potencial gravitacional) compõe a parte do código que é diretamente escrito em C/OpenCL, totalizando 7.7% (C), 13.3% (OpenCL) e 28.3% em *header files* compartilhados por ambos os *backends*. Os restantes $\sim 2\%$ constituem arquivos de configuração e *scripts* de instalação.

Futuramente outros módulos deverão ser incluídos no código. Por exemplo, simulações que levem em conta efeitos hidrodinâmicos bem como evolução estelar em sistemas astrofísicos serão possíveis com a inclusão de módulos implementando *Smoothed Particle Hydrodynamics* (SPH) e equações de evolução da estrutura estelar.

3.3 DETALHES DE IMPLEMENTAÇÃO

Nesta seção nós descrevemos brevemente os principais módulos que compõem tupan. Nós enfatizamos que o código encontra-se em pleno desenvolvimento e algumas das características descritas aqui podem sofrer modificações ou não estar completamente prontas para uso.

3.3.1 Módulo *io*

O módulo *io* implementa interfaces para *input/output* de dados gerados por tupan na criação de condições iniciais ou durante uma simulação. Estes são armazenados de maneira flexível e eficiente em formato HDF5⁸. tupan atualmente suporta dois diferentes *layouts* para o armazenamento dos dados: i) (série de *snapshots*, e ii) *Particle-Based Sampling Scheme* (PBaSS, Faber et al., 2010). PBaSS é, e. g., cerca de 10 vezes mais eficiente em termos de espaço utilizado em disco, porém requer sincronização das propriedades de diferentes partículas antes de serem usados para análise. Isto é feito através de uma conversão para uma série de *snapshots*⁹ em uma determinada faixa temporal. Métodos para conversão entre os dois *layouts* estão em fase de implementação.

3.3.2 Módulo *particles*

tupan possui suporte à diferentes tipos de partículas. O módulo *particles* mantém a implementação de diferentes tipos de partículas presentes em sistemas Astrofísicos. Atualmente somente os tipos Star e Blackhole estão completamente implementados e prontos para uso. Tipos como Gas, Darkmatter, etc. estão em fase de planejamento e serão incluídos futuramente. Em tupan cada um dos tipos de partículas são implementados como classes definindo seus atributos e métodos específicos. Estes, por sua vez, são combinados na classe *ParticleSystem*, que implementa uma interface agnóstica ao tipo de partícula, fornecendo métodos comuns ao sistema de partículas como um todo, tais como métodos para o cálculo do potencial gravitacional ou do centro de massa, por exemplo.

⁸ <http://www.hdfgroup.org/>.

⁹ Tais *snapshots* podem ser descartados após a análise ter sido completada.

3.3.3 Módulo *ics*

O módulo *ics* mantém a implementação de métodos para construção de condições iniciais para simulações. *tupan* possui suporte para a construção de sistemas com poucas partículas, como sistemas binários, alguns sistemas de 3- e 4-corpos com soluções conhecidas e o sistema solar (Sol, os 8 planetas e Plutão). Sistemas esféricos com N -corpos seguindo um perfil de densidade de Plummer,

$$\rho(r) \propto \frac{1}{(r^2 + 1)^{5/2}}, \quad (3.1)$$

em equilíbrio virial, podem também ser criados. O procedimento para construção desse modelo é inspirado no método descrito em [Aarseth \(2003\)](#). Futuramente, suporte para criação de modelos esféricos a partir do perfil generalizado de [Zhao \(1996, 1997\)](#), bem como modelos de discos serão adicionados.

É possível também usar vários tipos de **IMFs** para distribuição de massas das partículas. *tupan* suporta as seguintes **IMFs** (o significado dos parâmetros e uma discussão da teoria subjacente pode ser encontrado nas referências dadas):

EQUALMASS define uma **IMF** para partículas com massas iguais dada por

$$\psi(m) \propto 1, \quad (3.2)$$

SALPETER1955 define uma **IMF** tipo [Salpeter \(1955\)](#) dada por

$$\psi_{Sal}(m) \propto m^{-\Gamma}, \quad (3.3)$$

com $m_l < m < m_u$, onde $\Gamma = 1.35$, $m_l = 0.4 M_\odot$ e $m_u = 120.0 M_\odot$.

PADOAN2007 define uma **IMF** tipo [Padoan et al. \(2007\)](#) dada por

$$\psi_{Pad}(m) \propto m^{-\Gamma} \left\{ 1 + \operatorname{erf} \left[\frac{4 \log(m/m_{ch}) + \sigma^2}{2\sqrt{2}\sigma} \right] \right\}, \quad (3.4)$$

com $m_l < m < m_u$, onde $\Gamma = 1.4$, $m_{ch} = 1.0 M_\odot$, $\sigma = 1.8$, $m_l = 0.004 M_\odot$ e $m_u = 120.0 M_\odot$.

PARRAVANO2011 define uma **IMF** tipo [Parravano et al. \(2011\)](#) dada por

$$\psi_{Par}(m) \propto m^{-\Gamma} \left\{ 1 - \exp \left[-(m/m_{ch})^{\gamma+\Gamma} \right] \right\}, \quad (3.5)$$

com $m_l < m < m_u$, onde $\Gamma = 1.35$, $m_{ch} = 0.35 M_\odot$, $\gamma = 0.51$, $m_l = 0.004 M_\odot$ e $m_u = 120.0 M_\odot$.

Nós decidimos implementar as funções acima pelo fato de serem contínuas e contemplarem as peculiaridades das **IMFs** para um amplo espectro de massas. Outras **IMFs** amplamente utilizadas na literatura ([Scalo, 1986, 1998](#); [Kroupa, 2001, 2002](#); [Chabrier, 2003, 2005](#)) podem ser adicionadas se necessário.

Finalmente, métodos para composição hierárquica de diferentes sub-sistemas estão também implementados. Com isso, é possível criar, por exemplo, um aglomerado de estrelas onde cada partícula pode ser substituída hierarquicamente por sub-sistemas binários, ou triplos, ou sistemas planetários, ou até mesmo por sub-aglomerados. O contrário também é possível, i. e., a partir de um sistema triplo, por exemplo, pode-se substituir cada uma de suas partículas originais por um aglomerado de estrelas, etc. Em qualquer dos casos, os parâmetros das partículas originais (e. g., centro de massa, etc.) são herdados pelos seus respectivos sub-sistemas.

3.3.4 Módulo *lib*

Em *tupan*, sub-rotinas que escalam com $\mathcal{O}(N^2)$ dominam o custo computacional e compõe a parte do código que é diretamente implementada em linguagem compilada como forma de acelerar os cálculos. *tupan* suporta atualmente dois *backends*, escritos em C e OpenCL. O *backend* em C é implementado de maneira serial seguindo o princípio *KISS*, e nós o mantemos pelos seguintes motivos: i) como implementação de referência para verificação dos resultados calculados em paralelo com OpenCL, ii) como referência para medidas de ganho em performance, e iii) como meio tornar o código mais flexível, possibilitando seu uso mesmo em sistemas onde OpenCL não está disponível.

OpenCL é um padrão aberto da indústria de software, criado em 2008 baseado em C, que permite acessar os recursos computacionais de sistemas heterogêneos, tais como multi-core CPUs, GPUs e outras unidades de processamento paralelo, fornecendo um ambiente de programação multiplataforma e independente do fabricante do *hardware*, que possibilita o desenvolvedor escrever códigos portáteis e altamente eficientes. O modelo de programação paralela usado em OpenCL, em geral difere bastante do modelo de programação serial a que estamos acostumados. Uma descrição detalhada de seu funcionamento, entretanto, está fora do escopo do presente trabalho.

Aqui nós focamos em como a interface com CPUs e GPUs foi implementada em *tupan*. Na Fig. 3.2 nós mostramos o fluxo de execução que normalmente procede com o uso de *tupan*. No caso de desenvolvedores que desejarem acrescentar/modificar alguma funcionalidade ao código, é possível editá-lo antes de iniciar a sua execução. *tupan* automaticamente se encarrega de recompilar a versão modificada, caso esta não se encontre em *cache*. Nós enfatizamos que todo este processo é realizado de forma transparente ao usuário final.

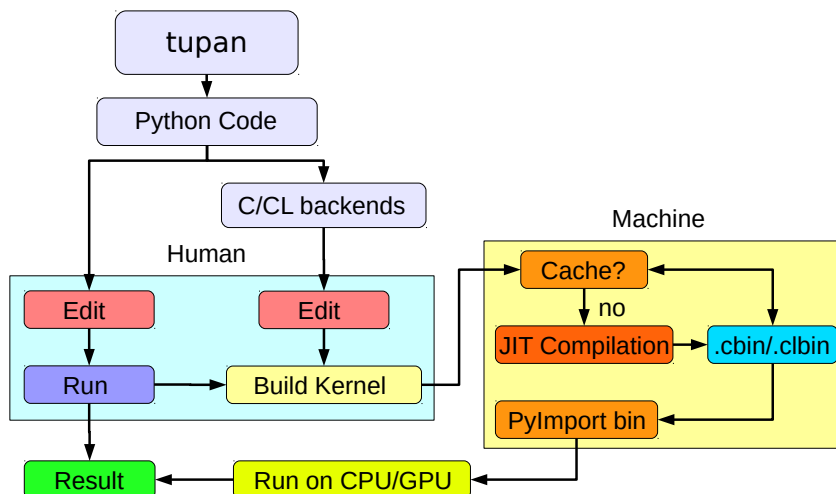


Figura 3.2: Fluxo de execução nas diferentes partes de *tupan*.

Do ponto de vista técnico, as partes do código que não dependem de métodos gravitacionais seguem um fluxo de execução direto, executando apenas código escrito em Python. Eventualmente, quando da chamada à qualquer dos

métodos gravitacionais ou, e. g., rotinas que escalam com $\mathcal{O}(N^2)$, o fluxo de execução é então desviado para um dos *backends*, em C ou OpenCL. A escolha do *backend* é predeterminada pelo usuário via linha de comando.

3.3.5 *Módulo integrator*

O módulo *integrator* mantém a implementação dos diferentes métodos de integração disponíveis em tupan. Atualmente, tupan conta com implementações do método de integração de Hermite (Makino, 1991b; Makino and Aarseth, 1992; Makino et al., 2006) com passo de tempo constante ou variável (porém o mesmo para todas as partículas), métodos de regularização algorítmica baseados em transformações Hamiltonianas num espaço de fase aumentado (Mikkola and Tanikawa, 1999a,b; Mikkola and Aarseth, 2002), bem como uma família de métodos de integração simplética, contendo um total de 48 integradores divididos em três grupos, usando diferentes esquemas de passo de tempo: constante, variável e *Block-Time-Steps* (BTSs, veja McMillan, 1986; Makino, 1991a). Além disso, tupan conta também com a implementação de referência do integrador sakura, desenvolvido pelo autor (Gonçalves Ferrari et al., 2014). Os métodos de Hermite e de regularização algorítmica não serão descritos neste trabalho. O leitor interessado encontrará informações detalhadas nas referências dadas. A família de integradores simpléticos e o integrador sakura são baseados na teoria apresentada no Cap. 2 e serão detalhados e discutidos no Cap. 4.

3.3.6 *Módulo simulation*

O módulo *simulation* mantém a implementação de interfaces para realização e diagnóstico de simulações de N -corpos. Suporte à interrupção e reinicialização de simulações está também implementada, permitindo ao usuário continuar uma simulação previamente interrompida, sem introduzir artefatos devido ao processo de reinicialização.

3.3.7 *Módulo animation*

tupan possui suporte à visualização 3D via OpenGL¹⁰, permitindo que o usuário possa facilmente visualizar uma simulação em tempo real, ou após esta ter sido completada. O usuário tem também a possibilidade de interagir com a visualização via comandos por teclado e *mouse* e gravá-la em vídeo, se assim desejar.

3.4 EXEMPLOS DE USO PRÁTICO COM TUPAN

Nesta seção nós apresentamos dois exemplos simples de como usar tupan: i) para construir condições iniciais e ii) para executar uma simulação.

¹⁰ <http://www.opengl.org/>.

3.4.1 Modelo de Plummer virializado com uma IMF tipo SALPETER1955

O trecho de código na List. 3.1 mostra como criar um modelo de Plummer virializado com $N = 4096$ partículas usando uma IMF tipo SALPETER1955 para um intervalo de massas entre $0.4 M_{\odot}$ e $120.0 M_{\odot}$. A Fig. 3.3 mostra o resultado do modelo em N -body units (Heggie and Mathieu, 1986).

Listagem 3.1: (Python shell): Usando tupan para gerar um modelo de Plummer virializado com uma IMF tipo SALPETER1955.

```

from tupan.ics.plummer import make_plummer
from tupan.io import IO
n = 4096 # no. of stars
imf = ("salpeter1955", 0.5, 120.0) # IMF description
ps = make_plummer(n, eps=4/n, imf, seed=1) # construct model
io = IO("plummer4096.hdf5", "w") # setup output file
io.dump_snapshot(ps) # saves data

```

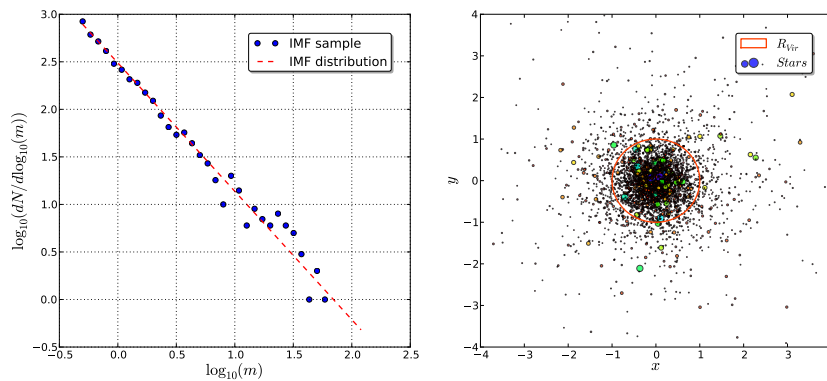


Figura 3.3: Modelo construído com o trecho de código na List. 3.1. O painel à esquerda mostra a distribuição das massas das partículas seguindo uma IMF tipo SALPETER1955. O painel à direita mostra um modelo de Plummer com 4096 partículas em equilíbrio virial. O tamanho dos símbolos é proporcional à massa das partículas, enquanto o círculo laranja demarca, para comparação, o raio virial do sistema.

3.4.2 Executando uma simulação com visualização em tempo real

Para executar uma simulação com o modelo criado anteriormente, basta uma única linha de comando, como mostrado na List. 3.2. tupan é executado com parâmetros definindo o arquivo de *input* (`-i/--input`), o método de integração numérica (`-m/--meth`), o parâmetro de precisão do integrador (`-e/--eta`), a duração da simulação em N -body Units (`-t/--t_end`), parâmetro definindo o arquivo de *output* (`-o/--output`), parâmetro definindo OpenCL como *backend* (`--backend`) e parâmetro solicitando que a simulação seja visualizada em tempo real (`--view`). tupan aceita vários outros parâmetros de entrada a fim de flexibilizar seu uso. Uma lista completa com uma breve descrição de cada um deles pode ser obtida via linha de comando com `./tupan --help`.

Listagem 3.2: (Unix shell): Usando tupan para executar uma simulação com visualização em tempo real.

```
./tupan simulation -i plummer4096.hdf5 -m sia21h.dkd -e 0.25 -t 10
-o snapshots.hdf5 --backend=CL --view
```



Figura 3.4: Janela de visualização 3D lançada por tupan quando a opção `--view` é ativada durante sua execução.

Na Fig. 3.5 nós mostramos a evolução temporal das energias cinética, potencial, virial e total do sistema. Uma vez que o sistema está inicialmente em equilíbrio virial, somente pequenas flutuações em torno de seus valores iniciais ocorre, como esperado para a evolução dessas grandezas. A qualidade da integração, como medida pelo erro na energia total do sistema, é vista na parte inferior da Fig. 3.5. Para o valor de η adotado, este permanece sempre $\sim 10^{-5}$. As flutuações mais intensas a partir de $t = 60$ são devido à formação dinâmica de binárias no sistema.

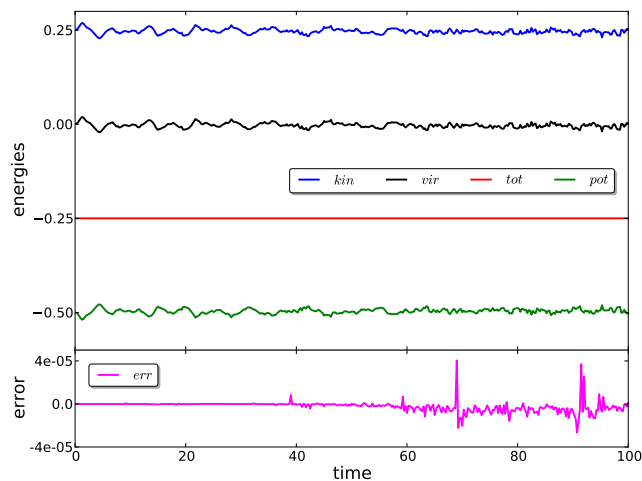


Figura 3.5: Evolução temporal das energias do sistema mostrado na Fig. 3.4.

3.5 MEDIDAS DE PERFORMANCE

Nesta seção nós discutimos alguns testes de desempenho. Tendo em vista que o método que domina o custo computacional em tupan é o cálculo da aceleração gravitacional do sistema de partículas, nós aqui avaliamos seu desempenho em CPUs e GPUs, a fim de ter uma estimativa da performance geral do código nestes dispositivos.

Medidas da performance de uma dada rotina geralmente são feitas em termos do tempo de execução, t_{exec} , para diferentes tamanhos do problema que ela resolve. Uma outra métrica comumente utilizada é o número de operações de ponto flutuante (*flops*) por segundo ($n_{flops/s}$). Em nossa implementação, o cálculo da força gravitacional para um par de partículas necessita de 20 flops^{11} . Consequentemente, o cálculo para um sistema de N partículas requer $20N^2$ operações. Assim, o número de *flops/s* para este caso é dado por

$$n_{flops/s} = \frac{20N^2}{t_{exec}}. \quad (3.6)$$

A Fig. 3.6 apresenta os resultados dos testes (os valores mostrados são médias sobre 10 execuções e não possuem dispersão significativa). No eixo horizontal nós variamos o número de partículas, N . O tempo de execução para cada valor de N é mostrado pelas curvas com pontos, correspondendo ao eixo vertical direito, enquanto $n_{flops/s}$ é dado no eixo vertical esquerdo em Giga-*flops/s*, e representado pelas barras. Para ambos os casos, em verde são mostrados resultados para a implementação de referência correspondente ao *backend* em C executada em um único *core* da CPU¹². Os resultados correspondentes ao *backend* em OpenCL são mostrados em azul para sua execução nos dois *cores* da mesma CPU, e em vermelho para sua execução nos 480 *cores* da GPU¹³. Em cada caso foi avaliado também a performance de execução em precisão simples (SP) bem como em precisão dupla (DP).

Nós notamos que para $N \lesssim 256$, a implementação paralela em OpenCL é menos eficiente do que a versão serial em C devido à uma pequena sobrecarga na comunicação dos dados com o dispositivo em uso (CPU ou GPU), frente ao tempo de cálculo. Para $N > 256$, a implementação OpenCL supera a performance da versão serial, chegando a um fator de ganho ~ 1.7 na CPU para $N = 131072$, compatível com o esperado (fator 2, para 2 *cores*). O pico de performance para a execução serial na CPU para esse valor de N foi de 1.09 Gflop/s em DP e 1.22 Gflop/s em SP, tomando cerca de 375.3 s e 337.0 s de tempo de processamento em DP e SP, respectivamente. Para a execução paralela na CPU, o pico de performance para o mesmo valor de N foi de 1.70 Gflop/s em DP e 2.10 Gflop/s em SP, o que tomou cerca de 241.8 s e 196.3 s de tempo de processamento em DP e SP, respectivamente.

Na GPU, para $N = 131072$, o pico de performance foi de 78.8 Gflop/s em DP e de 740.9 Gflop/s em SP, custando somente cerca de 5.2 s e 0.56 s de tempo

¹¹ Nós contabilizamos cada operação aritmética (+, -, *, /), bem como `sqrt`, como 1 *flop*. Em alguns sistemas, `sqrt` pode consumir até $\sim 18 \text{ flops}$, de modo que nossa estimativa representa um limite inferior.

¹² Modelo Intel Core 2 Duo, 2 GHz.

¹³ Modelo NVIDIA GeForce GTX 480, 1.401 GHz.

