Universidade do Vale do Paraíba – UNIVAP Instituto de Pesquisa & Desenvolvimento – IP&D Programa de Pós-Graduação em Física e Astronomia

ENG. LUIZ EDUARDO CAMARGO ARANHA SCHIAVO

SISTEMA DE IMERSÃO PARA SIMULAÇÃO DE N-CORPOS EM ASTROFÍSICA

São José dos Campos - SP

2018

ENG. LUIZ EDUARDO CAMARGO ARANHA SCHIAVO

SISTEMA DE IMERSÃO PARA SIMULAÇÃO DE N-CORPOS EM ASTROFÍSICA

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Física e Astronomia da Universidade do Vale do Paraíba, como complementação dos créditos necessários para obtenção do grau de Mestre em Física e Astronomia.

Orientador: Dr. Irapuan Rodrigues de Oliveira Filho Coorientador: Dr. Valdir Gil Pillat

São José dos Campos - SP2018





TERMO DE AUTORIZAÇÃO DE DIVULGAÇÃO DA OBRA

Ficha catalográfica

Camargo Aranha Schiavo, Luiz Eduardo SISTEMA DE IMERSÃO PARA SIMULAÇÃO DE N-CORPOS EM ASTROFÍSICA / Luiz Eduardo Camargo Aranha Schiavo; orientador, Irapuan Rodrigues de Oliveira Filho; co-orientador Valdir Gil Pillat. -São José dos Campos, SP, 2018. 1 CD-ROM, 103 p.

Dissertação (Mestrado Acadêmico) - Universidade do Vale do Paraíba, São José dos Campos. Programa de Pós-Graduação em Física e Astronomia.

Inclui referências

1. Física e Astronomia. 2. Dinâmica de Sistemas Estelares. 3. N-Corpos gravitacional.. 4. Visualização de Modelos. 5. Computação Gráfica e Realidade Virtual. I. Rodrigues de Oliveira Filho, Irapuan, orient. II. Gil Pillat, Valdir, co-orient. III. Universidade do Vale do Paraíba. Programa de Pós-Graduação em Física e Astronomia. IV. Título.

Eu, Luiz Eduardo Camargo Aranha Schiavo, autor(a) da obra acima referenciada:

Autorizo a divulgação total ou parcial da obra impressa, digital ou fixada em outro tipo de mídia, bem como, a sua reprodução total ou parcial, devendo o usuário da reprodução atribuir os créditos ao autor da obra, citando a fonte.

Declaro, para todos os fins e efeitos de direito, que o Trabalho foi elaborado respeitando os princípios da moral e da ética e não violou qualquer direito de propriedade intelectual sob pena de responder civil, criminal, ética e profissionalmente por meus atos.

São José dos Campos, 26 de Fevereiro de 2018.

Autor(a) da Obra

Data da defesa: _26 , 02 , 2018





LUIZ EDUARDO CAMARGO ARANHA SCHIAVO

"SISTEMA DE IMERSÃO PARA SIMULAÇÃO DE N-CORPOS EM ASTROFÍSICA."

Dissertação aprovada como requisito parcial à obtenção do grau de Mestre, do Programa de Pós-Graduação em Física e Astronomia, do Instituto de Pesquisa e Desenvolvimento da Universidade do Vale do Paraíba - Univap, pela seguinte banca examinadora:

PROF. DR. ALEXANDRE SOARES DE OLIVEIRA	percuper
PROF. DR. IRAPUAN RODRIGUES DE OLIVEIRA FILHO	full ks
PROF. DR. VALDIR GIL PILLAT	
PROF. DR. REINALDO ROBERTO ROSA - INPE	· 072

Prof. Dr. Leandro José Raniero

Diretor do IP&D – Univap

São José dos Campos, 26 de Fevereiro de 2018.

Dedico este trabalho à minha família que me apoiou durante essa jornada em busca de conhecimento, em especial à minha esposa Laura e a meus pais, Antonio e Hosana.

AGRADECIMENTOS

Chegando ao fim desta etapa na minha formação como professor, preciso agradecer pelo apoio dos familiares, amigos, professores e colegas que, de alguma forma colaboraram ao longo dessa jornada.

Em primeiro lugar à minha família, em especial à minha esposa Laura, sempre pronta para ajudar e tolerar minha ausência no árduo período da escrita e revisão de texto. Aos meus pais, Antonio e Hosana obrigado por tudo, meus irmãos Kadu, Gu e Ju pelo apoio e carinho.

Ao amigo e mestre Carlos Horst Wagner (in memoriam) pelas nossas tardes de trabalho e conversas que sempre serão lembradas com carinho.

Obrigado aos amigos e orientadores Irapuan Rodrigues de Oliveira Filho e Valdir Gil Pillat, que me deram força para continuar e me ajudaram nos momentos de dificuldade.

Pelo conhecimento transmitido e pela amizade dos professores Paulo Fagundes, Márcio Muella, Caius Lucius, Abalde Guede, Venkatesh Kavutarapu, Virginia Klausner, Sérgio Pilling, Arian Ojeda, Alan (tchê) Prestes, Alessandra Pacini, Alexandre Soares, Angela Krabbe, Oli Dors, Diana (Prince) Paula, Guga (Francisco Rocha), Marlos Rockenbach, Alexandre Tardelli, Priscila Fávero, Laurita dos Santos, Sandra Costa, Érico Rempel e Ivânio Puerari, muito obrigado por todo conhecimento compartilhado!

Ralamos muito juntos, então nada mais justo que agradecer aos amigos, Maukers Alem, Antonio Nilson, Alexandre Bergantini, Matheus Palhares, Murilo Martins, Brunno Augusto, Dino (sauro) Beghetto, Flávia Coelho, Flávio Roony, Victor Bonfim, Will Robson e todos os demais colegas do PPGFA.

Durante este período, alguns amigos se tornaram família: Lucas Caritá, Maira Silva, Hudson Zanin, Caroline Simões, Abalde Guede, Norma Jean e Diogo Correa MUITO obrigado por fazerem parte de minha vida, vocês foram fundamentais!

For millions of years, mankind lived just like the animals Then something happenend which unleashed the power of our imagination: We learned to talk

(Stephen Hawking em Pink Floyd - Keep Talking)

RESUMO

Esta dissertação acompanha um estudo de Dinâmica de Sistemas Estelares, especificamente na colisão de galáxias. Para tal é feita uma revisão de conceitos Matemáticos e Físicos que são utilizados na solução do problema de N-Corpos gravitacional. São abordados conceitos utilizados na criação de modelos numéricos de galáxias, algoritmos de cálculo de forças utilizados na sua interação e integradores de tempo para a evolução temporal. Uma revisão de Computação Gráfica é feita para conceituar o funcionamento da classe de softwares utilizados na visualização dos resultados destas simulações. Um grid com 72 simulações de colisão de galáxias foi rodado utilizando o código GADGET-2 e esse grid foi utilizado no GLnemo2, Gadget Viewer e no GraVity, que é uma nova ferramenta de visualização de modelos que teve seu desenvolvimento durante este mestrado. Para o caso específico da visualização utilizando o GraVity, a visualização pode ser feita em modo Desktop ou imerso em Realidade Virtual através do uso de um HMD.

Palavras-chave: Dinâmica de Sistemas Estelares, N-Corpos gravitacional, Visualização de Modelos, Computação Gráfica, Realidade Virtual.

IMMERSION SYSTEM FOR N-BODY SIMULATIONS IN ASTROPHYSICS

ABSTRACT

This Master's dissertation follow a study of Dynamics of Stellar Systems, specifically in the collision of galaxies. For this, a review of Mathematical and Physical concepts that are used in the solution of the problem of gravitational N-Body is made. The concepts used in the creation of numerical models of galaxies, algorithms for calculating the forces used in their interaction, and time integrators for temporal evolution are discussed. A review of Computer Graphics is done to conceptualize the operation of the software class used in visualizing the results of these simulations. A grid with 72 galaxy collision simulations was run using the GADGET-2 code and this grid was used as the object of study in this work. The results of the simulations were visualized in GLnemo2, Gadget Viewer and in GraVity. For the specific case of visualization using GraVity, the visualization can be done in Desktop mode or immersed in Virtual Reality through the use of an HMD. Both the GraVity code and its HMD were developed during this master's degree.

Keywords: Gravitational N-Body, Dynamics of Stellar Systems, Model Visualization, Computer Graphics, Virtual Reality.

LISTA DE ILUSTRAÇÕES

Figura 1 $-$	Digrama Tunning Fork proposto por Edwin Hubble (HUBBLE, 1926).	24
Figura 2 –	Galáxias Elípticas, da esquerda para direita: a) NGC 4261 (E2-3); b)	
	NGC 3377 (E5-6)	25
Figura 3 –	2MASX J00010463-5144109: galáxia discoidal S0	26
Figura 4 –	Exemplo de Galáxias Espirais, da esquerda para direita: a) NGC 2599	
	(Sa); b) Andrômeda M31 (Sb); c) M74 (Sc)	26
Figura 5 –	Galáxias Espirais Barradas, da esquerda para direita: a) NGC 1291	
	(SBa); b) NGC 1300 (SBb); c) NGC 7741 (SBc).	27
Figura 6 –	Galáxias Irregulares, da esquerda para direita: a) Grande Nuvem de	
	Magalhães; b) Pequena Nuvem de Magalhães	28
Figura 7 –	Galáxia com Anel Polar NGC 660.	29
Figura 8 –	A aparência de uma PRG e suas possíveis orientações, com inclinação	
	α em relação ao plano xy e β e com respeito ao eixo y	30
Figura 9 –	Diferença da excentricidade de órbitas, do Livro Astronomia Nova	32
Figura 10 –	Detalhando a relação dos raios com o semieixo maior de uma elipse	32
Figura 11 –	Lei das áreas de Kepler (esboço sem escala).	33
Figura 12 –	Dois corpos com massas iguais e vetores velocidade opostos, orbitando	
	seu centro de massa C_m	37
Figura 13 –	Modelo hierárquico baseado na árvore <i>octree</i> de Joshua Barnes e Piet	
	Hut	45
Figura 14 –	Modelo hierárquico baseado na bissecção recursiva de John Dubinski,	
	alocando recursos diretamente para núcleos de processamento	46
Figura 15 –	Esquema de integração usando Leapfrog	48
Figura 16 –	Representação de um pixel no sistema de coordenadas no monitor	51
Figura 17 –	Sistema de cores RGB	52
Figura 18 –	Sistema de cores CMYK	52
Figura 19 –	A ordem da transformação nos sistemas de coordenadas em CG	53
Figura 20 –	Exemplo do uso do algoritmo de Bresenham para traçar uma linha	
	entre o ponto $(0,0)$ ao ponto $(7,5)$	55
Figura 21 –	Ilustração da visualização da cena, usando perspectiva e câmera	58
Figura 22 –	Ilustração do uso de HMD.	59
Figura 23 –	Exemplo de utilização de HMD comercial.	60
Figura 24 –	Exemplo de visualização de galáxias em processo de colisão em modo	
	Desktop	64
Figura 25 –	Exemplo de visualização com tela dividida de galáxias em processo de	
	colisão em modo RV	64