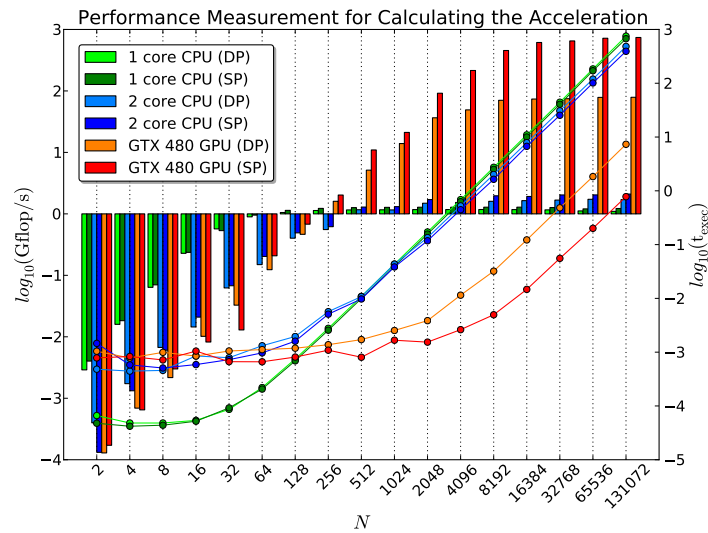


Figura 3.6: Medidas de desempenho para o cálculo da aceleração gravitacional com tupan, para diferentes números de partículas. As barras dão a performance em Gflops/s, correspondente ao eixo à esquerda. As linhas com pontos dão o tempo de execução em segundos, correspondente ao eixo à direita.

de processamento em DP e SP, respectivamente. O fator ~ 10 diferenciando a performance em DP e SP está dentro do esperado (fator 8) para o nosso modelo de GPU.

Comparando, por exemplo, os picos de performance em SP entre a GPU e a versão serial na CPU, nota-se que o ganho relativo pode atingir um fator tão alto quanto ~ 600 . Obviamente, esse fator possui um *bias* por conta da diferença na frequência de operação entre CPUs e GPU. Corrigindo esse fator pela razão entre estas frequências, nós obtemos que o ganho proporcionado pela GPU é ainda ~ 420 , o que mostra que seu uso para o cálculo da aceleração gravitacional melhora a eficiência de simulações de N -corpos por mais que duas ordens de grandeza!

Por fim, como uma estimativa, nós calculamos que uma simulação de N -corpos típica, usando $N \sim 10^5$ partículas e necessitando cerca de 10^6 passos de integração possa ser completada em uma GPU em um período de tempo de ~ 10 dias. Esta estimativa torna-se ainda mais otimista com o uso de GPUs mais recentes. Para fins de comparação, a mesma simulação tomaria cerca de 10 anos para ser completada usando um único *core* de uma CPU.

Talk is cheap. Show me the code.

— Linus Torvalds

4

NOVAS TÉCNICAS DE INTEGRAÇÃO NUMÉRICA PARA SISTEMAS DE N -CORPOS COM MÚLTIPLAS ESCALAS

Neste Capítulo nós apresentamos dois novos métodos para integração eficiente de sistemas gravitacionais de N -corpos com uma grande variação em escalas de tempo características. O primeiro método fundamenta a implementação de uma família de integradores simpléticos em tupan e é baseado numa separação hierárquica do Hamiltoniano, primeiramente introduzida por [Pelupessy et al. \(2012\)](#). Nós o descrevemos brevemente e apresentamos uma generalização à ordens mais altas que $\mathcal{O}(\tau^2)$, e mostramos como incluir correções PNs de maneira auto-consistente e temporalmente simétrica. Nós então apresentamos os fundamentos do integrador sakura, desenvolvido pelo autor ([Gonçalves Ferrari et al., 2014](#)), baseado numa separação Hamiltoniana em termos de N^2 problemas de Kepler, e discutimos alguns testes numéricos.

4.1 SEPARAÇÃO HIERÁRQUICA DO HAMILTONIANO: *the slow-fast split*

No Cap. 2 nós vimos que quando o operador Hamiltoniano, \mathcal{H} , pode ser separado em termos dos operadores associados às energias cinética e potencial, \mathcal{T} e \mathcal{U} , é possível aproximar o operador de evolução temporal $e^{\tau\mathcal{H}}$ em segunda ordem,

$$\phi_{\tau}^2 = e^{\frac{\tau}{2}\mathcal{T}} e^{\tau\mathcal{U}} e^{\frac{\tau}{2}\mathcal{T}}, \quad (4.1)$$

como dado pela eq. 2.41, ou pelo seu auto-adjunto (eq. 2.43). Este tipo de separação Hamiltoniana não é a única possível e muitas outras formas de subdividir o sistema foram investigadas previamente ([Wisdom and Holman, 1991](#); [Duncan et al., 1998](#); [Chambers, 1999](#); [Fujii et al., 2007](#)).

Em [Pelupessy et al. \(2012\)](#) uma nova estratégia de separação recursiva do Hamiltoniano foi introduzida, baseada no valor do passo de tempo atribuído à cada partícula no sistema. Para cada nível da recursão, todas as partículas sob consideração são divididas em grupos *slow* (S) e *fast* (F), comparando-se o passo de tempo individual da partícula, τ_i , à um passo de tempo de referência, τ . Assim, S e F são definidos por,

$$S \equiv \{i \in 1 \dots N : \tau_i \geq \tau\}, \quad (4.2)$$

$$F \equiv \{i \in 1 \dots N : \tau_i < \tau\}. \quad (4.3)$$

Usando os grupos de partículas S e F , é possível reescrever o Hamiltoniano do sistema como segue,

$$H = H_S + H_F + U_{SF}. \quad (4.4)$$

O sub-Hamiltoniano H_S pode ser visualizado como um “sistema Hamiltoniano fechado”, no qual as partículas em S interagem entre si sem sentir a influência das partículas em F . A mesma propriedade vale para o sub-Hamiltoniano H_F e suas partículas. Finalmente, o sub-Hamiltoniano U_{SF} contém os termos de interação entre as partículas em S e F .

Dado o Hamiltoniano na eq. 4.4, [Pelupessy et al. \(2012\)](#) propõe a seguinte aproximação em segunda ordem para o operador de evolução temporal,

$$e^{\tau\mathcal{H}} \approx e^{\frac{\tau}{2}\mathcal{H}_F} e^{\tau(\mathcal{H}_S+U_{SF})} e^{\frac{\tau}{2}\mathcal{H}_F}. \quad (4.5)$$

O sub-Hamiltoniano \mathcal{H}_F é fechado, e consiste de partículas onde $\tau_i < \tau$. A evolução temporal de \mathcal{H}_F é realizada pela aplicação recursiva do mapa na eq. 4.5, porém usando um passo de tempo de referência, $\tau/2$. Em contraste, ambos U_{SF} e $\mathcal{H}_S = \mathcal{T}_S + U_S$ podem ser decompostos em termos de *drifts* ($e^{\tau T}$) e *kicks* ($e^{\tau U}$) individuais, aplicados usando o passo de tempo de referência atual, τ . Assim,

$$e^{\tau(\mathcal{H}_S+U_{SF})} = e^{\tau(\mathcal{T}_S+U_S+U_{SF})} \approx e^{\frac{\tau}{2}\mathcal{T}_S} e^{\tau(U_S+U_{SF})} e^{\frac{\tau}{2}\mathcal{T}_S}. \quad (4.6)$$

O passo de tempo de referência é dividido pela metade a cada nível da recursão, e esta termina quando todas as partículas restantes em F são colocadas no grupo S de seu respectivo nível. O método resultante possui convergência em $\mathcal{O}(\tau^2)$ e realiza uma cadência hierárquica de passos de tempo similar àquela obtida utilizando o esquema convencional com *BTSs*. É importante notar que no método acima, *kicks* da partícula i sobre a partícula j são sempre emparelhados com *kicks* da partícula j sobre a partícula i . Isto, garante a conservação do momentum a cada nível de recursão. Além disso, contrário ao esquema *BTS*, onde cada partícula é individualmente evoluída sob influência da força *total* devido ao resto do sistema, no método de [Pelupessy et al. \(2012\)](#) *kicks* sobre uma dada partícula à qualquer passo de tempo dado, tipicamente corresponde à forças parciais, resultando em uma complexidade computacional $\mathcal{O}(N^\gamma)$ com tipicamente $\gamma \lesssim 2$, ao invés da convencional $\mathcal{O}(N^2)$.

Quando o passo de tempo individual das partículas é forçado ao valor mínimo dentre todas elas, o método acima reduz-se à um único nível de recursão, resultando num esquema de passos de tempo variável (porém o mesmo para todas as partículas). Um esquema com passos de tempo constante é obtido simplesmente forçando τ_i à um valor maior ou igual ao passo de tempo de referência inicial, τ .

Antes de apresentar a nossa generalização do método acima nós discutimos a seguir o critério para o passo de tempo.

4.1.1 Critério para o passo de tempo

Como discutido no final da seção 2.2, integradores simpléticos são, por construção, designados à utilizar passos de tempo constantes. Caso isto não seja

obedecido as integrais de movimento do sistema tendem a sofrer um desvio sistemático do nível de conservação esperado, frequentemente gerando soluções com artefatos numéricos (Gladman et al., 1991; Skeel and Gear, 1992). Interessantes soluções para este problema têm sido propostas ao longo dos anos (Hut et al., 1995; Lee et al., 1997; Duncan et al., 1998; Preto and Tremaine, 1999; Mikkola and Tanikawa, 1999b,a; Mikkola and Aarseth, 2002; Makino et al., 2006), e a razão para tal degradação parece estar relacionada à perda da simetria temporal durante a escolha do passo de tempo.

Por exemplo, dada a trajetória de uma partícula $\mathbf{w}(t) = (\mathbf{r}(t), \mathbf{p}(t))$, e um dado critério¹, $\vartheta(\mathbf{w})$, para a escolha do passo de tempo, pode-se determinar $\tau_0 \propto \vartheta(\mathbf{w}_0)$, onde $\mathbf{w}_0 = \mathbf{w}(t_0)$, e usar este valor para avançar o estado do sistema. Assim, $\mathbf{w}(t_0) \rightarrow \mathbf{w}(t_0 + \tau_0)$ leva o sistema à um novo estado $\mathbf{w}(t_1) \equiv \mathbf{w}(t_0 + \tau_0)$ cujo passo de tempo é $\tau_1 \propto \vartheta(\mathbf{w}_1)$. Se agora fizermos o caminho inverso, $\mathbf{w}(t_1) \rightarrow \mathbf{w}(t_1 - \tau_1)$ leva o sistema à um novo estado $\mathbf{w}(t_2) \equiv \mathbf{w}(t_1 - \tau_1)$ cujo passo de tempo é $\tau_2 \propto \vartheta(\mathbf{w}_2)$. Acontece que, e.g., $\tau_2 \neq \tau_0$, pois $\mathbf{w}_2 \neq \mathbf{w}_0$, de modo que o estado original do sistema em $t = t_0$ é perdido. Essa situação não teria acontecido se o passo de tempo tivesse a propriedade $\tau(t + \tau(t)) = \tau(t)$, que só é estritamente possível com $\tau = cte$.

Hut et al. (1995) propõe escolher o passo de tempo implicitamente, tal que

$$\tau_{n+1/2} \propto \frac{1}{2} [\vartheta(\mathbf{w}_n) + \vartheta(\mathbf{w}_{n+1})], \quad (4.7)$$

e usar $\tau_{n+1/2}$ para evoluir o sistema de um estado para o próximo ou vice-versa. Porém tal proposta torna o método de integração implícito e desnecessariamente caro computacionalmente. Além disso, como qualquer método implícito, iteração é necessária para garantir a convergência da solução (neste caso do valor do passo de tempo). Makino et al. (2006) mostrou que é possível construir uma hierarquia de passos baseada no esquema BTS e na eq. 4.7 acima, contudo aquele método, além de caro computacionalmente é desnecessariamente mais complexo de se implementar. Nós, ao contrário, pensamos que métodos explícitos são mais vantajosos ambos em termos do custo computacional quanto em termos da facilidade de implementação. Assim, nós superamos tais dificuldades da seguinte forma. Primeiramente nós redefinimos a maneira em que calculamos o passo de tempo. Ao invés de ϑ , nós usamos uma espécie de frequência associada à função ϑ , $\omega_n \equiv \omega(\mathbf{w}(t_n))$. Assim o passo de tempo $\tau_{n+1/2}$ fica definido como,

$$\tau_{n+1/2} = \frac{\eta}{\sqrt{\omega_{n+1/2}^2}}, \quad (4.8)$$

onde η é um parâmetro de entrada que define a precisão do integrador. Seguindo a proposta de Hut et al. (1995), nós definimos uma forma simetrizada para a frequência, $\omega_{n+1/2}^2$, de modo que,

$$\omega_{n+1/2}^2 = \frac{1}{2} [\omega_n^2 + \omega_{n+1}^2]. \quad (4.9)$$

¹ A forma da função ϑ não é importante no momento. $\vartheta \propto r_{ij}^{3/2}$, p. ex., é uma forma aceitável.

Aproximando o termo ω_{n+1}^2 por sua expansão em série em primeira ordem,

$$\begin{aligned}\omega_{n+1}^2 &\approx \omega_n^2 + \tau_n \left. \frac{d\omega^2}{dt} \right|_{t=t_n} + \mathcal{O}(\tau_n^2), \\ &\approx \omega_n^2 + \frac{\eta}{\omega_n} \frac{d\omega_n^2}{dt},\end{aligned}\quad (4.10)$$

e inserindo de volta na eq. 4.9 resulta em,

$$\omega_{n+1/2}^2 \approx \omega_n^2 + \frac{1}{2} \frac{\eta}{\omega_n} \frac{d\omega_n^2}{dt}. \quad (4.11)$$

Note que este procedimento de simetrização é genérico e completamente explícito. O erro $\mathcal{O}(\tau^2)$ na escolha de $\omega_{n+1/2}^2$ não representa um problema pois, para propósitos práticos, resulta apenas em um erro de ordem $\mathcal{O}(\tau^3)$ na simplecticidade de um integrador de segunda ordem. O fato deste procedimento introduzir apenas um erro da mesma ordem do erro do próprio integrador é que torna tal aproximação boa o suficiente para manter o nível esperado de conservação das integrais de movimento à longo prazo.

A efetividade do passo de tempo resultante depende, em última instância, da expressão adotada para o critério $\omega(\mathbf{w}(t))$. No presente trabalho nós adotamos a seguinte expressão,

$$\omega_{ij}^2(t) = \frac{v_{ij}^2}{r_{ij}^2} + \frac{2\mu_{ij}}{r_{ij}^3}, \quad (4.12)$$

onde $\mu_{ij} = \frac{m_i m_j}{m_i + m_j}$ é a massa reduzida do par ij . A eq. 4.12, quando corrigida pela eq. 4.11 e inserida na eq. 4.8, dá o valor simetrizado do passo de tempo para a interação do par ij . Seguindo a mesma lógica, o passo de tempo da i -ésima partícula é obtido a partir da versão simetrizada da soma,

$$\omega_i^2(t) = \sum_{j \neq i}^N \omega_{ij}^2(t). \quad (4.13)$$

[Pelupessy et al. \(2012\)](#) propõe uma estratégia similar para simetrização do passo de tempo, porém a nossa escolha por utilizar a frequência, ω , ao invés de um critério diretamente proporcional à τ , como feito naquele trabalho, resulta em expressões mais simples e uma implementação mais eficiente.

4.1.2 Generalização à uma família de integradores

Primeiramente, nós definimos os operadores *drift* e *kick*,

$$D_\tau \equiv e^{\tau\mathcal{T}}, \quad (4.14)$$

$$K_\tau \equiv e^{\tau\mathcal{U}}. \quad (4.15)$$

Assim, a eq. 4.1 pode ser reescrita como,

$$DKD_\tau = D_{\tau/2} K_\tau D_{\tau/2}, \quad (4.16)$$

e seu auto-adjunto,

$$KDK_\tau = K_{\tau/2}D_\tau K_{\tau/2}. \quad (4.17)$$

Em [Forest and Ruth \(1990\)](#) mostrou-se que o operador de evolução temporal $e^{\tau(\mathcal{T}+\mathcal{U})}$ pode ser fatorado como

$$e^{\tau(\mathcal{T}+\mathcal{U})} \approx \prod_{p=1}^P D_{a_p\tau} K_{b_p\tau} + O(\tau^{Y+1}), \quad (4.18)$$

onde, para um dado P , a_p e b_p são coeficientes numéricos que são encontrados a partir das condições de ordem necessárias para maximizar a ordem Y do integrador, com o vínculo dado pelas condições de simetria, $a_1 = 0$, $a_{p+1} = a_{p-p+1}$, $b_p = b_{p-p+1}$, ou $a_p = a_{p-p+1}$, $b_p = b_{p-p}$, $b_P = 0$. A eq. 4.18 é, de fato, mais geral do que o procedimento de [Yoshida \(1990\)](#) discutido no Cap. 2 e diversos estudos foram realizados para se determinar tais coeficientes para diversas ordens (veja, p. ex., [Omelyan et al., 2003](#)).

Dessa forma, para a_p e b_p conhecidos, as eqs. 4.16 e 4.17 podem ser generalizadas como,

$$DKD_\tau^{YZ} = \left[\prod_{z=1}^{Z-1} D_{a_z\tau} K_{b_z\tau} \right] D_{a_Z\tau} K_{b_Z\tau} D_{a_Z\tau} \left[\prod_{z=Z-1}^1 K_{b_z\tau} D_{a_z\tau} \right], \quad (4.19)$$

$$KDK_\tau^{YZ} = \left[\prod_{z=1}^{Z-1} K_{a_z\tau} D_{b_z\tau} \right] K_{a_Z\tau} D_{b_Z\tau} K_{a_Z\tau} \left[\prod_{z=Z-1}^1 D_{b_z\tau} K_{a_z\tau} \right], \quad (4.20)$$

onde Y define a ordem do integrador e Z o número de estágios.

Para generalizar o método de [Pelupessy et al. \(2012\)](#), nós primeiramente re-escrevemos a eq. 4.5 como segue,

$$\begin{aligned} e^{\tau\mathcal{H}} &\approx e^{\frac{\tau}{2}\mathcal{H}_F} e^{\tau(\mathcal{H}_S+\mathcal{U}_{SF})} e^{\frac{\tau}{2}\mathcal{H}_F}, \\ &\approx e^{\frac{\tau}{2}\mathcal{H}_F} e^{\frac{\tau}{2}\mathcal{H}_S} e^{\tau\mathcal{U}_{SF}} e^{\frac{\tau}{2}\mathcal{H}_S} e^{\frac{\tau}{2}\mathcal{H}_F}, \\ &\approx e^{\frac{\tau}{2}(\mathcal{H}_F+\mathcal{H}_S)} e^{\tau\mathcal{U}_{SF}} e^{\frac{\tau}{2}(\mathcal{H}_S+\mathcal{H}_F)}, \end{aligned} \quad (4.21)$$

onde a última linha acima, vem do fato que \mathcal{H}_S e \mathcal{H}_F comutam, i. e., $[\mathcal{H}_S, \mathcal{H}_F] = 0$. Essa pequena sutileza na forma como os operadores Hamiltonianos são organizados na eq. 4.21 é de extrema importância aqui. A primeira linha na eq. 4.21 corresponde ao método de [Pelupessy et al. \(2012\)](#) (eq. 4.5). A última linha na eq. 4.21, contudo, expressa o operador de evolução temporal $e^{\tau\mathcal{H}}$ no formato BRIDGE ([Fujii et al., 2007](#)), em que os sub-sistemas S e F podem ser tratados de maneira autônoma e evoluídos independentemente pelo intervalo τ . É importante salientar aqui que, embora no presente contexto S e F correspondam à diferentes sub-sistemas em um sistema gravitacional, a generalidade da formulação na eq. 4.21 permite que \mathcal{H}_F e \mathcal{H}_S correspondam, p. ex., à diferentes Hamiltonianos (ou, mais genericamente, diferentes códigos) modelando diferentes domínios astrofísicos, com \mathcal{U}_{SF} modelando o acoplamento entre tais domínios. Assim, definindo-se novos operadores *drift* e *kick* para a sub-divisão recursiva *slow-fast*,

$${}^{SF}D_\tau \equiv e^{\tau(\mathcal{H}_F+\mathcal{H}_S)}, \quad (4.22)$$

$${}^{SF}K_\tau \equiv e^{\tau\mathcal{U}_{SF}}, \quad (4.23)$$

é imediato notar que as eqs. 4.19 e 4.20, com $D_\tau \rightarrow {}^{SF}D_\tau$ e $K_\tau \rightarrow {}^{SF}K_\tau$, podem ser usadas como uma generalização do método de Pelupessy et al. (2012). Nós chamaremos a versão *SF* de tais equações como, ${}^{SF}DKD_\tau^{YZ}$ e ${}^{SF}KDK_\tau^{YZ}$, respectivamente. Assim,

$${}^{SF}DKD_\tau^{YZ} = \left[\prod_{z=1}^{Z-1} \dots \right] {}^{SF}D_{a_z\tau} {}^{SF}K_{b_z\tau} {}^{SF}D_{a_z\tau} \left[\prod_{z=Z-1}^1 \dots \right], \quad (4.24)$$

$${}^{SF}KDK_\tau^{YZ} = \left[\prod_{z=1}^{Z-1} \dots \right] {}^{SF}K_{a_z\tau} {}^{SF}D_{b_z\tau} {}^{SF}K_{a_z\tau} \left[\prod_{z=Z-1}^1 \dots \right], \quad (4.25)$$

onde os termos entre colchetes são similares aos das eqs. 4.19 e 4.20, porém usando a versão *SF* dos operadores *drift* e *kick* (eqs. 4.22 e 4.23).

Na tabela 4.1 nós apresentamos todos os 48 métodos de integração simplética resultantes da família acima e implementados em tupan de acordo com ordem de integração (Y), números de estágios (Z) e esquema de passos de tempo. Resta agora discutir como os operadores ${}^{SF}D_\tau$ e ${}^{SF}K_\tau$ nas eqs. 4.22 e 4.23 são fatorados em termos de D_τ e K_τ , o que será feito na seção seguinte, onde D_τ e K_τ serão também generalizados com a inclusão de termos *PNs*.

4.1.3 Operadores *drift* e *kick* com correções relativísticas

Nós começamos com o operador *drift*. D_τ , dado inicialmente pela eq. 4.14, é redefinido explicitamente como,

$$D_\tau \equiv D_\tau^{PN} \text{ if PN is True, else } D_\tau^N. \quad (4.26)$$

D_τ^N representa um *drift* à maneira Newtoniana (eq. 4.14), ou seja, apenas o avanço das coordenadas de posição das partículas. D_τ^{PN} faz o mesmo que D_τ^N e, adicionalmente, avança termos de correção para o centro de massa do sistema² (eq. 2.6).

Para o operador *kick*, nós começamos redefinindo K_τ , dado inicialmente pela eq. 4.15, como,

$$K_\tau \equiv K_\tau^{PN} \text{ if PN is True, else } K_\tau^N. \quad (4.27)$$

K_τ^N representa um *kick* à maneira Newtoniana (eq. 4.15), ou seja, apenas o avanço das coordenadas de momentum (velocidade) das partículas. K_τ^{PN} faz o mesmo que K_τ^N , porém utilizando o algoritmo de Hellström and Mikkola (2010) em que *kicks* devido à forças dependentes da velocidade, $\mathbf{f}(\mathbf{r}, \mathbf{v})$, são realizados de maneira *explícita* com o uso de uma velocidade auxiliar, \mathbf{u} , com $\mathbf{u}_0 = \mathbf{v}_0$, tal que,

$$\begin{aligned} \mathbf{v}_{n+\frac{1}{2}} &= \mathbf{v}_n + \frac{\tau}{2} \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}, \mathbf{u}_n), \\ \mathbf{u}_{n+1} &= \mathbf{u}_n + \tau \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}, \mathbf{v}_{n+\frac{1}{2}}), \\ \mathbf{v}_{n+1} &= \mathbf{v}_{n+\frac{1}{2}} + \frac{\tau}{2} \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}, \mathbf{u}_{n+1}). \end{aligned}$$

² Note que termos de correção para quantidades globais como centro de massa, energia, etc. não influencia a dinâmica das partículas. Estes são integrados apenas para fins de verificação da conservação das integrais de movimento do sistema.

Tabela 4.1: Família de 48 mapas simpléticos implementados em tupan, classificados por nome, ordem de integração (Y), números de estágios (Z), esquema de passos de tempo e as devidas referências para os coeficientes a_p e b_p .

name	order (Y)	stages (Z)	time-step	Ref.
sia21c.dkd / sia21c.kdk	2	1	constant	a
sia21a.dkd / sia21a.kdk			adaptive	
sia21h.dkd / sia21h.kdk			hierarchical	
sia22c.dkd / sia22c.kdk	2	2	constant	b
sia22a.dkd / sia22a.kdk			adaptive	
sia22h.dkd / sia22h.kdk			hierarchical	
sia43c.dkd / sia43c.kdk	4	3	constant	a
sia43a.dkd / sia43a.kdk			adaptive	
sia43h.dkd / sia43h.kdk			hierarchical	
sia44c.dkd / sia44c.kdk	4	4	constant	b
sia44a.dkd / sia44a.kdk			adaptive	
sia44h.dkd / sia44h.kdk			hierarchical	
sia45c.dkd / sia45c.kdk	4	5	constant	b
sia45a.dkd / sia45a.kdk			adaptive	
sia45h.dkd / sia45h.kdk			hierarchical	
sia46c.dkd / sia46c.kdk	4	6	constant	c
sia46a.dkd / sia46a.kdk			adaptive	
sia46h.dkd / sia46h.kdk			hierarchical	
sia67c.dkd / sia67c.kdk	6	7	constant	a
sia67a.dkd / sia67a.kdk			adaptive	
sia67h.dkd / sia67h.kdk			hierarchical	
sia69c.dkd / sia69c.kdk	6	9	constant	d
sia69a.dkd / sia69a.kdk			adaptive	
sia69h.dkd / sia69h.kdk			hierarchical	

a. Yoshida (1990)

b. Omelyan et al. (2003)

c. Blanes and Moan (2002)

d. Kahan and Li (1997)

A expressão acima é equivalente a usar um algoritmo de *leap-frog* para avançar as velocidades \mathbf{v} e \mathbf{u} . Após isso, a velocidade, \mathbf{v} , é usada para prosseguir com o avanço das posições. Anteriormente ao trabalho de Hellström and Mikkola (2010), o que se fazia era utilizar uma forma *implícita*, tal como

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \tau \mathbf{f} \left(\mathbf{r}_{n+\frac{1}{2}}, \frac{1}{2} (\mathbf{v}_n + \mathbf{v}_{n+1}) \right),$$

para o avanço das velocidades, o que, obviamente, requer iteração até convergência. Além disso, K_τ^{PN} avança também termos de correção para a energia total, momentum linear e momentum angular do sistema (eqs. 2.5, 2.7 e 2.8), de maneira similar ao avanço das velocidades.

Desta forma, ${}^{SF}D_\tau$, como dado pela eq. 4.22, pode ser decomposto em,

$${}^{SF}D_\tau = {}^{SF}DKD_{\tau/2}^{YZ}(F) DKD_{\tau/2}^{YZ}(S) {}^{SF}DKD_{\tau/2}^{YZ}(F), \quad (4.28)$$

ou

$${}^{SF}D_\tau = {}^{SF}KDK_{\tau/2}^{YZ}(F) KDK_{\tau/2}^{YZ}(S) {}^{SF}KDK_{\tau/2}^{YZ}(F), \quad (4.29)$$

onde ${}^{SF}DKD_{\tau/2}^{YZ}(F)$ (ou ${}^{SF}KDK_{\tau/2}^{YZ}(F)$) representa uma chamada recursiva à eq. 4.24 (ou eq. 4.25) sobre o sub-sistema F usando metade do passo de tempo de referência. $DKD_{\tau/2}^{YZ}(S)$ (ou $KDK_{\tau/2}^{YZ}(S)$) é simplesmente a aplicação da eq. 4.19 (ou eq. 4.20) sobre o sub-sistema S . Note que pela eq. 4.21, a eq. 4.28 poderia ser escrita como ${}^{SF}D_\tau = {}^{SF}DKD_{\tau/2}^{YZ}(F) DKD_{\tau/2}^{YZ}(S) = DKD_{\tau/2}^{YZ}(S) {}^{SF}DKD_{\tau/2}^{YZ}(F)$, e similar para sua versão auto-adjunta, dada a comutatividade dos operadores envolvidos, mas em tupan nós usamos a forma dada pela eq. 4.28 (ou eq. 4.29), que resulta ser mais eficiente quando alguns dos coeficientes a_p ou b_p são maiores que 1 ou são negativos.

Analogamente, ${}^{SF}K_\tau$, como dado pela eq. 4.23, pode ser decomposto em,

$${}^{SF}K_\tau = {}^{SF}K_\tau^{PN} \text{ if PN is True, else } {}^{SF}K_\tau^N, \quad (4.30)$$

onde ${}^{SF}K_\tau^N$ é definido como,

$${}^{SF}K_\tau^N = K_\tau^N(S;F)K_\tau^N(F;S) = K_\tau^N(F;S)K_\tau^N(S;F), \quad (4.31)$$

ou seja, é simplesmente a aplicação do operador K_τ^N sobre o sub-sistema S , sob influência do sub-sistema F , seguido do mesmo para o sub-sistema F , sob influência do sub-sistema S , ou vice-versa.

Para ${}^{SF}K_\tau^{PN}$, nós construímos uma extensão do algoritmo de [Hellström and Mikkola \(2010\)](#), de modo que,

$$\begin{aligned} \mathbf{v}_{n+\frac{1}{2}}^S &= \mathbf{v}_n^S + \frac{\tau}{2} \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}^F, \mathbf{u}_n^F), \\ \mathbf{v}_{n+\frac{1}{2}}^F &= \mathbf{v}_n^F + \frac{\tau}{2} \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}^S, \mathbf{u}_n^S), \\ \mathbf{u}_{n+1}^S &= \mathbf{u}_n^S + \tau \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}^F, \mathbf{v}_{n+\frac{1}{2}}^F), \\ \mathbf{u}_{n+1}^F &= \mathbf{u}_n^F + \tau \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}^S, \mathbf{v}_{n+\frac{1}{2}}^S), \\ \mathbf{v}_{n+1}^S &= \mathbf{v}_{n+\frac{1}{2}}^S + \frac{\tau}{2} \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}^F, \mathbf{u}_{n+1}^F), \\ \mathbf{v}_{n+1}^F &= \mathbf{v}_{n+\frac{1}{2}}^F + \frac{\tau}{2} \mathbf{f}(\mathbf{r}_{n+\frac{1}{2}}^S, \mathbf{u}_{n+1}^S), \end{aligned}$$

com as grandezas em S e F sendo calculadas e evoluídas simultaneamente, a fim de evitar assimetrias temporais. Além disso, ${}^{SF}K_\tau^{PN}$ avança também termos de correção para a energia total, momentum linear e momentum angular do sistema (eqs. 2.5, 2.7 e 2.8) associados à interação SF , de maneira similar ao avanço das velocidades acima.

Com isso, as eqs. 4.24 ou 4.25 em sua forma recursiva constituem o que nós chamamos de BRIDGE-hierárquico. Note que na presente formulação apenas dois sub-sistemas (S e F) ou, genericamente, dois domínios astrofísicos podem ser hierarquicamente acoplados. Em Gonçalves Ferrari et al., (2015, *in preparation*) nós generalizamos ainda mais a formulação acima para um número arbitrário Q de sub-sistemas (ou domínios astrofísicos).

4.2 SEPARAÇÃO KEPLERIANA DO HAMILTONIANO: SAKURA

Nesta seção nós apresentamos os fundamentos do integrador sakura, desenvolvido pelo autor em colaboração com Tjarda Boekholt e Simon Portegies Zwart (Gonçalves Ferrari et al., 2014), baseado numa separação Hamiltoniana em termos de N^2 problemas de Kepler. Uma versão de sakura paralelizada com *Message Passing Interface* (MPI)³ para sistemas com memória distribuída foi então escrita, e em 2014 nós fomos agraciados com o prêmio Win Nieuwpoort⁴ na Holanda, por ter alcançado alta eficiência paralela em um grande sistema para *High-Performance Computing* (HPC).

Nós motivamos o método notando que para um sistema arbitrário de N -corpos a existência de soluções analíticas somente ocorre para os casos $N = 1$ (partícula livre) e $N = 2$ (problema de Kepler). Dessa forma, nós decidimos investigar a possibilidade de decompor o problema geral de N -corpos em termos de problemas mais simples e analiticamente solúveis (casos $N = 1$ e $N = 2$). Assim, nós começamos reescrevendo o Hamiltoniano de um sistema de N -corpos em termos de Hamiltonianos de Kepler, K_{ij} ⁵, como segue,

$$\begin{aligned} H &= T + U, \\ &= \sum_{i=1}^N T_i + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N U_{ij}, \\ &= \sum_{i=1}^N T_i + \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N (K_{ij} - T_{ij}), \end{aligned} \quad (4.32)$$

com

$$T_i = \frac{1}{2} m_i v_i^2, \quad (4.33)$$

$$U_{ij} = G \frac{m_i m_j}{r_{ij}}, \quad (4.34)$$

e

$$K_{ij} = \mu_{ij} \left[\frac{1}{2} v_{ij}^2 - G \frac{(m_i + m_j)}{r_{ij}} \right], \quad (4.35)$$

$$T_{ij} = \frac{1}{2} \mu_{ij} v_{ij}^2. \quad (4.36)$$

³ http://en.wikipedia.org/wiki/Message_Passing_Interface.

⁴ Anúncio oficial disponível na seguinte url:

<https://www.surf.nl/en/agenda/2014/03/surfsara-data--computing-infrastructure-event/surfsara-data-and-computing-infrastructure-event.html>.

⁵ Não confundir com o operador *kick* definido na seção anterior.

Além disso, nós definimos,

$$U_i = \sum_{j \neq i}^N U_{ij}, \quad (4.37)$$

$$W_i = \sum_{j \neq i}^N W_{ij}, \quad (4.38)$$

com

$$W_{ij} \equiv K_{ij} - T_{ij}. \quad (4.39)$$

Note que W_i e W_{ij} são analiticamente idênticos à U_i e U_{ij} , respectivamente, pois até o momento não fizemos nenhuma aproximação.

Considere o mapa arbitrário $\varphi(\tau)$ atuando sobre coordenadas de posição e velocidade. Nós o definimos explicitamente como (ver eqs. 2.32 e 2.33),

$$\varphi(\tau) : \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix} + \tau \frac{d}{dt} \begin{pmatrix} \mathbf{r} \\ \mathbf{v} \end{pmatrix}.$$

Com isso, nós podemos agora definir os seguintes operadores de evolução temporal. Primeiramente, operadores associados à grandezas com dois subscritos são definidos explicitamente por,

$$e^{\tau \mathcal{U}_{ij}} : \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} + \tau \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_{ij} \end{pmatrix}, \quad (4.40)$$

$$e^{\tau(-\mathcal{T}_{ij})} : \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} - \tau \begin{pmatrix} \mathbf{v}_{ij} \\ \mathbf{0} \end{pmatrix}, \quad (4.41)$$

$$e^{\tau \mathcal{K}_{ij}} : \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} \rightarrow \text{kepler_solver}(\tau, m_{ij}, \mathbf{r}_{ij}, \mathbf{v}_{ij}), \quad (4.42)$$

onde $m_{ij} = m_i + m_j$ e $\mathbf{a}_{ij} = -m_{ij}\mathbf{r}_{ij}/r_{ij}^3$ é a aceleração relativa entre o par ij . Na eq. 4.42 a função `kepler_solver` dá a solução analítica para o par ij , pelo intervalo τ . Dada a definição de W_{ij} (eq. 4.39), podemos escrever

$$e^{\tau \mathcal{W}_{ij}} \approx e^{\frac{\tau}{2}(-\mathcal{T}_{ij})} e^{\tau \mathcal{K}_{ij}} e^{\frac{\tau}{2}(-\mathcal{T}_{ij})}, \quad (4.43)$$

de modo que, explicitamente,

$$e^{\tau \mathcal{W}_{ij}} : \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{r}_{ij} \\ \mathbf{v}_{ij} \end{pmatrix} + \begin{pmatrix} \delta \mathbf{r}_{ij} \\ \delta \mathbf{v}_{ij} \end{pmatrix}, \quad (4.44)$$

onde $\delta \mathbf{r}_{ij}$ e $\delta \mathbf{v}_{ij}$ resultam da composição dos operadores $e^{\tau \mathcal{K}_{ij}}$ e $e^{\tau(-\mathcal{T}_{ij})}$ na eq. 4.43. Por outro lado, operadores associados à grandezas com um subscrito são definidos explicitamente por,

$$e^{\tau \mathcal{T}_i} : \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} + \tau \begin{pmatrix} \mathbf{v}_i \\ \mathbf{0} \end{pmatrix}, \quad (4.45)$$

$$e^{\tau \mathcal{U}_i} : \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} + \tau \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_i \end{pmatrix}, \quad (4.46)$$

$$e^{\tau \mathcal{W}_i} : \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix} + \begin{pmatrix} \delta \mathbf{r}_i \\ \delta \mathbf{v}_i \end{pmatrix}. \quad (4.47)$$

Resta agora mostrar como operadores com dois subscritos estão relacionados aos operadores com um subscrito. Em particular, nós buscamos uma relação entre $(\delta\mathbf{r}_{ij}, \delta\mathbf{v}_{ij})$ e $(\delta\mathbf{r}_i, \delta\mathbf{v}_i)$. Analisemos, primeiramente, o caso do operador \mathcal{U} . A partir da eq. 4.37 e dado que \mathcal{U}_{ij} comutam entre si, ou seja, $[\mathcal{U}_{ij}, \mathcal{U}_{kl}] = 0, \forall kl$, é fácil ver que a igualdade,

$$\prod_{j \neq i}^N e^{\tau \mathcal{U}_{ij}} = e^{\tau \sum_{j \neq i}^N \mathcal{U}_{ij}} = e^{\tau \mathcal{U}_i}, \quad (4.48)$$

é válida para qualquer N . Assim, os incrementos associados ao operador \mathcal{U} nas eqs. 4.40 e 4.46 ficam relacionados por,

$$\tau \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_i \end{pmatrix} = \frac{1}{m_i} \sum_{j \neq i}^N \mu_{ij} \tau \begin{pmatrix} \mathbf{0} \\ \mathbf{a}_{ij} \end{pmatrix}, \quad (4.49)$$

como esperado, visto que a aceleração \mathbf{a}_i é a superposição da contribuição de todos os pares, \mathbf{a}_{ij} . Assim, equações similares para o operador \mathcal{W} seguem por analogia, tendo em vista a eq. 4.38,

$$\prod_{j \neq i}^N e^{\tau \mathcal{W}_{ij} + \mathcal{O}(\tau^3)} \approx e^{\tau \sum_{j \neq i}^N \mathcal{W}_{ij}} = e^{\tau \mathcal{W}_i}, \quad (4.50)$$

onde o erro, $\mathcal{O}(\tau^3)$, advém do mapa na eq. 4.43, de modo que os incrementos $(\delta\mathbf{r}_{ij}, \delta\mathbf{v}_{ij})$ e $(\delta\mathbf{r}_i, \delta\mathbf{v}_i)$ nas eqs. 4.44 e 4.47 ficam relacionados por,

$$\begin{pmatrix} \delta\mathbf{r}_i \\ \delta\mathbf{v}_i \end{pmatrix} = \frac{1}{m_i} \sum_{j \neq i}^N \mu_{ij} \begin{pmatrix} \delta\mathbf{r}_{ij} \\ \delta\mathbf{v}_{ij} \end{pmatrix} + \mathcal{O}(\tau^3). \quad (4.51)$$

Dessa forma, nós podemos finalmente definir um mapa de segunda ordem para a nossa separação Kepleriana do Hamiltoniano como segue,

$$\begin{aligned} \text{sakura}(\tau) &\equiv e^{\frac{\tau}{2}\mathcal{T}} e^{\tau \mathcal{W}} e^{\frac{\tau}{2}\mathcal{T}}, \\ &= \left[\prod_{i=1}^N e^{\frac{\tau}{2}\mathcal{T}_i} \right] \circ \left[\prod_{i=1}^N e^{\tau \mathcal{W}_i} \right] \circ \left[\prod_{i=1}^N e^{\frac{\tau}{2}\mathcal{T}_i} \right]. \end{aligned} \quad (4.52)$$

Cabe agora uma discussão das propriedades do mapa acima. Consideremos, primeiramente, a eq. 4.50 em vista da eq. 4.43. Supondo que ao invés da função `kepler_solver`, o operador $e^{\tau \mathcal{K}_{ij}}$ fosse definido como,

$$e^{\tau \mathcal{K}_{ij}} \equiv e^{\frac{\tau}{2}\mathcal{T}_{ij}} e^{\tau \mathcal{U}_{ij}} e^{\frac{\tau}{2}\mathcal{T}_{ij}}, \quad (4.53)$$

para o par ij . Assim, a eq. 4.43 reduziria-se à

$$e^{\tau \mathcal{W}_{ij}} \approx e^{\frac{\tau}{2}(-\mathcal{T}_{ij})} \left(e^{\frac{\tau}{2}\mathcal{T}_{ij}} e^{\tau \mathcal{U}_{ij}} e^{\frac{\tau}{2}\mathcal{T}_{ij}} \right) e^{\frac{\tau}{2}(-\mathcal{T}_{ij})} \Rightarrow e^{\tau \mathcal{W}_{ij}} = e^{\tau \mathcal{U}_{ij}}. \quad (4.54)$$

Com isso o erro na eq. 4.50 seria nulo, e esta seria analiticamente idêntica à eq. 4.48. Em outras palavras, os incrementos $(\delta\mathbf{r}_{ij}, \delta\mathbf{v}_{ij})$ na eq. 4.51 seriam dados por $\delta\mathbf{r}_{ij} = 0$ e $\delta\mathbf{v}_{ij} = \tau \mathbf{a}_{ij}$, de modo que esta reduziria-se à eq. 4.49. Isto, porém, tornaria o mapa `sakura`(τ) na eq. 4.52 essencialmente idêntico à um mapa

ordinário do tipo ϕ_τ^2 (eq. 2.41), para não dizer que desta forma $\text{sakura}(\tau)$ teria sido uma implementação extremamente ineficiente de ϕ_τ^2 .