Figura 2	26 -	Impressão do corpo do protótipo baseado no Zendome	65
Figura 2	27 -	Os modelos 3D e impressos do HMD: a) o separador do corpo de lentes;	
		b) o suporte da lente; c) o conjunto de suporte das lentes; d) o suporte	
		do LCD; e) os correspondentes impressos	66
Figura 2	28 -	Protótipo desmontado, sem o conjunto de lentes.	67
Figura 2	29 –	Tela de LCD de 7 polegadas, exibindo um frame de simulação com um	
		par de galáxias em interação.	67
Figura 3	30 -	Primeiros testes em bancada com o protótipo	68
Figura 3	31 -	Imagem sendo exibida nos testes iniciais em bancada com o protótipo.	68
Figura 3	32 -	Plotagem do disco de estrelas e gás do modelo A. a) Gadget Viewer	
		1.0.7; b) GLnemo2 1.10.0; c) GraVity 0.2.12	70
Figura 3	33 -	Plotagem com estrelas e gás do arquivo de condições iniciais do modelo	
		sim-Ax90Cy0-vel0-q8. a) Gadget Viewer 1.0.7; b) GLnemo 2 $1.10.0;$ c)	
		GraVity 0.2.12	71
Figura 3	34 -	Diagrama com todas as combinações de simulações possíveis	71
Figura 3	35 -	Comparativo do frame das componentes disco de estrelas e gás do Modelo	
		A após 0,21 bilhões de anos: a) Gadget Viewer 1.0.7; b) GLnemo2 1.10.0;	
		c) GraVity 0.2.12.	74
Figura 3	36 -	Comparativo do frame das componentes disco de estrelas e gás do Modelo	
		C após 0,21 bilhões de anos: a) Gadget Viewer 1.0.7; b) GL nemo 2 $1.10.0;$	
		c) GraVity 0.2.12. \ldots \ldots \ldots \ldots \ldots \ldots \ldots	75
Figura 3	37 –	Comparativo do frame das componentes disco de estrelas e gás do Modelo	
		D após 0,55 bilhões de anos: a) Gadget Viewer 1.0.7; b) GL nemo 2 $1.10.0;$	
		c) GraVity 0.2.12. \ldots \ldots \ldots \ldots \ldots \ldots \ldots	75
Figura 3	38 -	Comparativo do frame das componentes disco de estrelas e gás do Modelo	
		E após 0,55 bilhões de anos: a) Gadget Viewer 1.0.7; b) GL nemo 2 $1.10.0;$	
		c) GraVity 0.2.12. \ldots \ldots \ldots \ldots \ldots \ldots \ldots	75
Figura 3	39 -	Plotagem das componentes de disco de estrelas da simulação sim-	
		Ax90Cy0-vel 0-q12 em 1,125 bilhões de anos de evolução: a) vista lateral	
		dos modelos após colisão; a) vista superior após colisão com rotação de	
		90 graus em direção ao eixo x; c) vista lateral após colisão com rotação	
		de 90 graus em direção ao eixo y. \ldots	76
Figura 4	40 -	Plotagem das componentes de disco de estrelas e gás na evolução da	
		secundária para SBc após a colisão: a) sim-Ax90Cy0-vel 0-q8 em 1,07 $$	
		bilhões de anos de evolução; b) sim-Ax90Cy30-vel 10-q8 em 0,97 bilhões	
		de anos de evolução; d) sim-Ax90Cy60-vel 20-q8 em 0,97 bilhões de anos	
		de evolução	77

Figura 41 $-$ Plotagem das componentes de disco de estrelas e gás durante o processo
de colisão: a) sim-Ax90Cy0-vel0-q12 em 0,99bilhões de anos de evolução;
b) sim-Ax90Cy30-vel 10-q12 em 0,93 bilhões de anos de evolução; d)
sim-Ax90Cy60-vel20-q12 em 0,89 bilhões de anos de evolução 77
Figura 42 – Plotagem das componentes de disco de estrelas e gás na evolução da
secundária para SBc após a colisão: a) sim-Ax90Dy0-vel0-q8 em 1,01
bilhões de anos de evolução; b) sim-Ax90Dy30-vel 10-q8 em 0,96 bilhões
de anos de evolução; d) sim-Ax90Dy60-vel 20-q8 em 0,96 bilhões de anos
de evolução
Figura 43 $-$ Plotagem das componentes de disco de estrelas e gás durante o processo
de colisão: a) sim-Ax90Dy0-vel0-q8 em 1,09 bilhões de anos de evolução;
b) sim-Ax90Dy30-vel 10-q8 em 1,04 bilhões de anos de evolução; d)
sim-Ax90Dy60-vel20-q8 em 1,04 bilhões de anos de evolução 79
Figura 44 – Histograma de tempo de CPU
Figura 45 – Detalhando o tempo de CPU
Figura 46 – Raspberry Pi 2 modelo B, utilizada nos Projetos 1 e 2 (em tamanho real). 92
Figura 47 – Motion Plus com Nunchuk Wii, utilizada nos Projetos 1 e 2 95
Figura 48 – Oculus Rift Development Kit, considerado para o Projeto 1 95
Figura 49 – Teste de geração de imagem estereoscópica projetada, utilizado no
Projeto 1
Figura 50 – Teste de geração de imagem este reoscópica projetada, utilizado no
Projeto 1
Figura 51 – OSVR Development Kit, considerado para o Projeto 2 97
Figura 52 – Suporte das lentes do protótipo baseado no Cardboard 99
Figura 53 $-$ Modelo do corpo do suporte das lentes do protótipo baseado no Cardboard.100 $$
Figura 54 – Corpo do protótipo baseado no Cardboard
Figura 55 – Montagem e testes do protótipo baseado no Cardboard. \ldots \ldots \ldots 102

LISTA DE TABELAS

Tabela 1 –	Comparação entre as ideias de Copérnico e Kepler para o movimento	
	dos planetas.	31
Tabela 2 $\ -$	Aplicação da 3ª Lei de Kepler nos valores do sistema solar	35
Tabela 3 $\ -$	Protótipos de RV e seu estado de desenvolvimento	61
Tabela 4 –	Período de colisão dos modelos AC em bilhões de anos	77
Tabela 5 $$ –	Período de colisão dos modelos AD em bilhões de anos	79
Tabela 6 –	Tempo de processamento (<i>Wall time</i>) em [*HH:MM:SS,SSS] das simu-	
	lações com distância de pericentro de 8kpc	80
Tabela 7 –	Tempo de processamento (<i>Wall time</i>) em [*HH:MM:SS,SSS] das simu-	
	lações com distância de pericentro de 12kpc	81
Tabela 8 –	Tempo de CPU das simulações com distância de pericentro de 8kpc.	81
Tabela 9 –	Tempo de CPU das simulações com distância de pericentro de 12kpc	82

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface - Advanced RISC Machine ARM CAD - Computer Aided Design CG - Computação Gráfica CeCILL2 - Cea Cnrs Inria Logiciel Libre V2 CMYK - Cian, Magenta, Yellow and Black CUDA - Compute Unified Device Architecture Distro - Distribuição Linux $\mathbf{E}\mathbf{x}$ - Elliptical Galaxy¹ - GNU's Not Unix GNU GPU - Graphics Processing Unit GraVity - Immersive environment of Gas and Stars dynamics using Virtual Reality $\operatorname{GraVityVR}$ - Immersive environment of Gas and Stars dynamics using Virtual Reality in Virtual Reality GPL - General Public License HMD - Head Mounted Display OpenCL - Open Computing Language OpenGL - Open Graphics Library LCD - Liquid Crystal Display MPSS - Manycore Platform Software Stack Pixel - Picture Element RISC - Reduced Instruction Set Computer RV - Realidade Virtual

 1 $\,$ Exemplos de Galáxias Elípticas na Figura 2.

RGB	- Red, Green and Blue
Sx	- Spiral Galaxy ²
SBx	- Spiral Barred Galaxy ³
SDL2	- Simple Direct Media Layer 2
SDK	- Software Development Kit
SO	- Sistema Operacional
SPH	- Smoothed Particle Hidrodynamics
SRU	- Sistema de Referência do Universo
SRO	- Sistema de Referência do Objeto
SRN	- Sistema de Referência Normalizado
SRD	- Sistema de Referência do Dispositivo
TCP	- Transmission Control Protocol
UDP	- User Datagram Protocol
UNSIO	- Universal N-body Snapshot Input Output
VR	- Virtual Reality

 ² Exemplos de Galáxias Espirais na Figura 4.
 ³ Exemplos de Galáxias Espirais Barradas na Figura 5.

LISTA DE SÍMBOLOS

- O(N) Complexidade linear de um algoritmo.
- ${\cal O}(N^2)$ Complexidade quadrática de um algoritmo.
- O(NlogN) Complexidade log-linear de um algoritmo.

SUMÁRIO

1	INTRODUÇÃO	22
1.1	Organização deste trabalho	23
2	CONCEITOS DE FÍSICA	24
2.1	Galáxias	24
2.1.1	Galáxias Elípticas (E)	25
2.1.2	Galáxias (S0)	25
2.1.3	Galáxias Espirais (S)	26
2.1.4	Galáxias Espirais Barradas (SB)	27
2.1.5	Irregulares (I)	27
2.1.6	Galáxias com Anel Polar	28
2.2	Física clássica	30
2.2.1	Kepler	30
2.2.1.1	1ª Lei de Kepler	31
2.2.1.2	2ª Lei de Kepler	33
2.2.1.3	3ª Lei de Kepler	34
2.2.2	Newton	35
2.2.2.1	Gravitação Universal	35
2.2.2.2	Teoremas de Newton	36
2.2.3	Interação gravitacional	36
2.2.3.1	Interação gravitacional de dois corpos	36
2.2.3.2	Interação gravitacional de três corpos	37
2.2.3.3	Interação gravitacional de N-Corpos	38
2.3	Dinâmica de Sistemas Estelares	40
2.3.1	Tratamento Físico de Simulações	40
2.3.1.1	Modelos numéricos de galáxias	41
2.3.1.1.1	Halo de matéria escura	41
2.3.1.1.2	Βοjο	42
2.3.1.1.3	Disco de Estrelas e Gás	42
2.3.1.2	Dinâmica gravitacional sem colisão	43
2.3.2	Algoritmos de Cálculo de Força	43
2.3.2.1	Soma Direta	44
2.3.2.2	Tree Code	44
2.3.2.3	SPH e o TreeSPH	46
2.3.3	O Tratamento do tempo nas simulações	47
2.3.3.1	Leapfrog	48

2.3.3.2	Passo de tempo adaptativo	49
3	CONCEITOS DE COMPUTAÇÃO GRÁFICA	50
3.1	Princípios de Computação Gráfica	50
3.2	Álgebra e geometria utilizadas em CG	52
3.2.1	Sistemas de coordenadas	52
3.2.2	Primitivas gráficas	54
3.2.3	Transformações	55
3.3	OpenGL	57
3.3.1	Projeção e posição de câmera	57
3.4	Imersão em Realidade Virtual	58
4	RESULTADOS	61
4.1	O ambiente imersivo	61
4.1.1	O código GraVity	62
4.1.1.1	O Desenvolvimento	62
4.1.1.2	Interface com usuário	63
4.1.2	Construção do HMD OpenSource	65
4.2	O Grid de Simulações	69
4.2.1	Modelos	69
4.3	Análise dos resultados	74
4.3.1	Resultados comuns a todas simulações	74
4.3.2	Simulações AC	76
4.3.3	Simulações AD e AE	78
4.3.4	Análise de performance das simulações	80
5	CONSIDERAÇÕES FINAIS	85
	REFERÊNCIAS	87
	APÊNDICES	90
	APÊNDICE A – PROJETO 1 - VR BASEADO NO OCULUS RIFT	91
A.1	Módulo de Entrada	91
A.2	Módulo de Processamento	93
A.3	Módulo de Renderização	94
A.4	Resultados preliminares	94
	APÊNDICE B – PROJETO 2 - VR BASEADO NO OSVR	97
B.1	Módulo de Renderização	97
B.2	Resultados preliminares	98

	APÊNDICE C – PROJETO 3 - VR BASEADO NO CARDBOARD 99
C .1	Construção do HMD
C.2	Módulo Unificado
C.3	Resultados preliminares
	APÊNDICE D – ARTIGOS PUBLICADOS

PREFÁCIO

UMA JANELA PARA O PASSADO

Se pudermos olhar para o céu noturno longe da poluição luminosa das cidades, em uma noite escura e sem nuvens, podemos nos deparar com um céu cravejado de pontos brilhantes como pedras preciosas brilhando em um fundo escuro e, em seguida, ao nos atentarmos em algum objeto brilhante, este objeto pode ser o que chamamos de estrela.