Por outro lado, com o operador $e^{\tau\mathcal{K}_{ij}}$ definido corretamente, fazendo uso da função `kepler_solve` como na eq. 4.42, os incrementos $(\delta\mathbf{r}_{ij}, \delta\mathbf{v}_{ij})$ contêm os termos exatos da solução do problema de Kepler, e assim nós garantimos que todas as interações entre pares ij são sempre evoluídas corretamente na precisão da máquina. Isto permite que $\text{sakura}(\tau)$ use passos de tempo constante para evoluir qualquer tipo de interação entre pares de partículas — mesmo uma colisão frontal — e ainda assim produza o resultado numérico esperado.

Finalmente, é preciso notar aqui que, embora nós tenhamos usado o formalismo simplético para a derivação do mapa na eq. 4.52, este resultou em um método não-simplético. Isto, à princípio, não é evidente das equações acima, mas nós encontramos numericamente que este é de fato o caso, visto que $\text{sakura}(\tau)$ não conserva o momentum angular total do sistema na precisão da máquina, como esperado de um mapa simplético (o erro, entretanto, é pequeno o suficiente para propósitos práticos e diminui com o tamanho do passo de tempo). Nós especulamos que a razão para a não-simpleticidade da eq. 4.52 está no fato de que, embora o erro na eq. 4.43 seja Hamiltoniano (à maneira definida no Cap. 2), a composição desta na eq. 4.50 (ou, equivalentemente, na eq. 4.51) resulta em um erro não-Hamiltoniano, degradando assim a simpleticidade do mapa final. Nós suspeitamos que isso seja devido ao fato de que, p. ex., na eq. 4.51, cada par ij seja tratado de maneira independente das demais partículas no sistema, de modo que em casos em que uma terceira partícula esteja interagindo proximamente com o par em questão, um erro não-Hamiltoniano seja introduzido. Este último argumento constitui, de fato, a principal fonte de erro em sakura .

Recentemente, contudo, um estudo por [Hernandez and Bertschinger \(2015\)](#) argumenta que é possível restaurar a simpleticidade de sakura com a decomposição do operador central na eq. 4.52 em duas partes evoluindo todos os pares em ordem direta (parte 1) e reversa (parte 2) no espaço de índices das partículas, mas nós ainda não investigamos tal modificação.

4.3 VALIDAÇÃO NUMÉRICA

Nesta seção nós apresentamos alguns testes numéricos comparando resultados obtidos com os seguintes integradores: `sakura`, `sia21a.dkd` (identificado nas figuras como “Leapfrog”), ambos métodos de segunda ordem, e o método de Hermite de quarta ordem ([Makino, 1991b](#)), comumente utilizado pela comunidade. Para simulações com `sakura` foi utilizado $\tau = cte$. `sia21a.dkd` e Hermite usam passos de tempo variáveis, porém `sia21a.dkd` adota o critério desenvolvido na sec. 4.1.1, enquanto Hermite usa o critério padrão de Aarseth (e. g., [Aarseth, 2003](#)). O valor do passo de tempo constante usado em simulações com `sakura` foi escolhido de modo tal que resulte no mesmo número de passos realizados nas integrações com Hermite. Para `sia21a.dkd`, o parâmetro de precisão η (ver eq. 4.8) foi escolhido de modo a dar um número de passos similar aos métodos anteriores. Note que em simulações de N -corpos é comum a utilização de um parâmetro de *softening*, ε , tal que o potencial gravitacional, $U \sim 1/r$,

seja modificado em $U \sim 1/(r_{ij}^2 + \varepsilon^2)^{1/2}$. O parâmetro de *softening* é necessário para evitar possíveis divergências numéricas quando do encontro próximo entre duas partículas quaisquer, facilitando assim o trabalho dos integradores. Porém aqui, nós utilizamos $\varepsilon = 0$ para `sia21a.dkd` e `Hermite`, uma vez que `sakura`, por ser baseado na solução analítica de uma órbita Kepleriana, não admite (ou necessita) um parâmetro de *softening*. Todas as simulações foram realizadas utilizando apenas uma CPU (modelo *Intel core 2 Duo*, 2GHz).

4.3.1 Sistemas com poucas partículas

Nós começamos apresentando testes numéricos para sistemas com N -pequeno com soluções bem conhecidas, incluindo i) sistema `figure8` ($N = 3$, [Chenciner and Montgomery, 2000](#)), ii) sistema `pythagorean` ($N = 3$, [Szebehely and Peters, 1967](#)), iii) sistema `solarsystem` ($N = 10$ incluindo Plutão, [Ito and Tanikawa, 2002](#)). No sistema `figure8` as três partículas possuem massas iguais e seguem uma órbita estável em formato de 8. O sistema `pythagorean` é um sistema instável, onde as três partículas possuem uma razão de massas 3 : 4 : 5, inicialmente com velocidades nulas e posicionadas num triângulo cujos lados tem tamanho dado pela massa da partícula no vértice diametralmente oposto. O sistema `solarsystem`, constitui um exemplo de sistema quase-Kepleriano. Nós não discutimos o caso $N = 2$, uma vez que aqui `sakura` reduz-se à função `kepler_solver`, dando resultados exatos (na precisão da máquina). A duração das simulações é de 100 *N-body time units* ([Heggie and Mathieu, 1986](#)) no caso dos dois primeiros sistemas, e 10^3 yr no caso do `solarsystem`.

Na Fig. 4.1 nós apresentamos o erro relativo da energia total, $|dE/E|$, como uma função do tamanho do passo de tempo médio, $\langle\tau\rangle$, (painéis à esquerda) e o tempo de CPU vs $|dE/E|$ (painéis à direita) para `figure8` (painéis superiores), `pythagorean` (painéis do meio) e `solarsystem` (painéis inferiores). Nós notamos que para `figure8`, `Hermite` é mais preciso para níveis de conservação da energia $\lesssim 10^{-6}$. Nós atribuímos isto ao fato que para este sistema o tamanho do passo variável das partículas não muda significativamente durante a evolução orbital, e assim, para menores $\langle\tau\rangle$, `Hermite` converge mais rapidamente com $\mathcal{O}(\tau^4)$, do que `sakura` e `sia21a.dkd` (`Leapfrog`), que convergem com $\mathcal{O}(\tau^2)$. Não obstante, `sakura` é ligeiramente mais preciso que `sia21a.dkd` (`Leapfrog`) para todo o intervalo em $\langle\tau\rangle$.

Quando todas as 3 partículas interagem democraticamente como no sistema `pythagorean` (painéis do meio), espera-se que `sakura` não seja o método mais adequado para integração de tal sistema devido ao fato da não comutatividade das interações. À despeito disso, a Fig. 4.1 (painéis do meio) mostra que ainda assim, `sakura` é ligeiramente mais preciso que `sia21a.dkd` (`Leapfrog`), embora `Hermite` ainda seja mais eficiente para uma faixa em $\langle\tau\rangle = 10^{-5} - 10^{-3}$. Para `solarsystem`, em que a evolução orbital dos planetas é quase kepleriana, `sakura` fornece melhor conservação da energia por cerca de 4 ordens de magnitude quando comparado à `sia21a.dkd` (`Leapfrog`), sendo também mais preciso que `Hermite` para grande parte do intervalo em $\langle\tau\rangle$, a despeito das diferentes ordens de convergência.

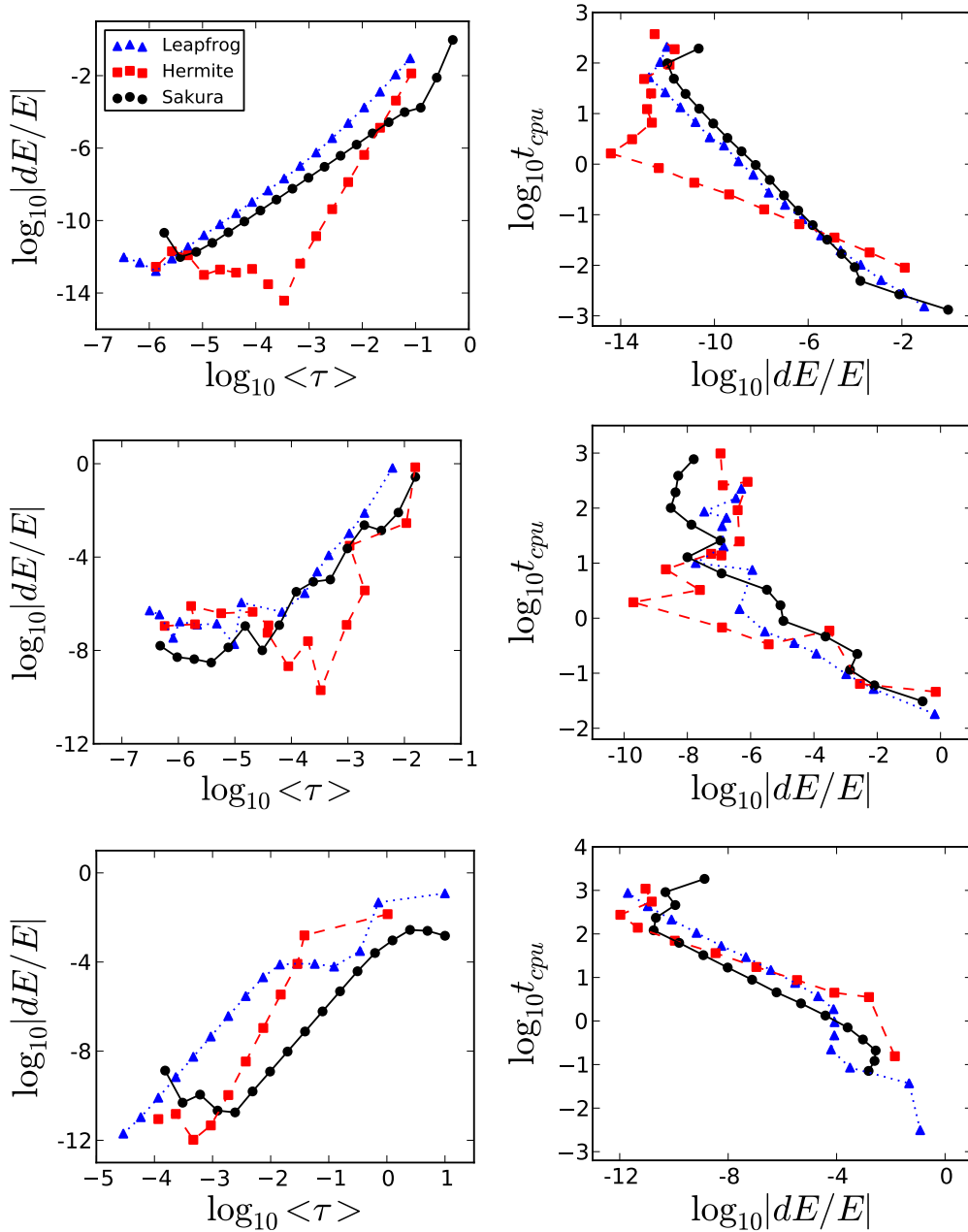


Figura 4.1: A figura mostra o erro relativo da energia total como função do tamanho médio do passo de tempo (painéis à esquerda) e o tempo de CPU como função do erro da energia (painéis à direita) para três sistemas: figure8 (painéis superiores), pythagorean (painéis do meio) e solarsystem (painéis inferiores). Resultados com sakura são mostrados em linhas cheias com círculos pretos, sia21a.dkd (Leapfrog) em linhas pontilhadas com triângulos azuis e Hermite em linhas tracejadas com quadrados vermelhos. As grandezas estão em N -body units e o tempo de CPU é dado em segundos.

Para passos de tempo muito pequenos, $\langle\tau\rangle \lesssim 10^{-4}$, todos os métodos apresentam erros devido à precisão finita da máquina, de modo que estes tornam-se comparáveis, independentemente da ordem de convergência do integrador. Cada passo de tempo em sakura é, tipicamente, um fator 2 – 4 computacio-

nalmente mais caro que nos outros métodos, devido ao fato de que temos que resolver a equação de Kepler para todos os pares de partículas. Contudo, uma vez que todos os integradores escalam com $O(N^2)$ e como *sakura* não requer uma diminuição do passo de tempo para integrar encontros próximos entre partículas, resulta que menos passos de tempo são necessários com *sakura*, o que em certos casos faz com que este seja mais eficiente que os demais métodos de integração comumente utilizados pela comunidade.

Afim de confirmar o argumento acima nós incluímos um teste adicional, usando como condições iniciais um sistema de 4-corpos com duas binárias interagindo em órbita circular, construídas de maneira hierárquica. Neste sistema, uma binária circular é originalmente construída com semi-eixo maior $a_{\text{outer}} = 1$ (*N-body units*). Cada partícula é então substituída hierarquicamente por um outro sistema binário menor, que herda os parâmetros da partícula originária, de acordo com o procedimento descrito no final da seção 3.3.3. O semi-eixo maior das binárias secundárias, a_{inner} , é então escolhido de modo que a razão $a_{\text{outer}}/a_{\text{inner}}$ seja dada por um dos valores na faixa [10, 100, 1000]. Desta forma, uma simulação da evolução orbital deste sistema é realizada durante P_{outer} , ou seja, o maior período orbital no sistema, no qual as binárias secundárias realizam várias revoluções orbitais.

Na Fig. 4.2 nós apresentamos o erro relativo da energia total, $|dE/E|$, como uma função de $\langle\tau\rangle$ (painéis à esquerda) e o tempo de CPU vs $|dE/E|$ (painéis à direita) para os diferentes esquemas de integração. Para o caso $a_{\text{outer}}/a_{\text{inner}} = 10$ (painéis superiores), *sakura* fornece o mesmo nível de conservação de energia que *sia21a.dkd* (Leapfrog), embora sendo ligeiramente menos eficiente. Por outro lado, *Hermite* conserva melhor a energia devido à sua convergência de ordem mais alta para $\langle\tau\rangle \lesssim 10^{-2}$. Para os casos $a_{\text{outer}}/a_{\text{inner}} = 100$ (painéis do meio) e $a_{\text{outer}}/a_{\text{inner}} = 1000$ (painéis inferiores) em que as partículas que formam as binárias interagem cada vez mais proximamente, *sakura* demonstra ter uma performance cada vez melhor com a presença de binárias mais compactas. Em particular, para um nível de conservação da energia de 10^{-6} , tipicamente adotado em simulações de *N*-corpos, *sakura* é mais que uma ordem de magnitude mais rápido que *Hermite* para o sistema com binárias mais compactas, embora tenha velocidade similar à *sia21a.dkd* (Leapfrog). Quando comparado em termos do tamanho do passo de tempo, *sakura* é mais preciso que os demais métodos por uma faixa em $\langle\tau\rangle$ de ~ 5 ordens de magnitude. Por outro lado, no caso mais compacto, os resultados com *Hermite* só começam a convergir com $\langle\tau\rangle \lesssim 10^{-5.5}$, o que em algumas circunstâncias pode ser impraticável em termos computacionais, quando sistemas deste tipo estão presentes em simulações de grande porte.

4.3.2 Sistemas com muitas partículas

Para testar como *sakura* se comporta com um problema de *N*-corpos mais geral, nós usamos como condições iniciais uma distribuição de Plummer com 128-corpos contendo um SMBH em seu centro. Nós assumimos massas iguais para as estrelas e construímos o modelo em equilíbrio virial, para diferentes razões de massa entre SMBH e estrelas, $q \equiv M_{\text{BH}}/M_{\text{star}}$, variando de $q = 1$ (sem

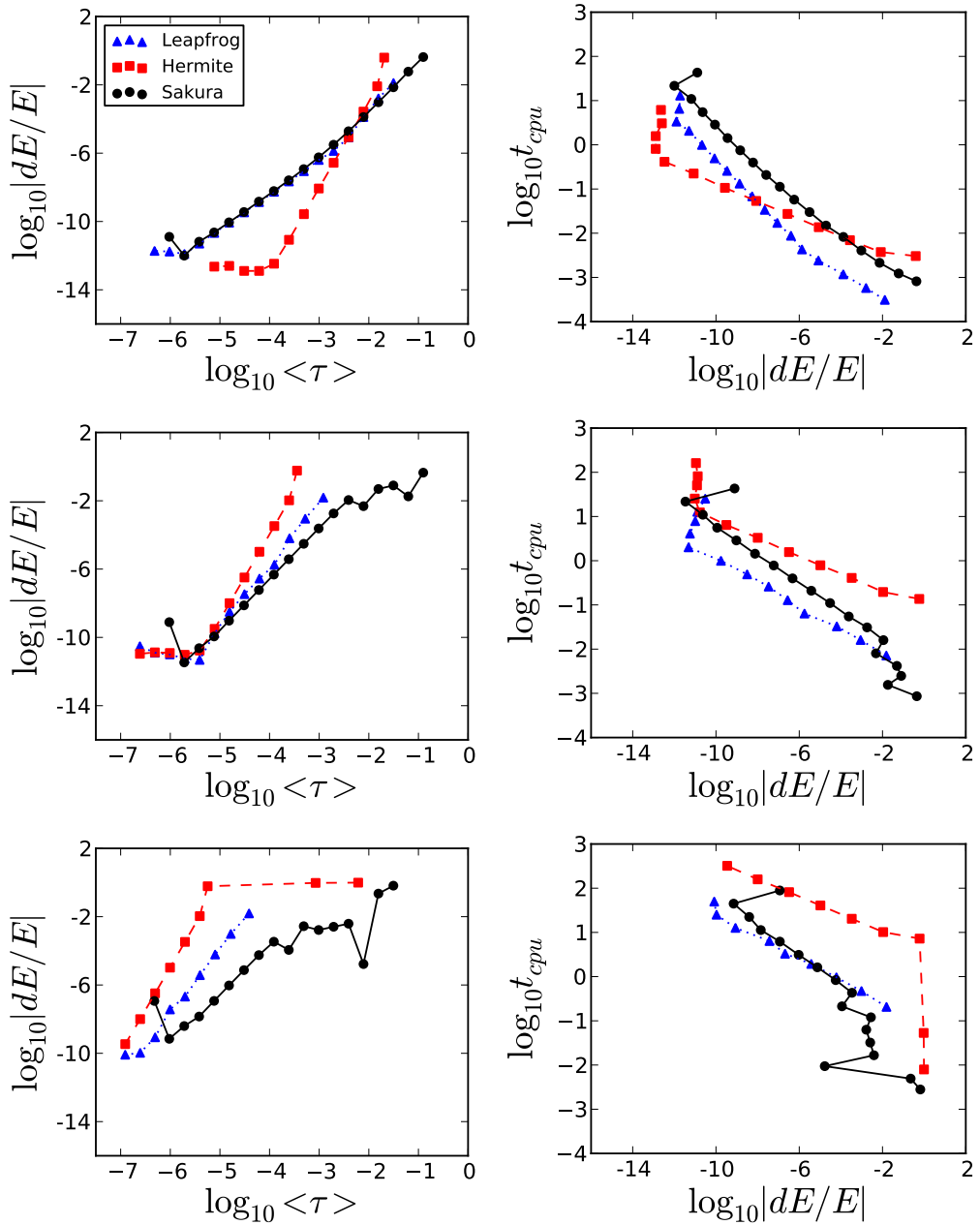


Figura 4.2: O mesmo que Fig. 4.1, mas para o sistema hierárquico de binárias com as seguintes razões de semi-eixo maior: $a_{outer}/a_{inner} = 10$ (painéis superiores), $a_{outer}/a_{inner} = 100$ (painéis do meio) e $a_{outer}/a_{inner} = 1000$ (painéis inferiores).

SMBH) à $q = 10^{12}$. Nós realizamos simulações para cada uma dessas condições iniciais por 1 N -body time unit. Como na seção anterior, sakura é comparado com sia21a.dkd (Leapfrog) e Hermite.

Na Fig. 4.3 nós apresentamos os resultados de tais simulações na forma do erro relativo da energia, $|dE/E|$, como função da razão de massas, para diferentes tamanhos do passo de tempo na faixa $\langle\tau\rangle = [10^{-3}, 10^{-4}, 10^{-5}]$ (linhas superiores, do meio e inferiores, respectivamente). O painel à esquerda compara sakura e sia21a.dkd (Leapfrog) e o painel à direita compara sakura e Hermite.

Nós vemos na Fig. 4.3 que $|dE/E|$, para todos os três métodos, inicialmente

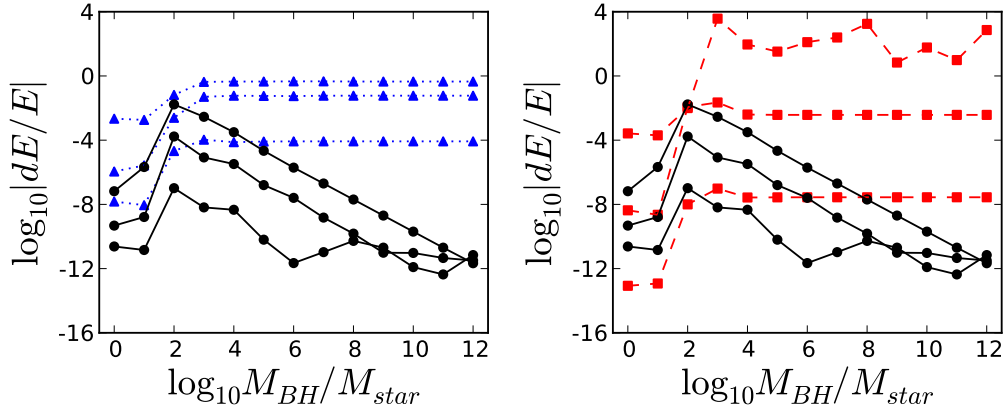


Figura 4.3: A figura mostra o erro relativo da energia total como função da razão de massas entre SMBH e estrelas, para três tamanhos do passo de tempo $\langle\tau\rangle = [10^{-3}, 10^{-4}, 10^{-5}]$ (linhas superiores, do meio e inferiores, respectivamente). sakura é comparado com sia21a.dkd (Leapfrog) (painel à esquerda) e Hermite (painel à direita). O esquema de cores é o mesmo da Fig. 4.1.

aumenta com a razão de massas até o ponto em que $q \sim 10^2$. Para razões de massa maiores, o comportamento de sakura difere claramente dos outros dois métodos. Enquanto em sia21a.dkd (Leapfrog) e Hermite $|dE/E|$ estabiliza em torno de um certo valor para $q > 10^2$, com sakura nós observamos que $|dE/E|$ diminui com o aumento da razão de massas. Em outras palavras, sakura torna-se cada vez mais preciso (e, conseqüentemente, mais eficiente) quando a razão de massas cresce, como pode-se ver na Fig. 4.4, que apresenta o tempo de CPU vs $|dE/E|$ para diferentes razões de massa: $q = 10^3$ (painel superior esquerdo), $q = 10^6$ (painel superior direito), $q = 10^9$ (painel inferior esquerdo) e $q = 10^{12}$ (painel inferior direito).

O comportamento observado na Fig. 4.3 é interpretado como segue. Quando não há uma partícula massiva dominante no sistema ($q \sim 1 - 10$), depois de decorrido 1 *N-body time unit* o sistema não teve tempo suficiente para evoluir dinamicamente e formar binárias (que são consequência de fortes interações de poucos-corpos, ver p. ex., Tanikawa et al., 2012). Como consequência, nestas circunstâncias a maior parte das partículas interagem fracamente entre elas, e todos os três métodos são aptos à integrar o sistema com relativamente boa conservação da energia. Em torno de uma razão de massas $q \sim 10 - 10^3$ a partícula massiva rapidamente forma um sistema binário com algum vizinho próximo, que eventualmente experimenta vários encontros próximos e perturbações de partículas no seu entorno, deteriorando assim a precisão da integração em todos os três métodos. Para razões de massa $q \gtrsim 10^3$ o movimento orbital das partículas é dominado pelo corpo massivo, tornando-se predominantemente Kepleriano. Neste regime, as orbitas no sistema tornam-se regulares, e encontros próximos entre estrelas são cada vez menos importantes. Como consequência, espera-se que o erro na energia convirja ao erro associado à cada um dos métodos de integração utilizados. Assim, em sia21a.dkd (Leapfrog) e Hermite a conservação da energia melhora com o decréscimo do passo de tempo, mas

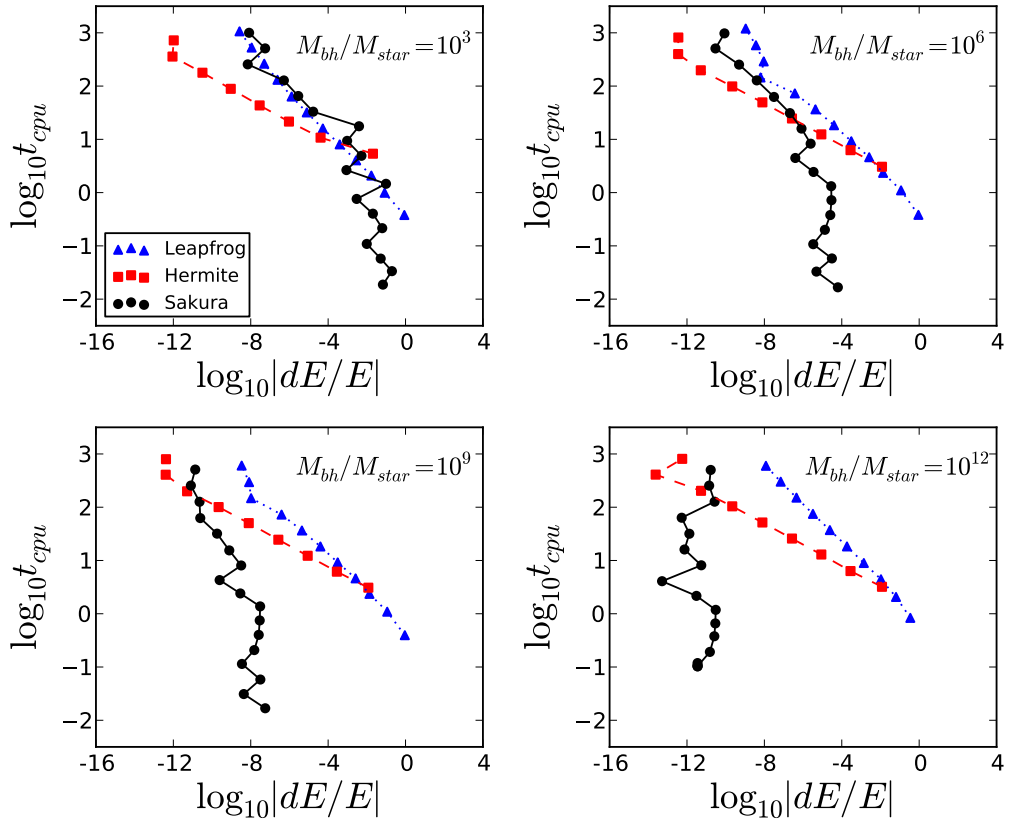


Figura 4.4: Para o mesmo sistema da Fig. 4.3, os painéis mostram o tempo de CPU como função do erro da energia para as seguintes razões de massa: $q = 10^3$ (superior esquerdo), $q = 10^6$ (superior direito), $q = 10^9$ (inferior esquerdo) and $q = 10^{12}$ (inferior direito).

permanece no mesmo nível de conservação a despeito da razão de massas (para $q \gtrsim 10^3$). sakura, por outro lado, torna-se cada vez mais preciso com a razão de massas. O erro em sakura vem de duas fontes: i) da função `kepler_solve` que é essencialmente na precisão da máquina, e ii) da não comutatividade das interações de 2-corpos durante encontros próximos de múltiplos pares. Assim, no regime em que $q \gtrsim 10^3$, o erro em sakura converge ao erro da função `kepler_solve`, visto que neste limite o movimento orbital das partículas é praticamente Kepleriano e perturbações de outras partículas são cada vez menos importantes.

Além disso, nós observamos na Fig. 4.4 que, para uma razão de massas $q = 10^3$, a performance de sakura é similar àquela de `sia21a.dkd` (Leapfrog), com Hermite sendo mais eficiente para integrações mais precisas. Com o aumento da razão de massas, a eficiência de sakura aumenta, e nós notamos também uma mudança na inclinação das curvas para sakura (linhas com círculos pretos). Isso indica que a partir de uma certa razão de massas, digamos $q = 10^6$, quando o sistema torna-se dominado por orbitas aproximadamente keplerianas, sakura é capaz de realizar a integração de forma precisa ($dE/E \sim 10^{-6} - 10^{-12}$), mesmo com o uso de passos de tempo relativamente grandes (t_{CPU} pequenos), salvando assim uma enorme quantidade ($\sim 3 - 4$

ordens de magnitude) de tempo computacional quando comparado com integrações com `sia21a.dkd` (Leapfrog) e Hermite.

Como um teste adicional, nós realizamos uma simulação da evolução dinâmica de um sistema de 1024-corpos até o momento do *core collapse*. Neste sistema as partículas possuem massas iguais e estão inicialmente distribuídas seguindo um perfil de densidade de Plummer em equilíbrio virial. A partir destas condições iniciais nós então evoluímos o sistema com sakura, `sia21a.dkd` (Leapfrog) e Hermite. Em Hermite nós escolhemos o valor usual para o parâmetro de precisão $\eta_{\text{Hermite}} \approx 0.03$, que tipicamente mantém níveis de conservação da energia $dE/E \sim 10^{-4}$ até o momento do *core collapse*. Para `sia21a.dkd` (Leapfrog) o parâmetro de precisão foi ajustado de modo que a conservação da energia fosse similar àquela de Hermite. Para sakura nós usamos passo de tempo constante, mas com diferentes tamanhos na faixa $\tau = [10^0, 10^{-1}, 10^{-2}, 10^{-4}]$.

Na Fig. 4.5 nós apresentamos a evolução temporal do raio do *core* para os diferentes esquemas de integração. Nós vemos que para tamanhos do passo de tempo suficientemente pequenos ($\tau \sim 10^{-4}$, curva preta mais inferior), sakura é apto a evoluir o sistema através do *core collapse*, que para esse sistema ocorre em torno de $t_{cc} \sim 400$ *N-body units*. O *core collapse* é um fenômeno bem conhecido que ocorre na evolução dinâmica de um sistema de *N*-corpos, devido ao processo de relaxação de dois corpos. Neste processo, trocas energéticas entres

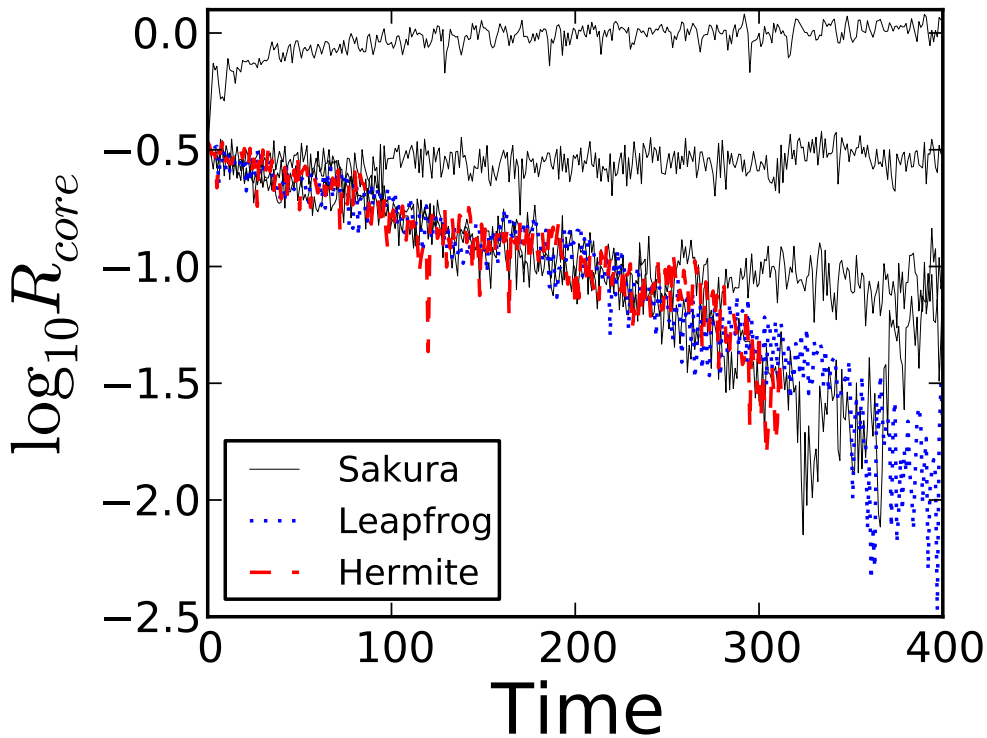


Figura 4.5: A figura mostra a evolução temporal do raio do *core* de um aglomerado com 1024-corpos inicialmente em equilíbrio virial. Nós comparamos sakura usando diferentes tamanhos do passo de tempo $\tau = [10^0, 10^{-1}, 10^{-2}, 10^{-4}]$ (linhas pretas de cima para baixo), com `sia21a.dkd` (Leapfrog) (linha pontilhada azul) e Hermite (linha tracejada vermelha), ambos usando passo de tempo variável. As grandezas estão em *N-body units*.

as partículas em regiões de maior densidade, levam à ejeção de algumas partículas do sistema. Os sub-sistemas que permanecem no core do aglomerado, por sua vez tornam-se gradativamente mais compactos, até que eventualmente binárias, ou sub-sistemas com poucas partículas extremamente compactos são formados dinamicamente, levando assim à uma posterior reexpansão do core. A evolução à longo prazo desse sistema se caracteriza por eventuais re-colapsos e reexpansões do core do aglomerado. O momento do primeiro colapso é que define o t_{cc} , cujo valor é dependente do número de partículas no sistema e é estimado como $t_{cc} \sim 10t_{rh}$ (ver eq. 2.16). Como esperado da instabilidade orbital exponencial (Goodman et al., 1993), resultados dos três métodos de integração diferem nos detalhes da evolução. À despeito disso, a fenomenologia que acabamos de descrever é de fato observada na Fig. 4.5, em especial para a linha preta mais inferior, correspondente ao cálculo realizado com sakura (para $\tau \sim 10^{-4}$).

Com sakura, o surgimento de binárias compactas no momento do *core collapse* não representa um desafio computacional. Consequentemente, não se observa perda de desempenho, como é o caso de outros métodos comumente utilizados. Dessa forma, a simulação mais cara usando sakura ($\tau \sim 10^{-4}$, curva preta mais inferior), foi completada em cerca de 3 dias de tempo de CPU. Simulações com sia21a.dkd (Leapfrog) e Hermite, levaram cerca de uma semana e um mês de tempo de CPU, respectivamente, sendo que a simulação com Hermite teve que ser interrompida antes de ser completada, devido a formação dinâmica de uma binária extremamente compacta e a consequente diminuição do tamanho do passo de tempo adaptativo à níveis que praticamente congelaram a integração. Enquanto tal fenômeno é, tipicamente, prevenido com a introdução de um parâmetro de *softening*, aqui nós usamos $\varepsilon = 0$. De qualquer modo, este teste mostra que a aparição de uma tal binária extremamente compacta pode acontecer e representa um sério desafio computacional. Com sakura nós colocamos agora uma solução para tal desafio!

Embora sakura resolva as interações entre todos os pares de partículas exatamente, a presença de perturbações devido à vizinhos próximos representa a principal fonte de erro durante a integração. O motivo para isso origina-se da suposição de que cada par de partículas possa ser tratado como um problema independente por certo intervalo de tempo, τ . Se τ é maior que a escala de tempo de interação do par ij com o vizinho k , a perturbação devido à k será retardada por $\Delta t \sim \tau$, levando à uma integração espúria de tal sub-sistema. Isto é uma consequência da não-comutatividade das interações de 2-corpos que compõe o operador $e^{\tau\mathcal{V}}$ na eq. 4.52. Assim, o uso de passos de tempo relativamente grandes na Fig. 4.5 revela esse problema. Ainda assim, embora o sistema como um todo permaneça ligado, é claro da Fig. 4.5 que para $\tau \gtrsim 10^{-2}$, fortes interações entres múltiplos pares no *core* do aglomerado não são corretamente integradas e, como consequência o raio do *core* expande. Contudo, à medida que se usa um τ pequeno o suficiente o problema acima torna-se irrelevante e, neste limite, sakura converge para a solução esperada. A questão de quão pequeno τ é pequeno o suficiente neste contexto, é algo que nós estamos ainda investigando, juntamente com a possibilidade de introduzir um critério para variabilidade do passo de tempo em sakura.

4.4 PARALELIZAÇÃO: *Win Nieuwpoort prize*

sakura foi originalmente implementado de maneira serial. Nós então decidimos investigar sua eficiência sob diferentes esquemas de paralelização. Dessa forma nós implementamos mais duas versões: i) utilizando `OpenCL` para execução em `GPUs`, e ii) utilizando `MPI` para execução em sistemas com memória distribuída. A versão utilizando `GPUs` mostrou-se pouco eficiente devido a presença de muitos *if statements*⁶ na função `kepler_solver`. Isto, em princípio, pode ser resolvido reescrevendo-se a função `kepler_solver` de forma vetorizada, mas nós ainda não o fizemos.

Aqui nós focamos na versão utilizando `MPI`. Nós adaptamos o método *j-parallelization* descrito em [Portegies Zwart et al. \(2008\)](#), de modo que cada processo paralelo possui uma cópia idêntica do sistema de partículas. Os testes consistem da evolução de uma distribuição de Plummer com N -partículas de massas iguais em equilíbrio virial, integradas com sakura por 1 N -body unit. Nós variamos N na faixa $1k \equiv 1024$ à $1M \equiv 1024^2 = 1048576$, e medimos o tempo de `CPU` (*Wall Clock Time*), $T_{CPU}(p)$, ao utilizar p -cores de processamento, com p variando de 1 à 6144. Os resultados são apresentados na forma do *strong scaling* ($T_{CPU}(p)$ vs p) e da eficiência paralela,

$$\text{Efficiency} \equiv \frac{T_{CPU}(p)}{pT_{CPU}(1)}. \quad (4.55)$$

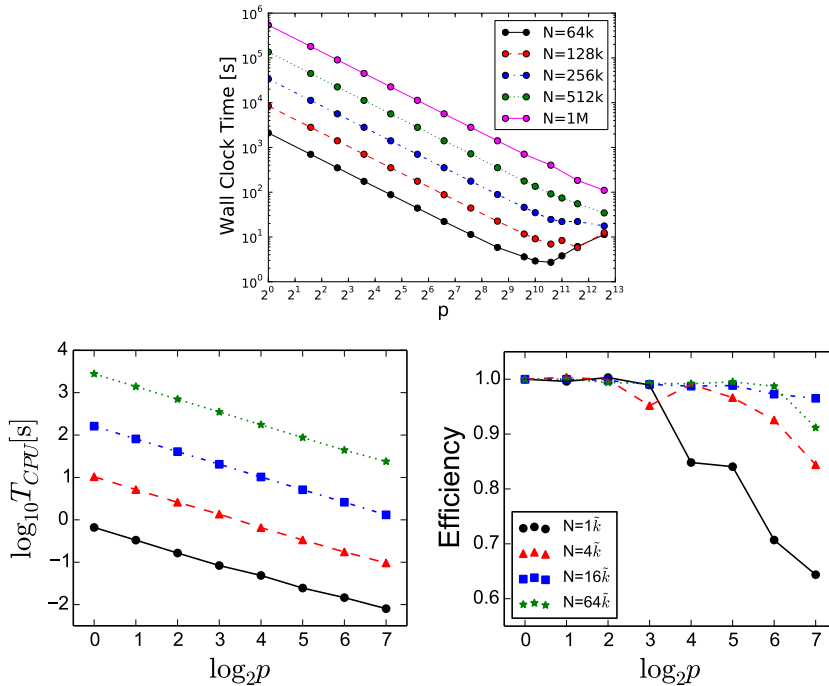


Figura 4.6: Painel superior: *strong scaling* para valores de $N \geq 64k$ com p até 6144 *cores* de processamento. Painéis inferiores: *strong scaling* (esquerda) e eficiência paralela (direita) para valores de $N \leq 64k$ com p até 128 *cores*.