Essa luz fraca e distante é feita de partículas de luz conhecidas como fótons, que têm em seu caminho trechos de nossa galáxia que contêm gás e poeira. Na sua trajetória, a luz emitida vai perdendo energia e vai se avermelhando até que atinja nossos olhos que, por fim, convertem este estímulo em impulsos elétricos que são interpretados por nossos cérebros na forma de imagens.

Uma das primeiras coisas que aprendemos na astronomia é que estudar o céu é o mesmo que estudar o passado. Semelhante aos arqueólogos, os astrônomos munidos de seus telescópios e radiotelescópios observam estrelas a vários anos-luz de distância. Por exemplo, ao estudarmos um corpo que se encontra a uma distância de 100 anos-luz, estamos recebendo informações luminosas na forma de fótons que foram emitidos há 100 anos.

Em busca de se entender melhor, o homem olha para o seu passado, buscando trilhar um caminho mais fácil em seu futuro. Na astronomia tentamos entender a evolução do universo estudando a luz que levou centenas ou milhares de anos viajando até que se torne visível para o homem e isso gera curiosidade.

A CURIOSIDADE E O MÉTODO CIENTÍFICO

Na busca por conhecimento, o homem observa, descobre padrões, classifica e tenta entender o funcionamento de objetos ao seu redor e isto leva ao método científico. Ao longo da história da humanidade e utilizando sua inteligência, o homem estudou uma diversidade de objetos, estando estes ao seu alcance ou inatingíveis e apenas observados à distância no longínquo céu.

Diferentes povos como os Astecas, Incas, Maias, Babilônios, Gregos, Chineses, entre outros, munidos de sua curiosidade e capacidade de abstração, foram capazes de encontrar padrões nos movimentos celestes e destes padrões construíram calendários complexos e detalhados que ajudavam a planejar suas colheitas e organizar seu dia a dia.

Hoje, para poder classificar qualquer coleção, é necessário abstrair padrões, nomear ou enumerar objetos em uma ordem minimamente criteriosa, o que não era muito diferente com estes povos antigos. Destas observações e posteriores classificações foram descobertos e nomeados planetas e estrelas. Do conjunto de estrelas agrupadas em uma região do céu foram concebidas as constelações.

A curiosidade realmente é uma das características mais marcantes no ser humano. Deste impulso de observar e compreender, descobrimos as bases necessárias para criarmos uma sociedade moderna, dotada de meios cada vez mais dinâmicos para nos comunicar, seja para troca de informações, conhecimentos ou para trivialidades do cotidiano.

Na Grécia antiga, um dos berços da sociedade científica, pensadores como Aristarco e Ptolomeu, munidos de seus interesses, observavam o movimento dos astros e tentavam compreender o funcionamento das diferentes órbitas de estrelas e planetas, tentando montar modelos que explicassem o movimento destes objetos ou corpos.

Apesar destes primeiros modelos terem sido obtidos sem a grande capacidade de observação que temos com o uso dos telescópios de hoje, eles eram razoáveis e diferiam basicamente em relação ao seu centro no então conhecido universo observável.

Novamente a partir da curiosidade, no século XVI Tycho Brahe em seu observatório catalogou uma enormidade de estrelas e órbitas, e posteriormente, Johannes Kepler munido dessas informações, conseguiu determinar as primeiras leis capazes de descrever movimentos destes corpos no céu. Começava então a astronomia moderna.

Posteriormente Sir Isaac Newton, estudando o funcionamento da gravidade e em concordância aos estudos de Kepler, propôs a Lei da Gravitação Universal. Ao se deparar com algo que não conseguia explicar, Albert Einstein deu continuidade à gravitação e nos propiciou a Relatividade, que nos permitiu dominar a física necessária para comunicação em alta velocidade dos satélites. Toda vez que atingimos algo até então além da compreensão, voltamos ao método científico e de forma incremental e cíclica, buscamos entender um pouco mais do objeto de estudo.

Talvez quando tivermos atingido um maior nível de conhecimento, nos voltemos para outros problemas que ainda nem sabemos que existem. Até lá, o homem vai continuar tentando descobrir novos caminhos e criar novas ferramentas para o entendimento do nosso universo conhecido.

A TECNOLOGIA NO MUNDO DE HOJE

Hoje, envoltos na constante evolução tecnológica, vivemos em um mundo totalmente dependente do transistor, em uma era digital, onde, com um simples aplicativo podemos transformar nosso celular em uma carta celeste, com detalhadas informações sobre os objetos que podemos observar no céu visível. Podemos buscar objetos de céu profundo, algo inimaginável para a vanguarda da Astronomia na Grécia antiga.

E notório que a chegada das novas resoluções HD, FullHD, 4k, possibilitadas pelo aumento na densidade dos pontos em uma tela, nos proporcionou aumentar de forma significativa o nível de detalhes que tanto ansiávamos ver em um sistema que gera imagens de forma digital. Por tal, imagino que nos primeiros trabalhos de imersão em Realidade Virtual não se sonhava em chegar em tamanha riqueza de detalhes que temos hoje.

É curioso o fato de que uma tecnologia como a Realidade Virtual, teorizada em meados da década de 50, tenha amadurecido de forma tão irregular, a ponto de ficar boa parte do início do século XXI pouco utilizada ou mesmo esquecida. Atualmente vemos um crescimento em seu uso e desenvolvimento, se tentássemos relacionar o seu crescimento com a Lei de Moore⁴ (MOORE, 1965) provavelmente falharíamos catastroficamente.

 $^{^4~}$ A Lei de Moore prevê a cada dois anos, um aumento de 100% na quantidade de transistores dos chips mantendo seu custo.

1 INTRODUÇÃO

A Dinâmica de Sistemas Estelares permite o estudo de sistemas interagindo gravitacionalmente. Diferentes estruturas podem ser estudadas na Dinâmica de Sistemas Estelares que vão desde sistemas planetários, aglomerados de estrelas, galáxias isoladas ou em interação. Neste trabalho vamos nos concentrar no estudo de Galáxias, que são os blocos básicos do universo, durante o processo de interação gravitacional.

Sabemos que galáxias em aglomerados estão relativamente próximas umas das outras, isto é, as separações entre elas não são grandes comparadas com seus tamanhos. Isso significa que, provavelmente, essas galáxias estão em frequentes interações umas com as outras (KEPLER; SARAIVA, 2013).

O tempo é um fator limitante no estudo do comportamento de pares de galáxias em colisão, pois a ordem de grandeza do tempo de vida da civilização humana é insignificante perante ao tempo de vida de uma galáxia. Por isso, como não temos como acompanhar estes sistemas interagindo por grandes períodos, realizamos simulações computacionais.

Para estudarmos a dinâmica de galáxias através de simulações computacionais, a descrição mínima de posicionamento e estado do movimento de uma galáxia, requer o conhecimento de um conjunto de 6 variáveis: 3 de posição (x, y, z) e 3 de velocidade (V_x, V_y, V_z) . Partimos de dados observacionais onde é possível abstrair uma função de duas variáveis referentes à posição e uma referente à velocidade, $f(x, y, V_z)$. Com base nestas três informações, estimamos e aproximamos as informações restantes de posição e de velocidade com o objetivo de criar modelos numéricos que nos permitam simular estes dados observacionais com maior riqueza de parâmetros.

Estas simulações demandam uma grande quantidade de tempo computacional para serem realizadas, posteriormente a visualização de modelos resultantes é feita utilizando uma classe de softwares gráficos como o Glnemo2 (LAMBERT, 2012). Por isso, neste trabalho utilizamos a Física Computacional como ferramenta para a visualização destes modelos.

Para tal estudo, iniciamos o desenvolvimento de um novo código chamado **GraVity** (*Immersive environment of Gas and Stars dynamics using Virtual Reality*) com o objetivo de criar um ambiente que será utilizado tanto em desktop como em realidade virtual, que permita ao usuário ter liberdade de visualizar e realçar diferentes componentes dos snapshots de uma simulação de N-Corpos, como por exemplo, matéria escura, halo, gás, estrelas e buraco negro.

1.1 ORGANIZAÇÃO DESTE TRABALHO

- No Capítulo 2 é apresentada uma breve revisão da classificação de galáxias em relação a sua estrutura morfológica. São tratadas a Matemática e a Física básica que subjazem a Dinâmica de Sistemas Estelares. Também são abordados conceitos sobre modelos numéricos, algoritmos de cálculo de forças e integradores de tempo que são necessários para realizarmos estas simulações;
- No Capítulo 3 abordamos os fundamentos da Álgebra e da Geometria necessários para a Computação Gráfica. Também são abordados conceitos de imersão em Realidade Virtual.
- O estudo de um grid de simulações numéricas de colisão de galáxias, bem como o desenvolvimento do código GraVity são abordados no Capítulo 4;
- Considerações finais são tratadas no Capítulo 5;
- Nos Apêndices A, B e C temos a diferentes versões de protótipos Realidade Virtual utilizados ao longo deste trabalho;
- Os artigos publicados ao longo deste trabalho estão no Apêndice D.

2 CONCEITOS DE FÍSICA

Neste Capítulo é feita uma revisão de conceitos de Física que norteiam a elaboração dos modelos e algoritmos utilizados na simulação de N-Corpos gravitacional. Na Seção 2.1 temos uma breve revisão sobre morfologia de galáxias. Na Seção 2.2 temos as abordagens matemáticas e físicas de Kepler e Newton. Na Seção 2.3 são abordados os conceitos físicos e matemáticos dos modelos numéricos de galáxias e os algoritmos utilizados para o cálculo de forças e a integração no tempo.

2.1 GALÁXIAS

Galáxias são sistemas auto-gravitantes formados principalmente por estrelas, gás, poeira e matéria escura. Apesar de terem as mesmas constituintes, elas diferem entre si em relação à sua morfologia e sua população. Inicialmente foram categorizadas em quatro tipos básicos: Elípticas, Espirais, Espirais Barradas e Irregulares (HUBBLE, 1926). Esta classificação foi proposta por Edwin Hubble em 1926 e pode ser vista na Figura 1.

Figura 1 – Digrama Tunning Fork proposto por Edwin Hubble (HUBBLE, 1926).



Fonte: The Realm of the Nebulae. (HUBBLE, 1936).