⁶ Essa é uma das peculiaridades da programação em `GPUs`: elas perdem eficiência paralela quando pelo menos uma das *threads* de execução diverge das demais.

Como se vê na Fig. 4.6, sakura exibe excelente *strong scaling* (painel superior) para valores de $N \geq 64k$ com até 6144 *cores* de processamento. Nos painéis inferiores nós mostramos o *strong scaling* e a eficiência paralela para valores de $N \leq 64k$ com até 128 *cores*. Nós vemos que mesmo nos casos em que o número de partículas por *core* é relativamente pequeno, a eficiência paralela ainda permanece próxima de 100% e, no pior caso ($N = 1k$, $p = 128$), onde tem-se apenas 8 partículas por *core* a eficiência paralela é ainda de $\sim 64\%$.

Os resultados desta seção foram obtidos utilizando o supercomputador Cartesius no centro Holandês para HPC, SURFsara. Por ter alcançado alta eficiência paralela em um grande sistema para HPC, nós fomos agraciados em 13 de Março de 2014 com o prêmio Win Nieuwpoort⁷ (veja Ap. B), oferecido pelo SURFsara em honra ao professor emérito Win Nieuwpoort, pioneiro da área de supercomputação na Holanda.

⁷ Anúncio oficial disponível na seguinte url:

<https://www.surf.nl/en/agenda/2014/03/surfsara-data--computing-infrastructure-event/surfsara-data-and-computing-infrastructure-event.html>.

Parte III

CONCLUSÕES

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth

5

CONCLUSÕES E PERSPECTIVAS

Nós começamos o nosso doutorado com o intuito de estender o nosso trabalho de mestrado (Gonçalves Ferrari, 2010; Ferrari et al., 2013) com uma investigação numérica das implicações do recuo gravitacional devido à emissão anisotrópica de ondas gravitacionais, focado especificamente no caso da galáxia M 83 (Dottori et al., 2010). No entanto, nós percebemos que os métodos e códigos de simulação disponíveis não contemplavam os requerimentos necessários para tal estudo. O nosso código, inicialmente desenvolvido no mestrado, era então o único que implementava todos os termos de correções PNs, porém os métodos de dinâmica gravitacional foram implementados de maneira muito simplificada e numericamente frágil. Nós decidimos assim, focar o nosso trabalho na construção de um código robusto e cientificamente competitivo, para então atacar os problemas de interesse astrofísico.

Com isso, o principal foco deste trabalho foi no desenvolvimento de novos métodos para simulações gravitacionais de sistemas de N -corpos. Contrário à metodologias prévias (e. g., Aarseth, 2003), que em grande parte são baseados em “receitas” numéricas, no presente trabalho nós fazemos uso de princípios físicos gerais, baseados na teoria Hamiltoniana e no formalismo canônico. Como tal, os métodos desenvolvidos aqui são genéricos e podem ser aplicados à qualquer sistema físico cuja evolução temporal é governada pela eq. 2.23.

Neste trabalho nós apresentamos essencialmente três contribuições para a área de astrofísica computacional:

1. Nós generalizamos e obtemos uma família de 48 novos mapas simpléticos com ordens $\mathcal{O}(\tau^2)$, $\mathcal{O}(\tau^4)$ e $\mathcal{O}(\tau^6)$, usando diferentes esquemas de passo de tempo, baseado em uma subdivisão recursiva do Hamiltoniano, primeiramente introduzida por Pelupessy et al. (2012). Durante seu desenvolvimento nós prestamos atenção especial ao caso com forças dependentes da velocidade, com vistas à inclusão de correções PN advindas da GR de maneira auto-consistente e temporalmente simétrica. Nossa formulação generaliza assim o método BRIDGE (Fujii et al., 2007) ao que nós chamamos de BRIDGE-hierárquico, permitindo o acoplamento hierárquico

de diferentes sub-sistemas ou domínios astrofísicos, contemplando diferentes escalas de tempo de interação.

2. Nós introduzimos uma nova subdivisão Hamiltoniana em termos de N^2 problemas de Kepler (Gonçalves Ferrari et al., 2014). Com este novo método, o problema gravitacional de N -corpos pode ser resolvido como uma composição de problemas de 2-corpos independentes. Dessa forma, cada par de partículas no sistema é automaticamente resolvido exatamente com a solução da equação de Kepler, permitindo que sub-sistemas compactos possam ser evoluídos sem grandes dificuldades numéricas durante uma simulação de grande porte. O método é particularmente robusto em simulações com sistemas hierárquicos ou sistemas em que a dinâmica das partículas é dominada por uma partícula massiva, tal como um SMBH.
3. Nossas ideias, algoritmos e métodos desenvolvidos neste trabalho, dentre outras coisas, estão implementados no código tupan. Seu *design* foi motivado por uma reavaliação das técnicas empregadas tradicionalmente e algumas mudanças de paradigma se fizeram necessárias. tupan é em grande parte escrito em python, com *backends* em C e OpenCL, para fácil acesso ao poder computacional de GPUs. tupan tem como premissas a simplicidade, elegância e a robustez dos algoritmos que o compõe e está livremente disponível para uso em <https://github.com/ggf84/tupan>.

Com relação às nossas perspectivas, nós temos como principal projeto a criação de um ambiente de *software* totalmente integrado para realização e análise de simulações astrofísicas de N -corpos. tupan constitui atualmente uma das bases deste projeto, implementando métodos gerais para *dinâmica gravitacional*. As outras duas bases necessárias para modelagem realística de qualquer sistema astrofísico constituem os domínios da (*magneto*-)*hidrodinâmica* e da *formação e evolução estelar*. Para além disso, existem ainda os sub-domínios da *cosmologia*, *transferência radiativa* e *evolução química*, por exemplo, que em certos casos podem representar fatores de grande importância na evolução de sistemas auto-gravitantes.

Há ainda alguns passos que precisam ser completados na implementação dos métodos para *dinâmica gravitacional* em tupan antes de podermos seguir adiante com a implementação/acoplamento de outros domínios astrofísicos. Por exemplo, a subdivisão *slow/fast* do Hamiltoniano apresentada na seção 4.1 não é a única possível, nem a mais eficiente para sistemas hierárquicos com múltiplos sub-sistemas (veja, p. ex., Jänes et al., 2014). Além disso, nossos desenvolvimentos com sakura certamente podem ser melhorados como apontados por Hernandez and Bertschinger (2015). Após isso e outras melhorias técnicas já planejadas, tupan v1.0 será apresentado à comunidade científica com a publicação de um *paper*. tupan v2.0 virá com a implementação do domínio da (*magneto*-)*hidrodinâmica*, seguido por *evolução estelar* (tupan v3.0) e assim por diante.

Por fim, faz-se necessário notar que a linha de pesquisa em astrofísica computacional é bastante promissora, especialmente tendo em vista os enormes avanços em poder computacional que tem-se alcançado em anos recentes. Isto coloca a pesquisa em astrofísica computacional em uma situação oposta àquela de algumas décadas atrás. Atualmente, qualquer *workstation* de última geração

conta com uma GPU cujo poder computacional é comparável ao de um *cluster* de computadores modesto. Dadas as circunstâncias, o *bottleneck* para o crescimento desta área está agora na formação de recursos humanos e no desenvolvimento de métodos e códigos capazes de modelar realisticamente o sistema astrofísico em estudo. Desta forma, nós posicionamos nosso projeto de pesquisa no sentido de contribuir para sanar tal *bottleneck*, com a construção de um ambiente de *software* que servirá como base para futura formação de recursos humanos, produção científica e eventuais desenvolvimentos que resultem deste processo.

Parte IV

APÊNDICES

A

FORMA EXPLÍCITA DOS TERMOS PÓS-NEWTONIANOS

As páginas que seguem, retiradas de [Itoh \(2004a\)](#) e [Itoh \(2009\)](#), mostram todos os termos de correções [PN](#) em sua forma explícita. Estas equações estão implementadas em nosso código tupan.

Note that this redefinition of the representative points does not affect the existence of the energy conservation, as was shown by Eq. (8.6). We can examine the effect of this redefinition onto the equation of motion using Eq. (8.5) (use $\delta_{A \ln}^i$ instead of $\delta_{A\Theta}^i$). Thence we have

$$\begin{aligned} m_1 a_1^i |_{\delta_{A \ln}} &= -\epsilon^6 \frac{3m_1 \delta_2^k \ln}{r_{12}^3} n_{12}^{(ik)} + \epsilon^6 \frac{3m_2 \delta_1^k \ln}{r_{12}^3} n_{12}^{(ik)} - \epsilon^6 \frac{d^2 \delta_1^i \ln}{d\tau^2} \\ &= -\frac{44m_1^4 m_2}{3r_{12}^5} n_{12}^i \ln\left(\frac{r_{12}}{\epsilon R_1}\right) + \frac{44m_1^2 m_2^3}{3r_{12}^5} n_{12}^i \ln\left(\frac{r_{12}}{\epsilon R_2}\right) \\ &\quad - \frac{22m_1^3 m_2}{r_{12}^4} [5(\vec{n}_{12} \cdot \vec{V})^2 n_{12}^i - V^2 n_{12}^i - 2(\vec{n}_{12} \cdot \vec{V}) \\ &\quad \times V^i] \ln\left(\frac{r_{12}}{\epsilon R_1}\right) + \frac{22m_1^3 m_2}{3r_{12}^4} \left(\frac{m_1}{r_{12}} n_{12}^i + \frac{m_2}{r_{12}} n_{12}^i \right. \\ &\quad \left. - V^2 n_{12}^i + 8(\vec{n}_{12} \cdot \vec{V})^2 n_{12}^i - 2(\vec{n}_{12} \cdot \vec{V}) V^i\right). \quad (11.2) \end{aligned}$$

Comparing the above equation with Eq. (10.4), we easily conclude that the representative point z_A^i of the star A defined by

$$\begin{aligned} D_{A\Theta, \text{New}}^i(\tau) &= \epsilon^{-6} \int_{B_A} d^3 y [y^i - z_A^i(\tau)] \Theta_N^{\tau\tau}(\tau, y^k) \\ &= \epsilon^4 \delta_A^i(\tau) \end{aligned} \quad (11.3)$$

obeys an equation of motion free from logarithms and hence free from any ambiguity up to 3PN order inclusively.

We mention here that Blanchet and Faye [40] have already noticed that in their 3PN equation of motion a suitable coordinate transformation removes (parts of) logarithmic dependences of arbitrary parameters corresponding (roughly) to our body zone radii.⁷ It is well known that choosing different values of dipole moments corresponds to the coordinate transformation.

By adding $m_1 a_1^i |_{\delta_{A \ln}}$ to Eq. (10.4), we obtain our 3PN equation of motion for two spherical compact stars whose representative points are defined by Eq. (11.3),

$$\begin{aligned} m_1 \frac{dv_1^i}{d\tau} &= -\frac{m_1 m_2}{r_{12}^2} n_{12}^i + \epsilon^2 \frac{m_1 m_2}{r_{12}^2} n_{12}^i \left[-v_1^2 - 2v_2^2 + 4(\vec{v}_1 \cdot \vec{v}_2) + \frac{3}{2}(\vec{n}_{12} \cdot \vec{v}_2)^2 + \frac{5m_1}{r_{12}} + \frac{4m_2}{r_{12}} \right] \\ &\quad + \epsilon^2 \frac{m_1 m_2}{r_{12}^2} V^i [4(\vec{n}_{12} \cdot \vec{v}_1) - 3(\vec{n}_{12} \cdot \vec{v}_2)] + \epsilon^4 \frac{m_1 m_2}{r_{12}^2} n_{12}^i \left[-2v_2^4 + 4v_2^2(\vec{v}_1 \cdot \vec{v}_2) - 2(\vec{v}_1 \cdot \vec{v}_2)^2 + \frac{3}{2}v_1^2(\vec{n}_{12} \cdot \vec{v}_2)^2 \right. \\ &\quad \left. + \frac{9}{2}v_2^2(\vec{n}_{12} \cdot \vec{v}_2)^2 - 6(\vec{v}_1 \cdot \vec{v}_2)(\vec{n}_{12} \cdot \vec{v}_2)^2 - \frac{15}{8}(\vec{n}_{12} \cdot \vec{v}_2)^4 - \frac{57}{4} \frac{m_1^2}{r_{12}^2} - 9 \frac{m_2^2}{r_{12}^2} - \frac{69}{2} \frac{m_1 m_2}{r_{12}^2} \right. \\ &\quad \left. + \frac{m_1}{r_{12}} \left(-\frac{15}{4}v_1^2 + \frac{5}{4}v_2^2 - \frac{5}{2}(\vec{v}_1 \cdot \vec{v}_2) + \frac{39}{2}(\vec{n}_{12} \cdot \vec{v}_1)^2 - 39(\vec{n}_{12} \cdot \vec{v}_1)(\vec{n}_{12} \cdot \vec{v}_2) + \frac{17}{2}(\vec{n}_{12} \cdot \vec{v}_2)^2 \right) \right. \\ &\quad \left. + \frac{m_2}{r_{12}} [4v_2^2 - 8(\vec{v}_1 \cdot \vec{v}_2) + 2(\vec{n}_{12} \cdot \vec{v}_1)^2 - 4(\vec{n}_{12} \cdot \vec{v}_1)(\vec{n}_{12} \cdot \vec{v}_2) - 6(\vec{n}_{12} \cdot \vec{v}_2)^2] \right] \\ &\quad + \epsilon^4 \frac{m_1 m_2}{r_{12}^2} V^i \left[\frac{m_1}{r_{12}} \left(-\frac{63}{4}(\vec{n}_{12} \cdot \vec{v}_1) + \frac{55}{4}(\vec{n}_{12} \cdot \vec{v}_2) \right) + \frac{m_2}{r_{12}} [-2(\vec{n}_{12} \cdot \vec{v}_1) - 2(\vec{n}_{12} \cdot \vec{v}_2)] + v_1^2(\vec{n}_{12} \cdot \vec{v}_2) \right. \\ &\quad \left. + 4v_2^2(\vec{n}_{12} \cdot \vec{v}_1) - 5v_2^2(\vec{n}_{12} \cdot \vec{v}_2) - 4(\vec{v}_1 \cdot \vec{v}_2)(\vec{n}_{12} \cdot \vec{V}) - 6(\vec{n}_{12} \cdot \vec{v}_1)(\vec{n}_{12} \cdot \vec{v}_2)^2 + \frac{9}{2}(\vec{n}_{12} \cdot \vec{v}_2)^3 \right] \\ &\quad + \epsilon^5 \frac{4m_1^2 m_2}{5r_{12}^3} \left[n_{12}^i(\vec{n}_{12} \cdot \vec{V}) \left(-6 \frac{m_1}{r_{12}} + \frac{52}{3} \frac{m_2}{r_{12}} + 3V^2 \right) + V^i \left(2 \frac{m_1}{r_{12}} - 8 \frac{m_2}{r_{12}} - V^2 \right) \right] \\ &\quad + \epsilon^6 \frac{m_1 m_2}{r_{12}^2} n_{12}^i \left[\frac{35}{16}(\vec{n}_{12} \cdot \vec{v}_2)^6 - \frac{15}{8}(\vec{n}_{12} \cdot \vec{v}_2)^4 v_1^2 + \frac{15}{2}(\vec{n}_{12} \cdot \vec{v}_2)^4 (\vec{v}_1 \cdot \vec{v}_2) + 3(\vec{n}_{12} \cdot \vec{v}_2)^2 (\vec{v}_1 \cdot \vec{v}_2)^2 - \frac{15}{2}(\vec{n}_{12} \cdot \vec{v}_2)^4 v_2^2 \right] \end{aligned}$$

⁷Unlike our case, their coordinate transformation does not remove the logarithmic dependences of their free parameters completely. The remaining logarithmic dependence was used to make their equation of motion conservative.