2.1.1 GALÁXIAS ELÍPTICAS (E)

As galáxias elípticas possuem forma elipsoidal, não desenvolvem estrutura espiral em grande escala e são estruturalmente semelhantes aos núcleos de galáxias espirais. Possuem pouco gás, pouca poeira e estrelas velhas. São classificadas em função de sua excentricidade projetada, conforme a equação (1), onde a e b são os semieixos maior e menor respectivamente. Exemplos de Elípticas são NGC 4261 e NGC 3377 na Figura 2.

$$E_n = \frac{10(a-b)}{b} \tag{1}$$

Figura 2 – Galáxias Elípticas, da esquerda para direita: a) NGC 4261 (E2-3); b) NGC 3377 (E5-6).



 $\label{eq:source} Fonte: NGC 4261 < http://ned.ipac.caltech.edu/img/1994CAG1..B...0000S/NGC4261:I: 103aO:s1994.jpg>, NGC 3377 < http://ned.ipac.caltech.edu/img/1994CAG1..B...0000S/NGC3377:I:103aO+WG2:s1994.jpg>. \\$

2.1.2 GALÁXIAS (S0)

Galáxias do tipo S0, também chamadas de lenticulares, são galáxias que possuem forma discoidal, constituídas por bojo¹, disco² e halo³, que não possuem traços de estrutura espiral. Na Figura 3 temos o objeto 2MASX J00010463-5144109 como exemplo de S0.

 $^{^1~}$ O bojo é um grupo esferoide de estrelas encontrado na região central da maioria das galáxias espirais (LéPINE, 2008).

 $^{^2}$ O disco é composto principalmente por estrelas, poeira e gás.

³ Segundo Slater (2015), o halo galáctico possui forma esferoide que se estende além do componente principal e visível das galáxias, composto por estrelas, gás e matéria escura.



Figura 3 – 2MASX J00010463-5144109: galáxia discoidal S0.

Fonte: <https://ned.ipac.caltech.edu/dss/2MASX_J00010463-5144109.fits.gz>.

2.1.3 GALÁXIAS ESPIRAIS (S)

As galáxias espirais possuem bojo, disco, halo e braços espirais. Elas são classificadas com relação aos tamanhos de bojo e desenvolvimento de braços espirais. Em geral possuem estrelas jovens e velhas e são subdividas nas categorias:

- Sa Possuem bojo grande, braços espirais pequenos e bem enrolados. Um exemplo é a NGC 2599 na Figura 4 a);
- Sb Possuem bojo mediano e braços espirais intermediários. A galáxia Andrômeda M31 é um exemplo na Figura 4 b);
- Sc Possuem bojo pequeno, braços espirais grandes e mais abertos. Outro exemplo é a M74 na Figura 4 c).
- Figura 4 Exemplo de Galáxias Espirais, da esquerda para direita: a) NGC 2599 (Sa); b) Andrômeda M31 (Sb); c) M74 (Sc).



Fonte: NGC2599 (BAILLARD et al., 2011) <http://ned.ipac.caltech.edu/img/2011A+A. ..532A..74B/gri/PGC_023941:I:gri:bbl2011.jpg>, M31 T. Rector and B. Wolpa (NOA-O/AURA/NSF) <http://hubblesite.org/image/2962/news_release/2012-04>, M74 <http://ned.ipac.caltech.edu/img/2004HSTHH.C.....:/MESSIER_074:I:GRB:HST2004.jpg>.

2.1.4 GALÁXIAS ESPIRAIS BARRADAS (SB)

As galáxias espirais barradas diferem das espirais apenas por apresentar uma estrutura em forma de barra que atravessa seu bojo. Possuem estrelas jovens e velhas e são subdividas nas categorias:

- SBa Possuem bojo e barra grandes, braços espirais pequenos e bem enrolados. Um exemplo é a NGC 1291 na Figura 5 a);
- SBb Possuem bojo e barra médios e braços espirais intermediários. Outro exemplo é a NGC 1300 na Figura 5 b);
- **SBc** Possuem bojo e barra pequenos, braços espirais grandes e mais abertos. Como exemplo a NGC 7741 na Figura 5 c);.

Figura 5 – Galáxias Espirais Barradas, da esquerda para direita: a) NGC 1291 (SBa); b) NGC 1300 (SBb); c) NGC 7741 (SBc).



Fonte: NGC1291 (JARRETT et al., 2003) <http://ned.ipac.caltech.edu/img/2003AJ... .125..525J/2MASS_NGC_1291_JHK.jpg>, NGC 1300 <http://ned.ipac.caltech.edu/ img/2005HSTHH.C.....:/NGC_1300:I:GRB:HST2005.jpg>, NGC 7741 <http://ned.ipac. caltech.edu/img/1994CAG2..B...0000S/NGC7741:I:103aO:s1994.jpg>.

2.1.5 IRREGULARES (I)

As galáxias irregulares não possuem qualquer simetria esférica e axial, ou seja, são as galáxias que não se encaixam nas classificações anteriores. Estas possuem estrelas de população tanto jovens como velhas. Exemplos podem ser vistos na Figura 6.

Figura 6 – Galáxias Irregulares, da esquerda para direita: a) Grande Nuvem de Magalhães; b) Pequena Nuvem de Magalhães.



Fonte: LMC <http://ned.ipac.caltech.edu/img/1994CAG2..B...0000S/LMC:I:103aE+RPlex:s1994.jpg>, SMC <http://ned.ipac.caltech.edu/img/1994CAG2..B...0000S/SMC:I:W10:s1994.jpg>.

2.1.6 GALÁXIAS COM ANEL POLAR

Segundo Beghetto Junior (2015 apud FINKELMAN; FUNES; BROSCH, 2012), galáxias com Anel Polar PRG⁴ são alguns dos objetos mais marcantes da grande família de galáxias aneladas. Em geral são formadas por uma galáxia primária (hospedeira) precoce, lenticular ou elíptica, rodeada por um anel estrelas e gás orbitando um plano quase perpendicular com relação ao plano da primária (RESHETNIKOV et al., 2006). Na Figura 7 temos uma imagem da galáxia NGC 660.

Whitmore et al. (1990) trabalhou na classificação de galáxias com anel polar e mais recentemente Bournaud e Combes (2003) no processo de formação, os trabalhos sugerem que:

- Galáxias hospedeiras são morfologicamente semelhantes às S0, mas suas fotometrias são típicas de espirais precoces, ou espirais do tipo tardio ricas em gás;
- Anéis polares podem ser tão grandes quanto ao raio do disco de sua hospedeira, e tão massivos quanto suas hospedeiras;
- Galáxias com anel polar podem possuir anéis duplos e anéis com braços espirais.

Conforme Lemes (2010), a identificação de galáxias com anel polar nem sempre é simples, pois é necessário que o anel e a hospedeira tenham velocidades sistemáticas semelhantes e que seus centros praticamente coincidam. A aparência de uma PRG, conforme



Figura 7 – Galáxia com Anel Polar NGC 660.

Fonte: NASA <https://apod.nasa.gov/apod/ap170217.html>.

o proposto por Whitmore et al. (1990) em seu catálogo, baseadas nas confiabilidades dadas pelas combinações dos ângulos de rotação das candidatas pode ser visto em detalhes na Figura 8.

Classificamos as PRGs em duas famílias:

- *Spindle*: PRGs tradicionais, com eixo da galáxia em forma de protuberância. O anel polar é quase perpendicular com relação ao plano da galáxia hospedeira. Quando o anel é bem definido e com pouca sobra de material classificamos como Spindle I, quando possui sobras Spindle II.
- Saturn: PRG com bojo esférico circulada por anel brilhante.

O estudo da dinâmica de galáxias com anel polar é uma tarefa difícil, pois o período de evolução observado é insignificante em comparação com o tempo de vida de qualquer galáxia, por isso, tenta-se estudá-las através de simulações de colisão com o objetivo de observar formação apenas pelo processo de acréscimo.

Figura 8 – A aparência de uma PRG e suas possíveis orientações, com inclinação α em relação ao plano xy e β e com respeito ao eixo y.



Fonte: Whitmore et al. (1990).

2.2 FÍSICA CLÁSSICA

A Física clássica utilizada no estudo de órbitas foi tratada por Kepler, que forneceu um arcabouço matemático e geométrico para tal e por Newton, que deu um tratamento físico e explicou esse arcabouço matemático de sistemas gravitantes.

2.2.1 KEPLER

No livro "Astronomia Nova" (KEPLER, 1609), o astrônomo e matemático alemão Johannes Kepler (1571-1630), publicou um estudo sobre o movimento dos planetas. Nesse trabalho ele sugere correções aos modelos anteriores propostos por Cláudio Ptolomeu, Nicolau Copérnico e Tycho Brahe e propõe as leis que hoje são conhecidas como 1^a e 2^a Leis de Kepler. Apesar de não concordar com as propostas de Brahe, estas descobertas só foram possíveis graças aos seus dados observacionais.

Por fim, Kepler levou mais de uma década para descobrir e propor sua 3^a Lei no livro *"Harmonices Mundi"* (KEPLER, 1619) - Harmonias do mundo, onde descobriu a

relação entre o comprimento do semieixo maior de uma órbita⁵ e o período orbital⁶.

Se compararmos as ideias de Copérnico⁷ com Kepler é possível perceber que Kepler não negou suas ideias, mas propôs correções, conforme a Tabela 1.

Tabela 1 – Comparação entre as ideias de Copérnico e Kepler para o movimento dos planetas.

Copérnico	Kepler
As órbitas dos planetas são circulares	As órbitas dos planetas não são
	circulares, mas elípticas
O Sol está no centro da órbita	O Sol não está no centro da órbita,
	mas em um dos focos de uma órbita
	elíptica
A velocidade do planeta em órbita é	A velocidade angular do planeta não é
constante	constante. A velocidade com a qual o
	planeta percorre uma área é constante

Fonte: Dados obtidos a partir de (KEPLER, 1609) e (KEPLER, 1619).

2.2.1.1 1^a LEI DE KEPLER

"Planetae moventur in ellipsibus (Ellipses Keplerianae), in quarum foco Sol est". Os planetas se movem em elipses (elipses assim chamadas de Keplerianas), em cujo foco o sol está (KEPLER, 1609).

Kepler propõe que ao contrário do pressuposto, as órbitas são elípticas e sabendo que um círculo é uma elipse de excentricidade zero, esta lei consegue explicar tanto as órbitas circulares, como as mais alongadas de maior diferença de tamanhos entre seus semieixos maior e menor. Um exemplo clássico de diferentes excentricidades de órbitas pode ser visto na Figura 9.

Uma elipse é definida pela equação

$$d_{PF1} + d_{PF2} = 2a, (2)$$

onde a é o semieixo maior, d_{PF1} e d_{PF2} são as distâncias entre os pontos P pertencentes elipse e os focos.

Representamos órbitas elípticas, conforme a Figura 10, onde b é o semieixo menor.

⁵ Definimos como órbita uma trajetória curva que um corpo percorre gravitando um sistema.

⁶ Definimos como período orbital o tempo de uma órbita completa.

⁷ Como objetivo de explicar as diferenças nas órbitas dos planetas e as órbitas circulares, Copérnico utilizou epiciclos.



Figura 9 – Diferença da excentricidade de órbitas, do Livro Astronomia Nova.

Fonte: Astronomia Nova (KEPLER, 1609).

Figura 10 – Detalhando a relação dos raios com o semieixo maior de uma elipse.



Fonte: O Autor.

A distância c do centro a um foco é dada por: $c=\sqrt{a^2-b^2}$, e a excentricidade eda elipse é dada por: $e=\frac{c}{a}=\frac{\sqrt{a^2-b^2}}{a}.$

Usando a lei dos cossenos⁸ em um triângulo como na Figura 10, considerando os lados d_{PF1} , d_{PF2} e 2c, com θ sendo o ângulo oposto ao lado d_{PF1} , chegamos a

$$d_{PF1}^2 = d_{PF2}^2 + (2ae)^2 + 2d_{PF2}(2ae)\cos\theta.$$
(3)

⁸ A lei dos cossenos nos permite utilizar uma expressão mais generalizada do Teorema de Pitágoras, aplicada a qualquer triângulo.

Utilizando a equação 2 e efetuando algumas manipulações algébricas em 4, chegamos à expressão 5 que nos permite calcular a distância de um planeta ao Sol ao longo de sua órbita elíptica.

$$d_{PF1} = 2a - d_{PF2}$$

$$(2a - d_{PF2})^2 = d_{PF2}^2 + 4a^2e^2 + 4d_{PF2}ae\cos\theta$$

$$4a^2 + d_{PF2}^2 - 4ad_{PF2} = d_{PF2}^2 + 4a^2e^2 + 4d_{PF2}ae\cos\theta$$

$$a^2(1 - e^2) = ad_{PF2}(1 + e\cos\theta)$$
(4)

$$d_{PF2} = \frac{a(1-e^2)}{1+e\cos\theta}.$$
(5)

2.2.1.2 2^a LEI DE KEPLER

"Linea (vector radii, radius conductorius), quae a Sole usque ad planetam traicitur, in periodis paribus superficies pares repingit. Qua de causa planeta prope Solem celerius movet quam procul a Sole (Lex arearum)". A linha (vetor raio ou raio condutor) que liga o planeta ao Sol varre áreas iguais em tempos iguais. Por esta razão, o planeta na proximidade do Sol se move mais rápido do que quando mais distante do Sol (Lei das áreas) (KEPLER, 1609).

Aqui Kepler propõe que, ao contrário do pressuposto, não são as velocidades que são constantes, mas o que ele chamou de *velocidade areal* ou velocidade das áreas, ou seja, o módulo da velocidade varia mas as áreas cobertas por intervalo de tempo são iguais. A Figura 11 mostra 3 áreas iguais cobertas em intervalos de tempo iguais sobre uma órbita elíptica.





Fonte: O Autor.

Kepler também definiu como *"perihelion"* (periélio) a menor distância do planeta ao Sol, bem como *"aphelion"* (afélio) a maior distância do planeta ao Sol.

O cálculo da área de uma elipse é dado pela integral do elemento de área na equação 6, onde r é o raio a partir da origem e θ é o ângulo do raio com o eixo x positivo, dado por

$$A = \int \int dA \qquad Onde: \begin{cases} x = ar\cos\theta\\ y = br\sin\theta \end{cases}$$
(6)

Calculamos o elemento de área através do Jacobiano

$$J = \begin{vmatrix} a\cos\theta & -ar\sin\theta \\ b\sin\theta & br\cos\theta \end{vmatrix} = abr.$$
 (7)

Integrando o elemento de área, chegamos à expressão

$$A_{elipse} = \int_0^{2\pi} \int_0^1 abr \, \mathrm{d}r \, \mathrm{d}\theta = \pi ab. \tag{8}$$

2.2.1.3 3^a LEI DE KEPLER

"Quadrata Periodorum orbitalium duarum planetarum cubis semiaxium maiorum respondent". Os quadrados dos períodos orbitais dos planetas são proporcionais aos cubos dos semieixos maiores (KEPLER, 1619).

Nesta lei, Kepler propõe que os planetas mais próximos ao Sol têm menor período orbital, ao passo que os mais distantes têm maior período orbital. Essa relação é vista na equação (9), onde P é o período orbital, e a é o semieixo maior.

$$\frac{P^2}{a^3} = constante \tag{9}$$

A partir dos dados de período orbital e semieixo maior obtidos do sistema solar, aplicando a 3^a Lei de Kepler, obtemos dados que corroboram sua relação. Estes dados estão disponíveis na Tabela 2.

	P (anos)	a (UA)	P^2	a^3	$\frac{P^2}{a^3}$
Mercúrio	0,241	0,387	0,05808	0,05796	1,00208
Vênus	0,615	0,723	0,37822	0,37793	1,00077
Terra	1,000	1,000	1,00000	1,00000	1,00000
Marte	1,881	1,524	3,53816	3,53961	0,99959
Júpiter	11,862	5,203	140,70704	140,85150	0,99897
Saturno	29,456	9,534	867,65594	866,61349	1,00120
Urano	84,323	19,229	7110,36833	7110,00815	1,00005
Netuno	164,790	30,110	27155,74410	27298,09033	0,99479

Tabela 2 – Aplicação da 3ª Lei de Kepler nos valores do sistema solar.

Fonte: O valores de "P" e "a" foram obtidos de (KEPLER; SARAIVA, 2013).

2.2.2 **NEWTON**

No livro "Philosophiae naturalis principia mathematica" (NEWTON, 1687), o astrônomo, filósofo, físico e matemático inglês Sir Isaac Newton (1642-1726) publicou um estudo sobre a Gravitação Universal. Newton parte dos trabalhos observacionais de Tycho Brahe, do trabalho teórico de Johannes Kepler e de outros estudiosos e então propõe as bases do que hoje conhecemos como Gravitação Universal.

2.2.2.1 GRAVITAÇÃO UNIVERSAL

A Gravitação Universal surge da tentativa de explicar a origem da força centrípeta na Lua estudando a 3^a Lei de Kepler. Newton deduziu que quaisquer dois corpos se atraem gravitacionalmente por meio de uma força que é diretamente proporcional ao produto de suas massas e inversamente proporcional ao quadrado de sua distância. Esta proposta concordava com a ideia central da 3^a Lei de Kepler, que assumia que o período Orbital caía com o semieixo maior, ou seja, quanto mais distantes dois corpos estiverem, menor a força.

Uma das suposições foi que se dois corpos não são pontuais, a distância estabelecida entre eles deve ser obtida em relação ao seu centro de massa, ou seja, pontos onde pode-se supor que está concentrada toda a massa do corpo ou sistema. Esta suposição é uma das bases de sistemas de N-Corpos interagindo gravitacionalmente.

A partir deste ponto será adotada a seguinte notação: um corpo qualquer será denotado pelo índice n, onde \vec{x}_n é sua posição em metros (m), m_n sua massa em quilogramas
(kg), \vec{a}_n sua aceleração em m/s² e \vec{F}_n a força que atua em n em Newtons (N).

A formulação clássica para Lei da Gravitação Universal pode ser vista na equação (10), onde r é a distância entre os corpos em metros (m) e G é a constante gravitacional $G = 6,67428 \times 10^{-11} \frac{\text{m}^3}{\text{kg s}^2}$.

$$\|\vec{F}\| = G \frac{m_1 m_2}{r^3} \vec{r} \tag{10}$$

2.2.2.2 TEOREMAS DE NEWTON

Newton propôs dois teoremas, que nos permitem calcular o potencial gravitacional de qualquer distribuição de massa com simetria esférica:

- Primeiro Teorema: Um corpo que está dentro de uma casca esférica de matéria não sofre força gravitacional líquida daquela casca.
- Segundo Teorema: A força gravitacional atuando sobre um corpo que fica fora de uma casca esférica de matéria é a mesma que sofreria caso a massa de toda casca fosse concentrada em um ponto em seu baricentro.

2.2.3 INTERAÇÃO GRAVITACIONAL

Esta seção tem o objetivo de fazer uma breve introdução sobre a interação gravitacional tratada nos problemas de 2, 3 e n corpos. Estes conceitos serão utilizados na Dinâmica de Sistemas Estelares na Seção 2.3.

2.2.3.1 INTERAÇÃO GRAVITACIONAL DE DOIS CORPOS

O problema de dois corpos é o estudo do movimento de dois corpos quaisquer, interagindo apenas através da atração gravitacional. Uma utilização é tratar a aceleração resultante no movimento de um sistema binário sofrendo atração gravitacional majoritariamente um do outro, supondo $n_1 e n_2$ isolados de qualquer outro tipo de força. As acelerações atuando em cada corpo serão

$$\vec{a}_1 = -Gm_2 \frac{\vec{x}_1 - \vec{x}_2}{\|\vec{x}_1 - \vec{x}_2\|^3} \qquad e \qquad \vec{a}_2 = -Gm_1 \frac{\vec{x}_2 - \vec{x}_1}{\|\vec{x}_2 - \vec{x}_1\|^3},\tag{11}$$

e a aceleração resultante é dada por

$$\vec{a}_2 - \vec{a}_1 = -G \ (m_1 + m_2) \ \frac{\vec{x}_2 - \vec{x}_1}{\|\vec{x}_2 - \vec{x}_1\|^3}.$$
(12)

Essa aceleração relativa do corpo 2 com respeito ao corpo 1 é utilizada para atualizar a velocidade do sistema binário como na Figura 12.

Figura 12 – Dois corpos com massas iguais e vetores velocidade opostos, orbitando seu centro de massa ${\cal C}_m.$



Fonte: O Autor.

2.2.3.2 INTERAÇÃO GRAVITACIONAL DE TRÊS CORPOS

O problema de três corpos refere-se ao estudo do movimento de três corpos quaisquer, interagindo apenas através da atração gravitacional. Diferente do problema de dois corpos, a solução analítica não é tão simples e aqui começam as aproximações numéricas utilizadas posteriormente no problema de N-Corpos.

O físico e matemático francês Jean le Rond d'Alembert publicou um trabalho chamado de *Traité de dynamique* em 1747, conhecido por ser um dos primeiros a trabalhar na tentativa de encontrar uma solução do problema.

Em 1760 Leonhard Euler, outro físico e matemático suíço, publicou nova abordagem do problema com uma solução analítica utilizando integrais elípticas, para o caso de dois campos gravitacionais fixos e uma massa orbitando-os livremente.

2.2.3.3 INTERAÇÃO GRAVITACIONAL DE N-CORPOS

O problema dos N-Corpos consite no estudo do movimento de n corpos quaisquer, interagindo majoritariamente através da sua atração gravitacional mútua. A solução deste problema veio da necessidade de entender sistemas com muitos corpos interagindo gravitacionalmente, dentre estes, destaca-se a tentativa de modelar o funcionamento de nosso sistema solar, de aglomerados de estrelas, galáxias, etc.

A formulação clássica do problema de N-Corpos gravitacional é relativamente simples. Dados valores de massas, posições e velocidades de n corpos, a tarefa é calcular as trajetórias futuras. Apesar dos movimentos serem determinados por equações diferenciais, soluções mais precisas são obtidas através de métodos numéricos (AARSETH, 2008).

O princípio da tratativa desse problema se deu no trabalho de Newton (ver Seção 2.2.2), onde ele simplifica a geometria dos corpos com simetria esférica, modelando-os como partículas de massa pontual.