$$\begin{aligned}
& + \frac{3}{2}(\vec{n}_{12} \cdot \vec{v}_2)^2 v_1^2 v_2^2 - 12(\vec{n}_{12} \cdot \vec{v}_2)^2 (\vec{v}_1 \cdot \vec{v}_2) v_2^2 - 2(\vec{v}_1 \cdot \vec{v}_2)^2 v_2^2 + \frac{15}{2}(\vec{n}_{12} \cdot \vec{v}_2)^2 v_2^4 + 4(\vec{v}_1 \cdot \vec{v}_2) v_2^4 - 2v_6^6 \\
& + \frac{m_1}{r_{12}} \left(-\frac{171}{8}(\vec{n}_{12} \cdot \vec{v}_1)^4 + \frac{171}{2}(\vec{n}_{12} \cdot \vec{v}_1)^3 (\vec{n}_{12} \cdot \vec{v}_2) - \frac{723}{4}(\vec{n}_{12} \cdot \vec{v}_1)^2 (\vec{n}_{12} \cdot \vec{v}_2)^2 + \frac{383}{2}(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^3 \right. \\
& - \frac{455}{8}(\vec{n}_{12} \cdot \vec{v}_2)^4 + \frac{229}{4}(\vec{n}_{12} \cdot \vec{v}_1)^2 v_1^2 - \frac{205}{2}(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) v_1^2 + \frac{191}{4}(\vec{n}_{12} \cdot \vec{v}_2)^2 v_1^2 - \frac{91}{8}v_1^4 \\
& - \frac{229}{2}(\vec{n}_{12} \cdot \vec{v}_1)^2 (\vec{v}_1 \cdot \vec{v}_2) + 244(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) - \frac{225}{2}(\vec{n}_{12} \cdot \vec{v}_2)^2 (\vec{v}_1 \cdot \vec{v}_2) + \frac{91}{2}v_1^2 (\vec{v}_1 \cdot \vec{v}_2) - \frac{177}{4}(\vec{v}_1 \cdot \vec{v}_2)^2 \\
& \left. + \frac{229}{4}(\vec{n}_{12} \cdot \vec{v}_1)^2 v_2^2 - \frac{283}{2}(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) v_2^2 + \frac{259}{4}(\vec{n}_{12} \cdot \vec{v}_2)^2 v_2^2 - \frac{91}{4}v_1^2 v_2^2 + 43(\vec{v}_1 \cdot \vec{v}_2) v_2^2 - \frac{81}{8}v_2^4 \right) \\
& + \frac{m_2}{r_{12}} \left[-6(\vec{n}_{12} \cdot \vec{v}_1)^2 (\vec{n}_{12} \cdot \vec{v}_2)^2 + 12(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^3 + 6(\vec{n}_{12} \cdot \vec{v}_2)^4 + 4(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) \right. \\
& + 12(\vec{n}_{12} \cdot \vec{v}_2)^2 (\vec{v}_1 \cdot \vec{v}_2) + 4(\vec{v}_1 \cdot \vec{v}_2)^2 - 4(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) v_2^2 - 12(\vec{n}_{12} \cdot \vec{v}_2)^2 v_2^2 - 8(\vec{v}_1 \cdot \vec{v}_2) v_2^2 + 4v_2^4 \\
& \left. + \frac{m_2^2}{r_{12}^2} \left(-(\vec{n}_{12} \cdot \vec{v}_1)^2 + 2(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) + \frac{43}{2}(\vec{n}_{12} \cdot \vec{v}_2)^2 + 18(\vec{v}_1 \cdot \vec{v}_2) - 9v_2^2 \right) \right] \\
& + \frac{m_1 m_2}{r_{12}^2} \left(\frac{415}{8}(\vec{n}_{12} \cdot \vec{v}_1)^2 - \frac{375}{4}(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) + \frac{1113}{8}(\vec{n}_{12} \cdot \vec{v}_2)^2 \right. \\
& \left. - \frac{615\pi^2}{64}(\vec{n}_{12} \cdot \vec{V})^2 + 18v_1^2 + \frac{123\pi^2}{64}V^2 + 33(\vec{v}_1 \cdot \vec{v}_2) - \frac{33}{2}v_2^2 \right) \\
& + \frac{m_1^2}{r_{12}^2} \left(-\frac{2069}{8}(\vec{n}_{12} \cdot \vec{v}_1)^2 + 543(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) - \frac{939}{4}(\vec{n}_{12} \cdot \vec{v}_2)^2 + \frac{471}{8}v_1^2 - \frac{357}{4}(\vec{v}_1 \cdot \vec{v}_2) + \frac{357}{8}v_2^2 \right) \\
& + \frac{16m_2^3}{r_{12}^3} + \frac{m_1^2 m_2}{r_{12}^3} \left(\frac{547}{3} - \frac{41\pi^2}{16} \right) - \frac{13m_1^3}{12r_{12}^3} + \frac{m_1 m_2^2}{r_{12}^3} \left(\frac{545}{3} - \frac{41\pi^2}{16} \right) \\
& + \epsilon^6 \frac{m_1 m_2}{r_{12}^2} V^i \left[\frac{15}{2}(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^4 - \frac{45}{8}(\vec{n}_{12} \cdot \vec{v}_2)^5 - \frac{3}{2}(\vec{n}_{12} \cdot \vec{v}_2)^3 v_1^2 + 6(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^2 (\vec{v}_1 \cdot \vec{v}_2) \right. \\
& - 6(\vec{n}_{12} \cdot \vec{v}_2)^3 (\vec{v}_1 \cdot \vec{v}_2) - 2(\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2)^2 - 12(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^2 v_2^2 + 12(\vec{n}_{12} \cdot \vec{v}_2)^3 v_2^2 \\
& + (\vec{n}_{12} \cdot \vec{v}_2) v_1^2 v_2^2 - 4(\vec{n}_{12} \cdot \vec{v}_1) (\vec{v}_1 \cdot \vec{v}_2) v_2^2 + 8(\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) v_2^2 + 4(\vec{n}_{12} \cdot \vec{v}_1) v_2^4 - 7(\vec{n}_{12} \cdot \vec{v}_2) v_2^4 \\
& + \frac{m_2}{r_{12}} \left[-2(\vec{n}_{12} \cdot \vec{v}_1)^2 (\vec{n}_{12} \cdot \vec{v}_2) + 8(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^2 + 2(\vec{n}_{12} \cdot \vec{v}_2)^3 \right. \\
& \left. + 2(\vec{n}_{12} \cdot \vec{v}_1) (\vec{v}_1 \cdot \vec{v}_2) + 4(\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) - 2(\vec{n}_{12} \cdot \vec{v}_1) v_2^2 - 4(\vec{n}_{12} \cdot \vec{v}_2) v_2^2 \right] \\
& + \frac{m_1}{r_{12}} \left(-\frac{243}{4}(\vec{n}_{12} \cdot \vec{v}_1)^3 + \frac{565}{4}(\vec{n}_{12} \cdot \vec{v}_1)^2 (\vec{n}_{12} \cdot \vec{v}_2) - \frac{269}{4}(\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^2 - \frac{95}{12}(\vec{n}_{12} \cdot \vec{v}_2)^3 + \frac{207}{8}(\vec{n}_{12} \cdot \vec{v}_1) v_1^2 \right. \\
& \left. - \frac{137}{8}(\vec{n}_{12} \cdot \vec{v}_2) v_1^2 - 36(\vec{n}_{12} \cdot \vec{v}_1) (\vec{v}_1 \cdot \vec{v}_2) + \frac{27}{4}(\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) + \frac{81}{8}(\vec{n}_{12} \cdot \vec{v}_1) v_2^2 + \frac{83}{8}(\vec{n}_{12} \cdot \vec{v}_2) v_2^2 \right) \\
& + \frac{m_2^2}{r_{12}^2} \left[4(\vec{n}_{12} \cdot \vec{v}_1) + 5(\vec{n}_{12} \cdot \vec{v}_2) \right] + \frac{m_1 m_2}{r_{12}^2} \left(-\frac{307}{8}(\vec{n}_{12} \cdot \vec{v}_1) + \frac{479}{8}(\vec{n}_{12} \cdot \vec{v}_2) + \frac{123\pi^2}{32}(\vec{n}_{12} \cdot \vec{V}) \right) \\
& \left. + \frac{m_1^2}{r_{12}^2} \left(\frac{311}{4}(\vec{n}_{12} \cdot \vec{v}_1) - \frac{357}{4}(\vec{n}_{12} \cdot \vec{v}_2) \right) \right] + O(\epsilon^7), \tag{11.4}
\end{aligned}$$

$$\begin{aligned}
(m_1 a_1^i)_{\leq 3.5 \text{PN}} = & (m_1 a_1^i)_{\leq 3 \text{PN}} + \frac{m_1^4 m_2}{r_{12}^5} \left[n_{12}^i \left\{ \frac{3992}{105} (\vec{n}_{12} \cdot \vec{v}_1) - \frac{4328}{105} (\vec{n}_{12} \cdot \vec{v}_2) \right\} - \frac{184}{21} V^i \right] + \frac{m_1^3 m_2^2}{r_{12}^5} \left[\frac{6224}{105} V^i \right. \\
& + n_{12}^i \left\{ \frac{2872}{21} (\vec{n}_{12} \cdot \vec{v}_2) - \frac{13576}{105} (\vec{n}_{12} \cdot \vec{v}_1) \right\} \left. \right] + \frac{m_1^3 m_2}{r_{12}^4} V^i \left[-\frac{132}{35} v_1^2 - \frac{48}{35} v_2^2 + \frac{52}{15} (\vec{n}_{12} \cdot \vec{v}_1)^2 \right. \\
& + \frac{152}{35} (\vec{v}_1 \cdot \vec{v}_2) - \frac{56}{15} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) - \frac{44}{15} (\vec{n}_{12} \cdot \vec{v}_2)^2 \left. \right] + \frac{m_1^3 m_2}{r_{12}^4} n_{12}^i \left[-\frac{4888}{105} (\vec{n}_{12} \cdot \vec{v}_1) v_1^2 \right. \\
& + \frac{5056}{105} (\vec{n}_{12} \cdot \vec{v}_2) v_1^2 - \frac{1028}{21} v_2^2 (\vec{n}_{12} \cdot \vec{v}_1) + 48 (\vec{n}_{12} \cdot \vec{v}_1)^3 + \frac{5812}{105} v_2^2 (\vec{n}_{12} \cdot \vec{v}_2) + \frac{2056}{21} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{v}_1 \cdot \vec{v}_2) \\
& - \frac{2224}{21} (\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) - \frac{696}{5} (\vec{n}_{12} \cdot \vec{v}_1)^2 (\vec{n}_{12} \cdot \vec{v}_2) + \frac{744}{5} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^2 - \frac{288}{5} (\vec{n}_{12} \cdot \vec{v}_2)^3 \left. \right] \\
& + \frac{m_1^2 m_2^3}{r_{12}^5} \left[\frac{6388}{105} V^i - \frac{3172}{21} (\vec{n}_{12} \cdot \vec{V}) n_{12}^i \right] + \frac{m_1^2 m_2}{r_{12}^3} V^i \left[\frac{334}{35} v_1^4 + \frac{654}{35} v_2^2 v_1^2 - \frac{1336}{35} (\vec{v}_1 \cdot \vec{v}_2) v_1^2 + \frac{292}{35} v_2^4 \right. \\
& - \frac{348}{5} (\vec{n}_{12} \cdot \vec{v}_1)^2 V^2 + 60 (\vec{n}_{12} \cdot \vec{V})^4 - \frac{1252}{35} v_2^2 (\vec{v}_1 \cdot \vec{v}_2) + \frac{684}{5} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) V^2 + \frac{1308}{35} (\vec{v}_1 \cdot \vec{v}_2)^2 \\
& - 66 (\vec{n}_{12} \cdot \vec{v}_2)^2 V^2 \left. \right] + \frac{m_1^2 m_2}{r_{12}^3} n_{12}^i \left[-\frac{246}{35} (\vec{n}_{12} \cdot \vec{V}) v_1^4 - \frac{534}{35} v_2^2 (\vec{n}_{12} \cdot \vec{v}_1) v_1^2 + \frac{90}{7} v_2^2 (\vec{n}_{12} \cdot \vec{v}_2) v_1^2 \right. \\
& + \frac{1068}{35} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{v}_1 \cdot \vec{v}_2) v_1^2 - \frac{984}{35} (\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) v_1^2 - \frac{204}{35} v_2^4 (\vec{n}_{12} \cdot \vec{v}_1) + 60 (\vec{n}_{12} \cdot \vec{v}_1)^3 V^2 \\
& - 56 (\vec{n}_{12} \cdot \vec{V})^5 + \frac{24}{7} v_2^4 (\vec{n}_{12} \cdot \vec{v}_2) + \frac{984}{35} v_2^2 (\vec{n}_{12} \cdot \vec{v}_1) (\vec{v}_1 \cdot \vec{v}_2) - \frac{732}{35} v_2^2 (\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) \\
& - 180 (\vec{n}_{12} \cdot \vec{v}_1)^2 (\vec{n}_{12} \cdot \vec{v}_2) V^2 - \frac{1068}{35} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{v}_1 \cdot \vec{v}_2)^2 + 174 (\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^2 V^2 \\
& + \frac{180}{7} (\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2)^2 - 54 V^2 (\vec{n}_{12} \cdot \vec{v}_2)^3 \left. \right] + \frac{m_1^2 m_2^2}{r_{12}^4} V^i \left[-\frac{152}{21} v_1^2 - \frac{1768}{105} v_2^2 + \frac{454}{15} (\vec{n}_{12} \cdot \vec{v}_1)^2 \right. \\
& + \frac{2864}{105} (\vec{v}_1 \cdot \vec{v}_2) - \frac{372}{5} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2) + \frac{854}{15} (\vec{n}_{12} \cdot \vec{v}_2)^2 \left. \right] + \frac{m_1^2 m_2^2}{r_{12}^4} n_{12}^i \left[\frac{1432}{35} (\vec{n}_{12} \cdot \vec{v}_1) v_2^2 \right. \\
& - \frac{5752}{105} (\vec{n}_{12} \cdot \vec{v}_2) v_2^2 - \frac{582}{5} (\vec{n}_{12} \cdot \vec{v}_1)^3 + \frac{3568}{105} (\vec{n}_{12} \cdot \vec{V}) v_1^2 - \frac{2864}{35} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{v}_1 \cdot \vec{v}_2) \\
& \left. + \frac{10048}{105} (\vec{n}_{12} \cdot \vec{v}_2) (\vec{v}_1 \cdot \vec{v}_2) + \frac{1746}{5} (\vec{n}_{12} \cdot \vec{v}_1)^2 (\vec{n}_{12} \cdot \vec{v}_2) - \frac{1954}{5} (\vec{n}_{12} \cdot \vec{v}_1) (\vec{n}_{12} \cdot \vec{v}_2)^2 + 158 (\vec{n}_{12} \cdot \vec{v}_2)^3 \right], \quad (65)
\end{aligned}$$

where the acceleration up to the 3 PN order is given in Paper III.

Equation (65) is in perfect agreement with the previous works in harmonic coordinates [34,38], the result in the ADMTT coordinate [10,15] by a suitable gauge transformation, and also the results from the energy balance argument [16–18]. We have used the local conservation law of the stress energy tensor of the matter and the gravitational field and the surface integral approach to derive our 3.5 PN equations of motion. We have not *a priori* assumed that the star follows a geodesic in any sense. The strong field point particle limit enables us to realize a point particle with strong internal gravity without using a Dirac delta functional. Nissanke *et al.* [38] assumed that a star follows a geodesic regularized by the Hadamard Parti Finie regularization (or any other regularization method that gives the same result, such as the dimensional regularization). Thereby, the perfect agreement between our present work and that work [38] confirms that a self-gravitating star

follows the regularized geodesic at least up to the 3.5 PN order inclusively.

ACKNOWLEDGMENTS

I am grateful to the anonymous referee who carefully read the original manuscript and kindly gave comments that have substantially improved this paper. This paper is a part of the outcome of the Japan Society of the Promotion of Science (JSPS) Global Center of Excellence (COE) Program (G01): Weaving Science Web beyond Particle-Matter Hierarchy at Tohoku University, Japan. Extensive use of the algebraic computation software programs Mathematica and MathTensor has been made.

APPENDIX A: χ PART

This section shows the functional expressions of $P_{A\chi}^\tau$ in terms of m_A , v_A^i , $V^i = v_1^i - v_2^i$, and r_{12}^i . Here we defined

WIN NIEUWPOORT AWARD



Figura B.1: A figura mostra o diploma referente ao prêmio Win Nieuwpoort, recebido na cerimônia de premiação em 13 de Março de 2014 em Amsterdam.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] S. J. Aarseth. Dynamical evolution of clusters of galaxies, I. *Monthly Notices of the Royal Astronomical Society*, 126:223–+, 1963. (Citado na página 10.)
- [2] S. J. Aarseth. *Gravitational N-Body Simulations*. November 2003. (Citado nas páginas 11, 22, 26, 44 e 57.)
- [3] J. Barnes and P. Hut. A hierarchical $O(N \log N)$ force-calculation algorithm. *Nature*, 324:446–449, December 1986. doi: 10.1038/324446a0. (Citado na página 4.)
- [4] J. Binney and S. Tremaine. *Galactic Dynamics: Second Edition*. Princeton University Press, 2008. (Citado nas páginas 12 e 13.)
- [5] L. Blanchet. Gravitational Radiation from Post-Newtonian Sources and Inspiral-ing Compact Binaries. *Living Reviews in Relativity*, 9:4–+, June 2006. (Citado na página 18.)
- [6] L. Blanchet and G. Faye. On the equations of motion of point-particle binaries at the third post-Newtonian order. *Physics Letters A*, 271:58–64, June 2000a. doi: 10.1016/S0375-9601(00)00360-1. (Citado na página 17.)
- [7] L. Blanchet and G. Faye. Hadamard regularization. *Journal of Mathematical Physics*, 41:7675–7714, November 2000b. doi: 10.1063/1.1308506. (Citado na página 17.)
- [8] L. Blanchet and G. Faye. General relativistic dynamics of compact binaries at the third post-Newtonian order. *Physical Review D*, 63(6):062005–+, March 2001a. doi: 10.1103/PhysRevD.63.062005. (Citado na página 17.)
- [9] L. Blanchet and G. Faye. Lorentzian regularization and the problem of point-like particles in general relativity. *Journal of Mathematical Physics*, 42:4391–4418, September 2001b. doi: 10.1063/1.1384864. (Citado na página 17.)
- [10] L. Blanchet and B. R. Iyer. Third post-Newtonian dynamics of compact binaries: equations of motion in the centre-of-mass frame. *Classical and Quantum Gravity*, 20:755–776, February 2003. (Citado na página 17.)
- [11] L. Blanchet, T. Damour, and G. Esposito-Farèse. Dimensional regularization of the third post-Newtonian dynamics of point particles in harmonic coordinates. *Physical Review D*, 69(12):124007–+, June 2004. doi: 10.1103/PhysRevD.69.124007. (Citado na página 17.)
- [12] S. Blanes and P. C. Moan. Practical symplectic partitioned Runge-Kutta and Runge-Kutta-Nyström methods. *Journal of Computational and Applied Mathematics*, 142:313–330, May 2002. doi: 10.1016/S0377-0427(01)00492-7. (Citado na página 39.)

- [13] M. Campanelli, C. Lousto, Y. Zlochower, and D. Merritt. Large Merger Recoils and Spin Flips from Generic Black Hole Binaries. *Astrophysical Journal Letters*, 659:L5–L8, April 2007. doi: 10.1086/516712. (Citado na página 16.)
- [14] F. Casas and A. Murua. An efficient algorithm for computing the Baker-Campbell-Hausdorff series and some of its applications. *Journal of Mathematical Physics*, 50(3):033513–+, March 2009. doi: 10.1063/1.3078418. (Citado na página 15.)
- [15] G. Chabrier. The Galactic Disk Mass Function: Reconciliation of the Hubble Space Telescope and Nearby Determinations. *Astrophysical Journal Letters*, 586: L133–L136, April 2003. doi: 10.1086/374879. (Citado na página 26.)
- [16] G. Chabrier. The Initial Mass Function: from Salpeter 1955 to 2005. In E. Corbelli, F. Palla, & H. Zinnecker, editor, *The Initial Mass Function 50 Years Later*, volume 327 of *Astrophysics and Space Science Library*, page 41, January 2005. (Citado na página 26.)
- [17] J. E. Chambers. A hybrid symplectic integrator that permits close encounters between massive bodies. *Monthly Notices of the Royal Astronomical Society*, 304:793–799, April 1999. doi: 10.1046/j.1365-8711.1999.02379.x. (Citado na página 33.)
- [18] S. Chandrasekhar. Dynamical Friction. I. General Considerations: the Coefficient of Dynamical Friction. *Astrophysical Journal*, 97:255–+, March 1943. doi: 10.1086/144517. (Citado na página 12.)
- [19] A. Chenciner and R. Montgomery. A remarkable periodic solution of the three-body problem in the case of equal masses. *ArXiv Mathematics e-prints*, October 2000. (Citado na página 45.)
- [20] Y.-Z. Chu. n-body problem in general relativity up to the second post-Newtonian order from perturbative field theory. *Physical Review D*, 79(4): 044031–+, February 2009. doi: 10.1103/PhysRevD.79.044031. (Citado na página 19.)
- [21] T. Damour, P. Jaranowski, and G. Schäfer. Poincaré invariance in the ADM Hamiltonian approach to the general relativistic two-body problem. *Physical Review D*, 62(2):021501–+, July 2000. doi: 10.1103/PhysRevD.62.021501. (Citado na página 17.)
- [22] T. Damour, P. Jaranowski, and G. Schäfer. Erratum: Poincaré invariance in the ADM Hamiltonian approach to the general relativistic two-body problem [Phys. Rev. D 62, 021501(R) (2000)]. *Physical Review D*, 63(2):029903–+, January 2001a. doi: 10.1103/PhysRevD.63.029903. (Citado na página 17.)
- [23] T. Damour, P. Jaranowski, and G. Schäfer. Equivalence between the ADM-Hamiltonian and the harmonic-coordinates approaches to the third post-Newtonian dynamics of compact binaries. *Physical Review D*, 63(4):044021–+, February 2001b. doi: 10.1103/PhysRevD.63.044021. (Citado na página 17.)

- [24] T. Damour, P. Jaranowski, and G. Schäfer. Dimensional regularization of the gravitational interaction of point masses. *Physics Letters B*, 513:147–155, July 2001c. doi: 10.1016/S0370-2693(01)00642-6. (Citado na página 17.)
- [25] T. Damour, P. Jaranowski, and G. Schäfer. Erratum: Equivalence between the ADM-Hamiltonian and the harmonic-coordinates approaches to the third post-Newtonian dynamics of compact binaries [Phys. Rev. D 63, 044021 (2001)]. *Physical Review D*, 66(2):029901–+, July 2002. doi: 10.1103/PhysRevD.66.029901. (Citado na página 17.)
- [26] V. C. de Andrade, L. Blanchet, and G. Faye. Third post-Newtonian dynamics of compact binaries: Noetherian conserved quantities and equivalence between the harmonic-coordinate and ADM-Hamiltonian formalisms. *Classical and Quantum Gravity*, 18:753–778, March 2001. doi: 10.1088/0264-9381/18/5/301. (Citado na página 17.)
- [27] H. Dottori, R. J. Díaz, J. F. Albacete-Colombo, and D. Mast. The Missing Goliath’s Slingshot: Massive Black Hole Recoil at M83. *Astrophysical Journal Letters*, 717:L42–L46, July 2010. doi: 10.1088/2041-8205/717/1/L42. (Citado na página 57.)
- [28] M. J. Duncan, H. F. Levison, and M. H. Lee. A Multiple Time Step Symplectic Algorithm for Integrating Close Encounters. *Astronomical Journal*, 116:2067–2077, October 1998. doi: 10.1086/300541. (Citado nas páginas 33 e 35.)
- [29] G. Efstathiou, M. Davis, S. D. M. White, and C. S. Frenk. Numerical techniques for large cosmological N-body simulations. *Astrophysical Journal Supplement Series*, 57:241–260, February 1985. doi: 10.1086/191003. (Citado na página 4.)
- [30] A. Einstein, L. Infeld, and B. Hoffmann. The Gravitational Equations and the Problem of Motion. *Annals of Mathematics*, 39:65–100, January 1938. (Citado nas páginas 17 e 19.)
- [31] N. T. Faber, D. Stibbe, S. Portegies Zwart, S. L. W. McMillan, and C. M. Boily. Particle-based sampling of N-body simulations. *Monthly Notices of the Royal Astronomical Society*, 401:1898–1903, January 2010. doi: 10.1111/j.1365-2966.2009.15775.x. (Citado na página 25.)
- [32] G. G. Ferrari, H. Dottori, and R. J. Díaz. Modeling the Black Hole Recoil from the Nucleus of M83. *Journal of Modern Physics*, 4:55–63, 2013. doi: 10.4236/jmp.2013.47A1007. (Citado nas páginas ix e 57.)
- [33] E. Forest and R. D. Ruth. Fourth-order symplectic integration. *Physica D Non-linear Phenomena*, 43:105–117, May 1990. doi: 10.1016/0167-2789(90)90019-L. (Citado na página 37.)
- [34] M. Fujii, M. Iwasawa, Y. Funato, and J. Makino. BRIDGE: A Direct-Tree Hybrid N-Body Algorithm for Fully Self-Consistent Simulations of Star Clusters and Their Parent Galaxies. *Publications of the Astronomical Society of Japan*, 59:1095–, December 2007. (Citado nas páginas 6, 33, 37 e 57.)