A solução geral considera n massas pontuais m_i , onde i = 1, 2, 3, 4, ..., n em um sistema inercial⁹, unicamente sob influência de força gravitacional de seus n corpos interagindo em função do tempo t. Cada uma destas n partículas possui um vetor posição \vec{x}_i . Partindo da segunda Lei de Newton que diz que a massa m_i multiplicada pelo seu vetor aceleração \vec{a}_i , na equação (13) é a força resultante agindo sobre m_i

$$\vec{F}r = m_i \frac{\mathrm{d}^2 \vec{x}_i}{\mathrm{d}t^2}.$$
(13)

A lei da Gravitação de Newton diz que a força gravitacional F_{ij} que atua na partícula de massa m_i é dada pela equação

$$\vec{F}_{ij} = Gm_i m_j \frac{\vec{x}_j - \vec{x}_i}{\|\vec{x}_j - \vec{x}_i\|^3}.$$
(14)

O somatório das forças atuando sobre cada massa do sistema resulta na solução do problema dos N-Corpos gravitacional, dado pela equação

$$m_{i} \frac{\mathrm{d}^{2} \vec{x}_{i}}{\mathrm{d}t^{2}} = \sum_{\substack{j=1\\j\neq i}}^{n} \frac{Gm_{i}m_{j}(\vec{x}_{j} - \vec{x}_{i})}{\|\vec{x}_{j} - \vec{x}_{i}\|^{3}} = \frac{\partial U}{\partial \vec{x}_{i}}.$$
(15)

⁹ Para que algo seja considerado um sistema inercial, a resultante da aceleração tem que ser zero, ou seja, precisa estar parado ou em movimento com velocidade constante.

Onde U é a energia potencial

$$U = \sum_{1 \le i < j \le n} \frac{Gm_i m_j}{\|\vec{x}_j - \vec{x}_i\|}.$$
 (16)

O momento é dado por

$$\vec{p_i} = m_i \frac{\mathrm{d}\vec{x_i}}{\mathrm{d}t}.\tag{17}$$

A energia cinética Tdo sistema será

$$T = \sum_{i=1}^{n} \frac{\|\vec{p}_i\|^2}{2m_i}.$$
(18)

O Hamiltoniano do sistema será dado pelas equações

$$\frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t} = \frac{\partial H}{\partial \vec{p}_i}$$

$$\frac{\mathrm{d}\vec{p}_i}{\mathrm{d}t} = -\frac{\partial H}{\partial \vec{x}_i}.$$
(19)

Onde definimos o Hamiltoniano como a soma das energias potencial Ue cinética ${\cal T}$ do sistema

$$H = U + T. (20)$$

O sistema Hamiltoniano do problema de N-Corpos é um sistema de 6 equações diferenciais ordinárias, em um espaço de fase 6N-dimensional. Um ponto neste espaço de fase, chamamos de microestado do sistema.

O centro de massa de um sistema de N-corpos pode ser calculado por

$$\vec{C} = \frac{\sum_{i=1}^{n} m_i \vec{x}_i}{\sum_{i=1}^{n} m_i}.$$
(21)

2.3 DINÂMICA DE SISTEMAS ESTELARES

Os encontros de sistemas estelares têm modificado substancialmente as estruturas cósmicas ao longo de toda a evolução do Universo. Estes fenômenos são implacavelmente determinados pelo caráter puramente atrativo da força da gravidade que, por sua vez, induz na interação dos grandes sistemas a força de maré, o atrito dinâmico, etc (RODRIGUES, 1999).

Um sistema estelar é um agrupamento de estrelas ou outras massas pontuais, gravitacionalmente ligado (BINNEY; TREMAINE, 2011). O comportamento desses sistemas é determinado pelas leis de movimento e da Gravitação de Newton, sendo o estudo desse comportamento tratado na **dinâmica de sistemas estelares** (BINNEY; TREMAINE, 2011).

Sem técnicas numéricas, o imenso progresso feito nesses campos teria sido quase impossível, uma vez que os cálculos analíticos são frequentemente restritos a problemas idealizados de alta simetria ou a aproximação de tratamentos de problemas inerentemente não-lineares (SPRINGEL; YOSHIDA; WHITE, 2001).

O uso de simulações numéricas de sistemas gravitantes possibilita uma melhor abstração de informações destas interações. No presente trabalho tratamos do problema específico das galáxias em interação. Para realizarmos estas simulações de galáxias precisamos gerar modelos fisicamente plausíveis, calcular os parâmetros necessários para uma trajetória de colisão e por fim lançá-los um contra o outro.

O procedimento básico para que sejam realizadas estas simulações parte da criação de dois modelos de galáxias isoladas. Em seguida, caso seja pertinente, rotacionamos os modelos em relação a um ou mais eixos e por fim calculamos sua trajetória, distância de pericentro e velocidades relativas. Com essas informações já é possível juntarmos os dois modelos em arquivo de condições iniciais e simular a interação durante a evolução temporal. As simulações realizadas neste trabalho utilizaram o código $GADGET-2^{10}$ na versão 2.0.7.

2.3.1 TRATAMENTO FÍSICO DE SIMULAÇÕES

Partimos do princípio que a força gravitacional que atua sobre uma estrela é espacialmente suave, ou seja, supomos que cada estrela acelera no campo de força gravitacional

¹⁰ O software GADGET-2 (**GA**laxies with **D**ark matter and **G**as int**E**rac**T**) pode ser obtido em <https://wwwmpa.mpa-garching.mpg.de/gadget/> por ser distribuído sobre licença GNU GPL.

gerado pela galáxia como um todo. Por isso é possível tratar a força gravitacional que age nas estrelas como uma distribuição suave em vez de massas pontuais.

Por adotarmos esse tratamento com potenciais contínuos, utilizamos um espaço de fase 6N-dimensional e descrevemos nossos potenciais em forma de funções de densidade das componentes no modelo, por exemplo, bojo, halo, disco e gás.

Neste trabalho vamos utilizar o problema de N-Corpos gravitacional na Dinâmica de Sistemas Estelares para simular alguns cenários de colisão de galáxias espirais, com o objetivo de acompanhar a evolução de sua estrutura morfológica ao longo de sua evolução temporal.

2.3.1.1 MODELOS NUMÉRICOS DE GALÁXIAS

Os modelos numéricos de galaxias são construídos utilizando o código *MakeNewDisk*, que apesar de não ser distribuído livremente, foi gentilmente fornecido pelo autor Dr. Volker Springel.

Cada modelo de galáxia consiste em um halo de matéria escura, um bojo e um disco de gás e estrelas com suporte de rotação(SPRINGEL; MATTEO; HERNQUIST, 2005). O código permite a construção de modelos onde as componentes são parametrizadas de forma independente. Como consequência desse detalhamento independente de suas componentes, podemos criar modelos com uma ampla gama de tipos morfológicos.

2.3.1.1.1 Halo de matéria escura

O halo de matéria escura é modelado em formato esférico, a partir de uma função de densidade com um perfil de Hernquist (SPRINGEL; MATTEO; HERNQUIST, 2005 apud HERNQUIST, 1990), onde

$$\rho_{dm}(r) = \frac{M_{dm}}{2\pi} \frac{a}{r(r+a)^3},$$
(22)

onde M_{dm} é a massa de matéria escura, r é o raio e a é um fator de escala, onde

$$a = \sqrt{2\left[\ln(1+c) - \frac{c}{1+c}\right]},$$
(23)

cé um índice de concentração, dado por

$$c = \frac{r_{200}}{r_s},$$
 (24)

onde r_{200} é o raio onde a densidade de matéria escura fechada por este é 200 vezes a densidade crítica do universo¹¹, e r_s é o um fator de escala do halo no modelo proposto por Navarro, Frenk e White (1996).

2.3.1.1.2 Bojo

O bojo é modelado pela mesma função de densidade do halo, também esférico com perfil de Hernquist (SPRINGEL; MATTEO; HERNQUIST, 2005 apud HERNQUIST, 1990), dado por

$$\rho_{db}(r) = \frac{M_b}{2\pi} \frac{b}{r(r+b)^3},$$
(25)

onde M_b é a massa do bojo, r é o raio e b é um fator de escala do bojo.

2.3.1.1.3 Disco de Estrelas e Gás

As componentes de estrelas e gás do disco são modeladas com um perfil exponencial de densidade de superfície, com comprimento de escala h, dado por

$$\sum_{estrelas}(r) = \frac{M_{estrelas}}{2\pi\hbar^2} e^{\frac{-r}{\hbar}},\tag{26}$$

$$\sum_{gas}(r) = \frac{M_{gas}}{2\pi\hbar^2} e^{\frac{-r}{\hbar}},\tag{27}$$

onde a massa do disco $M_{disco} = M_{estrelas} + M_{gas}$ e h é dada em relação ao momento angular do disco conforme sugerido por Springel e White (1999).

¹¹ Segundo Team (2014), a estimativa atual para densidade crítica do universo é de aproximadamente cinco átomos de hidrogênio por metro cúbico.

2.3.1.2 DINÂMICA GRAVITACIONAL SEM COLISÃO

Sendo \vec{x} o vetor posição e \vec{v} o vetor velocidade de um corpo, a massa contida em um elemento de volume dxdv no ponto (\vec{x}, \vec{v}) do espaço de fase no tempo t pode ser definida como uma função de distribuição

$$f(\vec{x}, \vec{v}, t) \mathrm{d}x \mathrm{d}v = M_t(\vec{x}, \vec{v}). \tag{28}$$

A matéria escura e as estrelas são modeladas como fluidos auto-gravitantes sem colisão, ou seja, elas cumprem a equação de Boltzmann para o caso não colisional (CBE) (SPRINGEL; YOSHIDA; WHITE, 2001), dada por

$$\frac{\mathrm{d}f}{\mathrm{d}t} \equiv \frac{\partial f}{\partial t} + \vec{v}\frac{\partial f}{\partial \vec{x}} - \frac{\partial \Phi}{\partial \vec{x}}\frac{\partial f}{\partial \vec{v}} = 0,\tag{29}$$

onde denotamos por Φ o potencial que se relaciona com a função de distribuíção pela equação de Poisson,

$$\nabla^2 \Phi(\vec{x}, t) = 4\pi G \int f(\vec{x}, \vec{v}, t) \mathrm{d}v.$$
(30)

Como as simulações deste trabalho têm uma natureza não colisional e não há formação estelar, a equação 29 é uma homogênea, ou seja, não são criadas nem destruídas estrelas neste tipo de simulação.

Este é essencialmente um método de Monte Carlo, aproximando f contínua por uma amostragem de n corpos, que possuem massa m_i , vetor posição $\vec{x_i}$ e vetor velocidade $\vec{v_i}$, para todo $i \in 1, ..., n$. Em simulações auto-consistentes, o campo gravitacional Φ é gerado pelos n corpos. Portanto, não se deve confundir os corpos nos modelos com as estrelas do sistema físico real, pois cada corpo numa simulação pode representar um conjunto de muitas estrelas em um sistema estelar (BEGHETTO JUNIOR, 2015 apud RODRIGUES, 1999).

2.3.2 ALGORITMOS DE CÁLCULO DE FORÇA

A partir deste ponto começamos a tratar os algoritmos utilizados em uma simulação de N-Corpos gravitacional para o cálculo de forças.

2.3.2.1 SOMA DIRETA

O primeiro algoritmo de cálculo de forças é chamado de Soma Direta, pois envolve o cálculo de todas as forças que atuam sobre cada partícula. Neste algoritmo nenhuma simplificação é feita, por isso, dizemos que ele tem uma complexidade computacional¹² $O(N^2)^{13}$.

Um dos primeiros a trabalhar com algoritmos de Soma Direta foi Aarseth (1963), que produziu diversas otimizações usadas nos algoritmos de N-Corpos, como o parâmetro ϵ usado na suavização na equação de forças, dada por

$$\vec{F}_i = -\sum_{j \neq i} \frac{Gm_i m_j (\vec{x}_j - \vec{x}_i)}{(\|\vec{x}_j - \vec{x}_i\|^2 + \epsilon^2)^{3/2}}.$$
(31)

O parâmetro ϵ , conhecido como parâmetro de *softening* na equação 31 impede que o denominador fique perigosamente pequeno, resultando em uma força que tenda para o infinito (AARSETH; HOYLE, 1964). Hoje, em vez de um parâmetro fixo, utilizamos um kernel de suavização, com uma função que tende a zero à medida que as partículas se distanciam, para minimizar erros no cálculo da força.

2.3.2.2 TREE CODE

Em uma evolução dos códigos de soma direta, surgiram os modelos hierárquicos como o proposto por Joshua Barnes e Piet Hut denominado *Tree Code* (BARNES; HUT, 1986). Os *Tree Codes* são baseados em árvores com 8 terminações chamadas de *octree*.

Este algoritmo é usado para subdividir o volume contido na caixa de simulação em células cúbicas, de modo que apenas partículas de células próximas precisam ser tratadas individualmente. As partículas em células distantes são tratadas como uma única grande partícula reduzida em seu centro de massa.

Um exemplo desse modelo é dado na Figura 13. Esta aproximação diminui o custo computacional do algoritmo para $O(NlogN)^{14}$.

¹² Complexidade computacional é uma subárea da Ciência da Computação e da Matemática Computacional que se concentra em classificar problemas computacionais de acordo com sua dificuldade inerente.

¹³ A notação O grande é uma notação matemática que descreve o comportamento limitante de uma função quando o argumento tende para um valor particular ou infinito. Para o caso $O(N^2)$ temos um esforço quadrático. Dentre os exemplos deste tipo, temos o uso de algoritmos simples de ordenação como o *bubble sort*, ou ainda neste caso a interação partícula-partícula, ou n * n, em um sistema de



Figura 13 – Modelo hierárquico baseado na árvore octree de Joshua Barnes e Piet Hut.

Fonte: O Autor.

O cálculo de forças é feito percorrendo os caminhos da árvore somando as contribuições de cada um dos nós. Na caminhada padrão de uma *octree* de Barnes e Hut (1986), a expansão multipolar de um nó de tamanho l é usada somente se

$$x > \frac{l}{\theta},\tag{32}$$

onde x é a distância de uma partícula ao centro de massa da célula e θ é um parâmetro de precisão atribuído pelo usuário.

Ainda na evolução dos modelos hierárquicos, começam a surgir códigos paralelos com balanceamento de carga como o proposto por Dubinski (1996). Este código utiliza balanceamento pelo método da bissecção recursiva ortogonal para subdividir o sistema em volumes retangulares independentes, onde cada um dos volumes é processado em paralelo em diferentes núcleos. Este modelo pode ser visto na Figura 14.

N-Corpos.

¹⁴ Para o caso O(NlogN) temos um esforço log-linear. Dentre os exemplos deste tipo, temos o uso da Transformada Rápida de Fourier e os algoritmos mais rápidos de ordenação, como o *mergesort* ou *heapsort*, ou ainda neste caso a interação de partículas em modelos hierárquicos como os usados nos Tree Codes.

Figura 14 – Modelo hierárquico baseado na bissecção recursiva de John Dubinski, alocando recursos diretamente para núcleos de processamento.



Fonte: Dubinski (1996) <https://www.cita.utoronto.ca/~dubinski/treecode/node9.html>.

2.3.2.3 SPH E O TREESPH

Smoothed Particle Hydrodynamics é uma técnica Lagrangeana utilizada para resolver problemas de hidrodinâmica. Conforme Springel (2005) SPH utiliza partículas traçadoras para descrever o estado de um fluido através de quantidades contínuas. Segundo Springel, Yoshida e White (2001) SPH é adequada para resolver problemas astrofísicos tridimensionais que não necessitem de precisão em frentes de choque. As partículas de gás possuem vetor posição \vec{x}_i , vetor velocidade \vec{v}_i e massa m_i , o cálculo de sua densidade é dado por

$$\rho_i = \sum_{j=1}^n m_j W(||\vec{x}_{ij}||, h_i), \tag{33}$$

onde $\vec{x}_{ij} = \vec{x}_i - \vec{x}_j$, h é o número de vizinhos e $W(||x_{ij}||, h_i)$ é o kernel de suavização, definido por

$$W(x,h) = \frac{8}{\pi h^3} \begin{cases} 1 - 6(\frac{x}{h})^2 + 6(\frac{x}{h})^3, 0 \le \frac{x}{h} \le \frac{1}{2} \\ 2(1 - \frac{x}{h})^3, \frac{1}{2} < \frac{x}{h} \le 1 \\ 0, \frac{x}{h} > 1 \end{cases}$$
(34)

O *TreeSPH* de Hernquist e Katz (1989) é uma implementação do SPH onde seu modelo hierárquico é baseado no *Tree Code*. Por ser mais flexível que outras implementações é uma das formas adotadas pelo GADGET-2 no tratamento de fluidos auto-gravitantes tridimensionais.

2.3.3 O TRATAMENTO DO TEMPO NAS SIMULAÇÕES

Nem sempre é viável medir a escala de tempo de uma simulação em anos ou segundos. Em nosso caso, para modelos de galáxias, definimos 2 tempos fundamentais: o tempo de cruzamento t_{cr} e o tempo de relaxação t_{relax} .

Definimos o tempo cruzamento t_{cr} como o tempo que uma partícula teste leva para cruzar uma distância do tamanho do raio do modelo em que a partícula se encontra, dado por

$$t_{cr} = \frac{\vec{r}}{\vec{v}},\tag{35}$$

onde \vec{v} é sua velocidade e \vec{r} é o raio do modelo, que pode ser o raio de um aglomerado, galáxia ou de uma região de interesse.

Já o tempo de relaxação t_{relax} é definido como o tempo que uma galáxia leva para entrar em um pseudo equilíbrio,

$$t_{relax} = \frac{0, 1N}{\ln(N)} t_{cr},\tag{36}$$

onde N é o número de partículas no sistema.

Pioneiro na área de N-corpos Aarseth (1963) frequentemente utilizava frações do tempo t_{relax} (AARSETH; HOYLE, 1964) como parâmetro para definir o passo de tempo (*timestep*) fixo utilizado nos integradores de tempo,

$$\Delta t = \alpha t_{relax},\tag{37}$$

com α variando entre 0,02 e 0,03.

2.3.3.1 LEAPFROG

O Leapfrog é um dos integradores simpléticos¹⁵ mais utilizados na integração temporal envolvendo problemas de N-corpos gravitacional. Segundo Quinn et al. (1997) o Leapfrog tem várias vantagens em relação a outros métodos:

- 1. Para precisão de segunda ordem, apenas um cálculo de força e uma cópia do estado físico do sistema são necessários. Isto é particularmente benéfico para simulações de N-Corpos, onde o custo de cálculo de forças é muito elevado;
- 2. O campo de força em uma simulação de N-Corpos não é muito suave, então, uma ordem superior não significa necessariamente maior precisão;
- 3. Sendo um integrador simplético, o Leapfrog, preserva as propriedades específicas dos sistemas hamiltonianos.

Este integrador funciona de forma alternada, primeiro atualizando as coordenadas de posição, conforme

$$\vec{x}_{n+1} = \vec{x}_n + \tau \vec{v}_{n+\frac{1}{2}},\tag{38}$$

onde τ é o passo de tempo. Em seguida, atualizam-se as velocidades, dado por

$$\vec{v}_{n+\frac{3}{2}} = \vec{v}_{n+\frac{1}{2}} + \tau \vec{a}(\vec{x}_{n+1}). \tag{39}$$

Feito isso, repetem-se os passos até chegar no tempo final desejado, conforme a Figura 15.

Figura 15 – Esquema de integração usando Leapfrog.

Fonte: O Autor.



 $^{^{15}\,}$ Um integrador simplético fornece uma solução exata para um sistema hamiltoniano discreto que está próximo do hamiltoniano contínuo.

2.3.3.2 PASSO DE TEMPO ADAPTATIVO

Ao realizarmos simulações de dinâmica de galáxias, trabalhamos com partículas atuando a grandes distâncias em modelos hierárquicos como o *Tree code*. O integrador é uma variante do Leapfrog, mas atuando de forma dinâmica no passo de tempo.

Segundo Springel (2005), em regiões de alta densidade, como nos centros de galáxias, onde reside uma grande fração da massa, são necessários passos de tempo menores do que nas regiões de baixa densidade do meio intergaláctico. Evolução de todas as partículas com o menor passo de tempo necessário implica, portanto, um desperdício substancial de recursos computacionais. Um esquema de integração com etapas de tempo individuais tenta lidar com esta situação de forma mais eficiente.

A principal desvantagem em relação ao Leapfrog original, segundo Quinn et al. (1997), é que o uso de diferentes passos de tempo de forma adaptativa, inviabiliza a reversibilidade do integrador original.

Seu integrador funciona de forma alternada, primeiro atualizando as coordenadas de posição. Esse procedimento é chamado de *drift* $D(\frac{\tau}{2})$, conforme

$$\vec{x}_{n+\frac{1}{2}} = \vec{x}_n + \frac{1}{2}\tau\vec{v}_n,\tag{40}$$

onde τ é o passo de tempo. Em seguida, atualizam-se as velocidades. Esse procedimento é chamado de kick $K(\tau)$, dado por

$$\vec{v}_{n+1} = \vec{v}_n + \tau \vec{a}(\vec{x}_{n+\frac{1}{2}}),\tag{41}$$

e então um novo procedimento de drift $D(\frac{\tau}{2})$, por

$$\vec{x}_{n+1} = \vec{x}_{n+\frac{1}{2}} + \frac{1}{2}\tau\vec{v}_{n+1}.$$
(42)

Feito isso, repetem-se os passos até chegar no tempo final desejado, resultando em iterações de $D(\frac{\tau}{2})K(\tau)D(\frac{\tau}{2})$. Esta representação é equivalente à representação no Hamiltoniano na forma

$$H_n = H_D + H_K = \frac{1}{2}\vec{v}^2 + \Phi(\vec{x}), \tag{43}$$

onde H_D é a atualização da posição, e H_K é a correspondente do potencial gerando a aceleração.

3 CONCEITOS DE COMPUTAÇÃO GRÁFICA

Assim como a Física e a Matemática, a Computação Gráfica também possui um papel importante nesta dissertação, pois é através dela que são criadas ferramentas para visualização de modelos numéricos no computador.

Neste Capítulo é feita uma revisão da Computação Gráfica que utilizamos neste trabalho. Na Seção 3.1 temos uma introdução aos princípios que deram origem à Computação Gráfica, na Seção 3.2 temos uma revisão dos sistemas de coordenadas bem como os arcabouços algébrico e geométrico que são utilizados para tratar os objetos tridimensionais deste trabalho e, por fim, na Seção 3.4 temos os conceitos de Imersão em Realidade Virtual.

3.1 PRINCÍPIOS DE COMPUTAÇÃO GRÁFICA

O termo Computação Gráfica (CG) foi criado por uma equipe de desenvolvedores chefiada por Verne Hudson em 1959 nos laboratórios da Boeing, em meio a um projeto para criação de uma ferramenta CAD (*Computer-Aided Design*).