- [35] M. Giersz and D. C. Heggie. Statistics of N-Body Simulations - Part One - Equal Masses Before Core Collapse. *Monthly Notices of the Royal Astronomical Society*, 268:257–+, May 1994. (Citado na página 12.)
- [36] B. Gladman, M. Duncan, and J. Candy. Symplectic integrators for long-term integrations in celestial mechanics. *Celestial Mechanics and Dynamical Astronomy*, 52:221–240, September 1991. doi: 10.1007/BF00048485. (Citado na página 35.)
- [37] G. Gonçalves Ferrari. Recuo Gravitacional na Fusão de Buracos Negros em Centros Galácticos: O Caso de NGC 5236. Master's thesis, Universidade Federal do Rio Grande do Sul, 2010. (Citado nas páginas 18, 23 e 57.)
- [38] G. Gonçalves Ferrari, T. Boekholt, and S. F. Portegies Zwart. A Keplerian-based Hamiltonian splitting for gravitational N-body simulations. *Monthly Notices of the Royal Astronomical Society*, 440:719–730, May 2014. doi: 10.1093/mnras/stu282. (Citado nas páginas ix, 24, 28, 33, 41 e 58.)
- [39] J. Goodman, D. C. Heggie, and P. Hut. On the Exponential Instability of N-Body Systems. *Astrophysical Journal*, 415:715, October 1993. doi: 10.1086/173196. (Citado na página 52.)
- [40] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, December 1987. doi: 10.1016/0021-9991(87)90140-9. (Citado na página 4.)
- [41] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric numerical integration : structure-preserving algorithms for ordinary differential equations*. Springer series in computational mathematics. Springer, Berlin, Heidelberg, New York, 2006. ISBN 978-3-540-30663-4. URL <http://opac.inria.fr/record=b1119901>. (Citado na página 14.)
- [42] S. W. Hawking and W. Israel. *Three hundred years of gravitation*. 1987. (Citado na página 17.)
- [43] D. C. Heggie and R. D. Mathieu. Standardised Units and Time Scales. In P. Hut and S. L. W. McMillan, editors, *The Use of Supercomputers in Stellar Dynamics*, volume 267 of *Lecture Notes in Physics*, Berlin Springer Verlag, page 233, 1986. doi: 10.1007/BFb0116419. (Citado nas páginas 11, 29 e 45.)
- [44] C. Hellström and S. Mikkola. Explicit algorithmic regularization in the few-body problem for velocity-dependent perturbations. *Celestial Mechanics and Dynamical Astronomy*, 106:143–156, February 2010. doi: 10.1007/s10569-009-9248-8. (Citado nas páginas 38, 39 e 40.)
- [45] M. Hénon. Two Recent Developments Concerning the Monte Carlo Method. In A. Hayli, editor, *Dynamics of the Solar Systems*, volume 69 of *IAU Symposium*, pages 133–+, 1975. (Citado na página 12.)
- [46] D. M. Hernandez and E. Bertschinger. Symplectic integration for the collisional gravitational N-body problem. *Monthly Notices of the Royal Astronomical Society*, 452:1934–1944, September 2015. doi: 10.1093/mnras/stv1439. (Citado nas páginas 44 e 58.)

- [47] P. Hut. Dense stellar systems as laboratories for fundamental physics. *New Astronomy Reviews*, 54:163–172, March 2010. doi: 10.1016/j.newar.2010.09.009. (Citado nas páginas 3 e 23.)
- [48] P. Hut, J. Makino, and S. McMillan. Building a better leapfrog. *Astrophysical Journal Letters*, 443:L93–L96, April 1995. doi: 10.1086/187844. (Citado na página 35.)
- [49] T. Ito and K. Tanikawa. Long-term integrations and stability of planetary orbits in our Solar system. *Monthly Notices of the Royal Astronomical Society*, 336:483–500, October 2002. doi: 10.1046/j.1365-8711.2002.05765.x. (Citado na página 45.)
- [50] Y. Itoh. Equation of motion for relativistic compact binaries with the strong field point particle limit: Third post-Newtonian order. *Physical Review D*, 69(6):064018–+, March 2004a. doi: 10.1103/PhysRevD.69.064018. (Citado nas páginas 17, 18 e 63.)
- [51] Y. Itoh. On the equation of motion of compact binaries in the post-Newtonian approximation. *Classical and Quantum Gravity*, 21:529–+, March 2004b. doi: 10.1088/0264-9381/21/5/021. (Citado na página 17.)
- [52] Y. Itoh. Third-and-a-half order post-Newtonian equations of motion for relativistic compact binaries using the strong field point particle limit. *Physical Review D*, 80(12):124003–+, December 2009. doi: 10.1103/PhysRevD.80.124003. (Citado nas páginas 17, 18 e 63.)
- [53] Y. Itoh and T. Futamase. New derivation of a third post-Newtonian equation of motion for relativistic compact binaries without ambiguity. *Physical Review D*, 68(12):121501–+, December 2003. doi: 10.1103/PhysRevD.68.121501. (Citado na página 17.)
- [54] B. R. Iyer and C. M. Will. Post-Newtonian gravitational radiation reaction for two-body systems. *Physical Review Letters*, 70:113–116, January 1993. doi: 10.1103/PhysRevLett.70.113. (Citado na página 17.)
- [55] B. R. Iyer and C. M. Will. Post-Newtonian gravitational radiation reaction for two-body systems: Nonspinning bodies. *Physical Review D*, 52:6882–6893, December 1995. doi: 10.1103/PhysRevD.52.6882. (Citado na página 17.)
- [56] W. Jaffe. A simple model for the distribution of light in spherical galaxies. *Monthly Notices of the Royal Astronomical Society*, 202:995–999, March 1983. (Citado na página 12.)
- [57] J. Jänes, I. Pelupessy, and S. Portegies Zwart. A connected component-based method for efficiently integrating multi-scale N-body systems. *Astronomy and Astrophysics*, 570:A20, October 2014. doi: 10.1051/0004-6361/201423831. (Citado na página 58.)
- [58] P. Jaranowski and G. Schäfer. Radiative 3.5 post-Newtonian ADM Hamiltonian for many-body point-mass systems. *Physical Review D*, 55:4712–4722, April 1997. doi: 10.1103/PhysRevD.55.4712. (Citado na página 17.)

- [59] P. Jaranowski and G. Schäfer. Third post-Newtonian higher order ADM Hamilton dynamics for two-body point-mass systems. *Physical Review D*, 57:7274–7291, June 1998. doi: 10.1103/PhysRevD.57.7274. (Citado na página 17.)
- [60] P. Jaranowski and G. Schäfer. Binary black-hole problem at the third post-Newtonian approximation in the orbital motion: Static part. *Physical Review D*, 60(12):124003–+, December 1999. doi: 10.1103/PhysRevD.60.124003. (Citado na página 17.)
- [61] P. Jaranowski and G. Schäfer. Erratum: Third post-Newtonian higher order ADM Hamilton dynamics for two-body point-mass systems [Phys. Rev. D 57, 7274 (1998)]. *Physical Review D*, 63(2):029902–+, January 2001. doi: 10.1103/PhysRevD.63.029902. (Citado na página 17.)
- [62] W. Kahan and R.-C. Li. Composition constants for raising the orders of unconventional schemes for ordinary differential equations. *Mathematics of Computation*, 66:1089–1099, 1997. (Citado na página 39.)
- [63] C. Königsdörffer, G. Faye, and G. Schäfer. Binary black-hole dynamics at the third-and-a-half post-Newtonian order in the ADM formalism. *Physical Review D*, 68(4):044004–+, August 2003. doi: 10.1103/PhysRevD.68.044004. (Citado na página 17.)
- [64] P. Kroupa. On the variation of the initial mass function. *Monthly Notices of the Royal Astronomical Society*, 322:231–246, April 2001. doi: 10.1046/j.1365-8711.2001.04022.x. (Citado na página 26.)
- [65] P. Kroupa. The Initial Mass Function of Stars: Evidence for Uniformity in Variable Systems. *Science*, 295:82–91, January 2002. doi: 10.1126/science.1067524. (Citado na página 26.)
- [66] M. H. Lee, M. J. Duncan, and H. F. Levison. Variable Time Step Integrators for Long-Term Orbital Integrations. In D. A. Clarke & M. J. West, editor, *Computational Astrophysics; 12th Kingston Meeting on Theoretical Astrophysics*, volume 123 of *Astronomical Society of the Pacific Conference Series*, pages 32–+, 1997. (Citado na página 35.)
- [67] H. A. Lorentz and J. Droste. The motion of a system of bodies under the influence of their mutual attraction, according to Einstein’s theory. *Versl. K. Akad. Wetensch. Amsterdam*, 26:392, 1917. (Citado nas páginas 17 e 19.)
- [68] J. Makino. A Modified Aarseth Code for GRAPE and Vector Processors. *Publications of the Astronomical Society of Japan*, 43:859–876, December 1991a. (Citado na página 28.)
- [69] J. Makino. Optimal order and time-step criterion for Aarseth-type N-body integrators. *Astrophysical Journal*, 369:200–212, March 1991b. doi: 10.1086/169751. (Citado nas páginas 28 e 44.)
- [70] J. Makino and S. J. Aarseth. On a Hermite integrator with Ahmad-Cohen scheme for gravitational many-body problems. *Publications of the Astronomical Society of Japan*, 44:141–151, April 1992. (Citado na página 28.)

- [71] J. Makino, P. Hut, M. Kaplan, and H. Saygin. A time-symmetric block time-step algorithm for N-body simulations. *New Astronomy*, 12:124–133, November 2006. doi: 10.1016/j.newast.2006.06.003. (Citado nas páginas 28 e 35.)
- [72] S. L. W. McMillan. The Vectorization of Small-N Integrators. In P. Hut and S. L. W. McMillan, editors, *The Use of Supercomputers in Stellar Dynamics*, volume 267 of *Lecture Notes in Physics*, Berlin Springer Verlag, page 156, 1986. doi: 10.1007/BFb0116406. (Citado na página 28.)
- [73] S. Mikkola and S. Aarseth. A Time-Transformed Leapfrog Scheme. *Celestial Mechanics and Dynamical Astronomy*, 84:343–354, December 2002. (Citado nas páginas 28 e 35.)
- [74] S. Mikkola and K. Tanikawa. Explicit Symplectic Algorithms For Time-Transformed Hamiltonians. *Celestial Mechanics and Dynamical Astronomy*, 74:287–295, August 1999a. doi: 10.1023/A:1008368322547. (Citado nas páginas 28 e 35.)
- [75] S. Mikkola and K. Tanikawa. Algorithmic regularization of the few-body problem. *Monthly Notices of the Royal Astronomical Society*, 310:745–749, December 1999b. doi: 10.1046/j.1365-8711.1999.02982.x. (Citado nas páginas 28 e 35.)
- [76] X. X. Newhall, E. M. Standish, and J. G. Williams. DE 102 - A numerically integrated ephemeris of the moon and planets spanning forty-four centuries. *Astronomy and Astrophysics*, 125:150–167, August 1983. (Citado na página 17.)
- [77] S. Nisanke and L. Blanchet. Gravitational radiation reaction in the equations of motion of compact binaries to 3.5 post-Newtonian order. *Classical and Quantum Gravity*, 22:1007–1031, March 2005. doi: 10.1088/0264-9381/22/6/008. (Citado na página 17.)
- [78] I. P. Omelyan, I. M. Mryglod, and R. Folk. Symplectic analytically integrable decomposition algorithms: classification, derivation, and application to molecular dynamics, quantum and celestial mechanics simulations. *Computer Physics Communications*, 151:272–314, April 2003. doi: 10.1016/S0010-4655(02)00754-3. (Citado nas páginas 16, 37 e 39.)
- [79] P. Padoan, Å. Nordlund, A. G. Kritsuk, M. L. Norman, and P. S. Li. Two Regimes of Turbulent Fragmentation and the Stellar Initial Mass Function from Primordial to Present-Day Star Formation. *Astrophysical Journal*, 661:972–981, June 2007. doi: 10.1086/516623. (Citado na página 26.)
- [80] A. Parravano, C. F. McKee, and D. J. Hollenbach. An Initial Mass Function for Individual Stars in Galactic Disks. I. Constraining the Shape of the Initial Mass Function. *Astrophysical Journal*, 726:27, January 2011. doi: 10.1088/0004-637X/726/1/27. (Citado na página 26.)
- [81] M. E. Pati and C. M. Will. Post-Newtonian gravitational radiation and equations of motion via direct integration of the relaxed Einstein equations. II. Two-body equations of motion to second post-Newtonian order, and radiation reaction to 3.5 post-Newtonian order. *Physical Review D*, 65(10):104008–+, May 2002. doi: 10.1103/PhysRevD.65.104008. (Citado na página 17.)

- [82] F. I. Pelupessy, J. Jänes, and S. Portegies Zwart. N-body integrators with individual time steps from Hierarchical splitting. *New Astronomy*, 17:711–719, November 2012. doi: 10.1016/j.newast.2012.05.009. (Citado nas páginas 33, 34, 36, 37, 38 e 57.)
- [83] F. I. Pelupessy, A. van Elteren, N. de Vries, S. L. W. McMillan, N. Drost, and S. F. Portegies Zwart. The Astrophysical Multipurpose Software Environment. *Astronomy and Astrophysics*, 557:A84, September 2013. doi: 10.1051/0004-6361/201321252. (Citado na página 5.)
- [84] S. Portegies Zwart, S. McMillan, D. Groen, A. Gualandris, M. Sipior, and W. Vermin. A parallel gravitational N-body kernel. *New Astronomy*, 13:285–295, July 2008. doi: 10.1016/j.newast.2007.11.002. (Citado na página 53.)
- [85] S. F. Portegies Zwart, P. Hut, S. L. W. McMillan, and F. Verbunt. Star cluster ecology. II. Binary evolution with single-star encounters. *Astronomy and Astrophysics*, 328:143–157, December 1997a. (Citado na página 4.)
- [86] S. F. Portegies Zwart, P. Hut, and F. Verbunt. Star cluster ecology. I. A cluster core with encounters between single stars. *Astronomy and Astrophysics*, 328:130–142, December 1997b. (Citado na página 4.)
- [87] S. F. Portegies Zwart, J. Makino, S. L. W. McMillan, and P. Hut. Star cluster ecology. III. Runaway collisions in young compact star clusters. *Astronomy and Astrophysics*, 348:117–126, August 1999. (Citado na página 4.)
- [88] S. F. Portegies Zwart, S. L. W. McMillan, P. Hut, and J. Makino. Star cluster ecology - IV. Dissection of an open star cluster: photometry. *Monthly Notices of the Royal Astronomical Society*, 321:199–226, February 2001. (Citado na página 4.)
- [89] S. F. Portegies Zwart, P. Hut, S. L. W. McMillan, and J. Makino. Star cluster ecology - V. Dissection of an open star cluster: spectroscopy. *Monthly Notices of the Royal Astronomical Society*, 351:473–486, June 2004. doi: 10.1111/j.1365-2966.2004.07709.x. (Citado na página 4.)
- [90] S. F. Portegies Zwart, H. Baumgardt, S. L. W. McMillan, J. Makino, P. Hut, and T. Ebisuzaki. The Ecology of Star Clusters and Intermediate-Mass Black Holes in the Galactic Bulge. *Astrophysical Journal*, 641:319–326, April 2006. doi: 10.1086/500361. (Citado na página 4.)
- [91] S. F. Portegies Zwart, S. L. W. McMillan, and J. Makino. Star cluster ecology - VII. The evolution of young dense star clusters containing primordial binaries. *Monthly Notices of the Royal Astronomical Society*, 374:95–106, January 2007. doi: 10.1111/j.1365-2966.2006.11142.x. (Citado na página 4.)
- [92] M. Preto and S. Tremaine. A Class of Symplectic Integrators with Adaptive Time Step for Separable Hamiltonian Systems. *Astronomical Journal*, 118:2532–2541, November 1999. doi: 10.1086/301102. (Citado na página 35.)
- [93] E. E. Salpeter. The Luminosity Function and Stellar Evolution. *Astrophysical Journal*, 121:161, January 1955. doi: 10.1086/145971. (Citado na página 26.)

- [94] J. Scalo. The IMF Revisited: A Case for Variations. In G. Gilmore & D. Howell, editor, *The Stellar Initial Mass Function (38th Herstmonceux Conference)*, volume 142 of *Astronomical Society of the Pacific Conference Series*, page 201, 1998. (Citado na página 26.)
- [95] J. M. Scalo. The stellar initial mass function. *Fundamentals of Cosmic Physics*, 11: 1–278, May 1986. (Citado na página 26.)
- [96] R. D. Skeel and C. W. Gear. Does variable step size ruin a symplectic integrator? *Physica D Nonlinear Phenomena*, 60:311–313, November 1992. doi: 10.1016/0167-2789(92)90247-K. (Citado na página 35.)
- [97] V. Springel. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, 364:1105–1134, December 2005. doi: 10.1111/j.1365-2966.2005.09655.x. (Citado na página 4.)
- [98] D. Sugimoto, Y. Chikada, J. Makino, T. Ito, T. Ebisuzaki, and M. Umemura. A special-purpose computer for gravitational many-body problems. *Nature*, 345:33–35, May 1990. doi: 10.1038/345033a0. (Citado na página 4.)
- [99] V. Szebehely and C. F. Peters. Complete solution of a general problem of three bodies. *Astronomical Journal*, 72:876, September 1967. doi: 10.1086/110355. (Citado na página 45.)
- [100] A. Tanikawa, P. Hut, and J. Makino. Unexpected formation modes of the first hard binary in core collapse. *New Astronomy*, 17:272–280, April 2012. doi: 10.1016/j.newast.2011.09.001. (Citado na página 49.)
- [101] J. H. Taylor. Pulsar timing and relativistic gravity. *Classical and Quantum Gravity*, 10:167–+, December 1993. doi: 10.1088/0264-9381/10//017. (Citado na página 17.)
- [102] J. H. Taylor and J. M. Weisberg. A new test of general relativity - Gravitational radiation and the binary pulsar PSR 1913+16. *Astrophysical Journal*, 253:908–920, February 1982. doi: 10.1086/159690. (Citado na página 17.)
- [103] J. H. Taylor, L. A. Fowler, and P. M. McCulloch. Measurements of general relativistic effects in the binary pulsar PSR 1913+16. *Nature*, 277:437–440, February 1979. doi: 10.1038/277437a0. (Citado na página 17.)
- [104] T. S. van Albada. Numerical integrations of the N-body problem. *Bulletin of the Astronomical Institutes of the Netherlands*, 19:479, January 1968. (Citado na página 10.)
- [105] S. von Hoerner. Die numerische Integration des n-Körper-Problemes für Sternhaufen. I. *Z. Astrophys*, 50:184–214, 1960. (Citado na página 10.)
- [106] J. Wisdom and M. Holman. Symplectic maps for the n-body problem. *Astronomical Journal*, 102:1528–1538, October 1991. doi: 10.1086/115978. (Citado nas páginas 6 e 33.)
- [107] H. Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150:262–268, November 1990. doi: 10.1016/0375-9601(90)90092-3. (Citado nas páginas 15, 16, 37 e 39.)

- [108] H. Zhao. Analytical models for galactic nuclei. *Monthly Notices of the Royal Astronomical Society*, 278:488–496, January 1996. (Citado na página 26.)
- [109] H. Zhao. Analytical dynamical models for double power-law galactic nuclei. *Monthly Notices of the Royal Astronomical Society*, 287:525–537, May 1997. (Citado na página 26.)