CG utiliza como base alguns conceitos matemáticos que são considerados seus pilares. Dentre eles, destacam-se:

- A Geometria Euclidiana: criada pelo matemático Euclides de Alexandria entre 300 e 250 a.C. que nos propiciou uma forma de descrever figuras geométricas, que são utilizadas como primitivas para geração de imagens;
- A perspectiva: criada por estudiosos gregos e romanos, posteriormente amadurecida pelo arquiteto italiano Filippo Brunelleschi em 1425 que nos permitiu fazer representações bidimensionais de objetos e paisagens tridimensionais através de perspectiva cônica e pontos de fuga;
- Os sistemas de coordenadas cartesianas bi e tri dimensionais: criados pelo físico e matemático francês René Descartes em 1637 que nos permitiu descrever e posicionar um objeto no espaço.

O legado de outros grandes estudiosos, como Euler, em sua imensa contribuição matemática e Sylvester, em sua notação matricial, serviu de arcabouço ferramental utilizado na Computação Gráfica. Segundo o autor Eduardo Azevedo "A Computação Gráfica é uma área que engloba, para melhor descrição didática, pelo menos três grandes subáreas: a **Síntese de Imagens**, o **Processamento de Imagens** e a **Análise de Imagens**." (AZEVEDO; CONCI, 2003). Neste trabalho utilizamos a Síntese de Imagens como ferramenta para gerar imagens a partir de modelos numéricos tridimensionais, provenientes de simulações de Dinâmica de Galáxias.

Síntese ou Geração de Imagens é o ato de representar de forma visual, objetos virtuais, utilizando pontos e coordenadas. A primitiva básica utilizada para exibir imagens é o pixel (*picture element*), que é o menor elemento que um dispositivo gráfico pode exibir, em geral, representado por um quadrado unitário, suas coordenadas x e y, e suas intensidades de cor. O posicionamento de um pixel em um monitor pode ser visto na Figura 16.

Figura 16 – Representação de um pixel no sistema de coordenadas no monitor.



Fonte: O Autor.

O sistema básico de cor da maioria dos dispositivos, que exibe imagens computacionais é o sistema RGB (*Red, Green and Blue*), visto na Figura 17, com um canal individual para cada cor, onde cada componente de cor primária tem sua intensidade representada em 8 bits, ou seja, 2^8 combinações, que representam 256 intensidades para cada canal de cor. Esta mescla de 3 canais resulta em uma imagem com 24 bits, que nos permite exibir imagens utilizando uma paleta de 256^3 cores ou 16,7 milhões de cores. Figura 17 – Sistema de cores RGB.



Fonte: O Autor.

Apesar de o sistema de cores RGB ser o de uso mais geral, existem outros sistemas com mesma finalidade, que é exibir imagens, mas com viés diferente. O CMYK (*Cian, Magenta, Yellow and Black*), visto na Figura 18, é outro sistema, que por conseguir exibir uma maior paleta de cores que o sistema RGB (4 canais com 8 bits por canal, totalizando 32 bits) é amplamente utilizado para editoração e publicidade. Existem ainda outros sistemas de cores como os derivados do YUV, que utiliza 1 canal de 8 bits para brilho e 2 canais de 8 bits para cor, amplamente utilizado para transmissão de imagens comprimidas em streaming. Neste trabalho foi utilizado o sistema de cores RGB como padrão.

Figura 18 – Sistema de cores CMYK.



Fonte: O Autor.

3.2 ÁLGEBRA E GEOMETRIA UTILIZADAS EM CG

3.2.1 SISTEMAS DE COORDENADAS

Para posicionarmos um objeto, em geral, precisamos definir 4 sistemas de coordenadas:

• SRU - Sistema de coordenadas de referência do universo: este é o sistema utilizado para descrever um objeto no mundo real, utilizado para descrever as dimensões de uma galáxia em parsecs, de um terreno em metros, um carro em centímetros etc.

- SRO Sistema de coordenadas de referência do objeto: este é o sistema utilizado para descrever através de primitivas um modelo físico na forma computacional, por exemplo um modelo que descreve as dimensões em variáveis e escalas próprias de uma galáxia, de um terreno, de um carro etc.
- SRN Sistema de coordenadas de referência normalizado: este é o sistema de coordenadas utilizado pelas bibliotecas gráficas como OpenGL, por exemplo $-1 \le x \le 1, -1 \le y \le 1, -1 \le z \le 1$.
- SRD Sistema de coordenadas de referência do dispositivo: este é o sistema de coordenadas utilizado pelos dispositivos gráficos bidimensionais, como monitores e projetores com X versus Y pixels.

A primeira tarefa ao tentar representar um objeto do mundo físico em CG é fazer uma abstração de suas dimensões físicas no SRU, para criar um modelo computacional representado no SRO através de pontos. Os pontos do SRO serão ligados para descrever arestas, vértices e faces, ou mesmo pontos isolados de forma a representar o modelo físico em um modelo computacional.

Depois de representado nosso modelo no SRO é feita a normalização através de uma transformação de escala, para que o modelo seja descrito no SRN que é o sistema nativo das bibliotecas gráficas. É neste sistema que os dispositivos gráficos fazem a maior parte do processamento, durante a visualização de um objeto no que chamamos de cena, por exemplo, usando transformações de rotação, de escala e mudando a posição desta cena.

Por fim, a biblioteca gráfica utilizada faz novamente uma transformação de escala e perspectiva a fim de converter a cena no SRN em uma imagem bidimensional, compatível com as dimensões do SRD em janela gráfica, que então é exibida através de um dispositivo como monitor ou projetor. A ordem da transformação nos sistemas de coordenadas pode ser vista na Figura 19.

Figura 19 – A ordem da transformação nos sistemas de coordenadas em CG.



Fonte: O Autor.

3.2.2 PRIMITIVAS GRÁFICAS

A fim de representar qualquer objeto sintético em CG, faz-se necessário definirmos um sistema de coordenadas e a forma mais simples é através de uma notação matricial. Qualquer objeto representado na CG é construído através de uma ou mais primitivas, sendo a primeira o ponto.

A primitiva básica é o **ponto**. A partir dela desenhamos todas as outras primitivas. Em geral utiliza-se um vetor linha ou coluna para se representar um ponto em um plano bidimensional, dado por

$$P(x,y) = \begin{bmatrix} x, y \end{bmatrix}$$

ou
$$P(x,y) = \begin{bmatrix} x \\ y \end{bmatrix}$$

(44)

para o caso de um ponto em um plano tridimensional, como segue

$$P(x, y, z) = \begin{bmatrix} x, y, z \end{bmatrix}$$
ou
$$P(x, y, z) = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$
(45)

A segunda primitiva básica é a **linha**. O cientista da computação Jack Elton Bresenham criou em 1962 o algoritmo para traçar linhas retas entre dois pontos determinados, que ficou conhecido como Algoritmo de Bresenham. Uma implementação em C++ deste algoritmo é vista a seguir e a Figura 20 apresenta um resultado da sua aplicação para desenhar uma linha.

/* Algoritmo de Bresenham para traçar linhas retas */

```
void bresenham(int x0, int y0, int x1, int y1) {
    int dx, dy, sx, sy, erro, erro2;
    dx = abs(x1-x0);
    dy = abs(y1-y0);
    sx = x0<x1 ? 1 : -1;
    sy = y0<y1 ? 1 : -1;
    erro = (dx>dy ? dx : -dy)/2;
```

```
for(;;){
   std::cout << "X: " << x0 << " Y: " << y0 << std::endl;
   if (x0==x1 && y0==y1)
        break;
   erro2 = erro;
   if (erro2 >-dx){
      erro -= dy;
      x0 += sx;
   }
   if (erro2 < dy){
      erro += dx;
      y0 += sy;
   }
}</pre>
```





Fonte: O Autor.

Utilizando as duas primitivas ponto e linha, pode-se extrapolar para quaisquer objetos mais complexos bi e tri dimensionais como, por exemplo, elipses, esferas, cubos e outros.

3.2.3 TRANSFORMAÇÕES

Como precisamos constantemente fazer transformações em objetos, como translação, escala, rotação entre estes sistemas é prático adotarmos algumas operações básicas de

transformações.

A primeira transformação é a **translação**, que consiste em deslocar um objeto de uma posição para outra. Para deslocar um objeto, somamos o deslocamento às suas coordenadas,

$$Trans(x, y, z, dx, dy, dz) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} x + dx \\ y + dy \\ z + dz \end{bmatrix}.$$
(46)

A segunda transformação é a **escala**, que consiste em aumentar ou diminuir o tamanho de um objeto. Para fazer uma escala de um objeto, multiplicamos as suas coordenadas por um ou mais fatores de escala, como a seguir

$$Esc(x, y, z, e_x, e_y, e_z) = \begin{bmatrix} x \ y \ z \end{bmatrix} \times \begin{bmatrix} e_x & 0 & 0 \\ 0 & e_y & 0 \\ 0 & 0 & e_z \end{bmatrix} = \begin{bmatrix} x \times ex \\ y \times ey \\ z \times ez \end{bmatrix}.$$
(47)

A terceira transformação é a **rotação**, que consiste em girar um objeto em torno de um eixo. Para isso, multiplicamos as suas coordenadas por uma matriz de rotação. As operações de rotação em torno do eixo x, y, z são mostradas respectivamente, a seguir:

$$RotX(x, y, z, \alpha) = \begin{bmatrix} x \ y \ z \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix},$$
(48)

$$RotY(x, y, z, \beta) = \begin{bmatrix} x \ y \ z \end{bmatrix} \times \begin{bmatrix} \cos(\beta) & 0 & -sen(\beta) \\ 0 & 1 & 0 \\ sen(\beta) & 0 & \cos(\beta) \end{bmatrix},$$
(49)

$$RotZ(x, y, z, \gamma) = \begin{bmatrix} x \ y \ z \end{bmatrix} \times \begin{bmatrix} cos(\gamma) & sen(\gamma) & 0 \\ -sen(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$
 (50)

A quarta transformação é a **reflexão**, que consiste em inverter um objeto com respeito a uma certa direção. Para fazer uma reflexão de um objeto, multiplicamos as suas coordenadas por uma matriz de reflexão. Reflexões nas direções x, y, z são:

$$Ref X(x, y, z) = \begin{bmatrix} x \ y \ z \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
(51)

$$RefY(x, y, z) = \begin{bmatrix} x \ y \ z \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
(52)

$$RefZ(x, y, z) = \begin{bmatrix} x \ y \ z \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$
 (53)

Pode-se ainda fazer combinações destas matrizes de transformações, por exemplo, de rotação e escala para posteriormente multiplicar a matriz resultante pelas coordenadas de um objeto. Durante a execução de um software gráfico, essas operações acontecem constantemente e por terem uma natureza massiva, estas operações são realizadas na GPU *Graphics Processing Unit*, que possibilita a execução de instruções de forma massivamente paralela.

3.3 OPENGL

OpenGL - Open Graphics Library é uma API^1 gráfica, amplamente utilizada para o desenvolvimento de aplicações gráficas como: jogos, CAD, visualizadores de modelos e outros. Desenvolvida em linguagem C e C++ pela Khronos Group é suportada de forma nativa em toda GPU fabricada atualmente e tem constantes atualizações desde seu lançamento em 1991. No desenvolvimento deste trabalho utilizamos amplamente OpenGL, por isso fez-se necessário conceituá-la brevemente.

3.3.1 PROJEÇÃO E POSIÇÃO DE CÂMERA

Como os meios de visualização (monitor, projetor e etc.) são bidimensionais, para que possamos ter a sensação tridimensional é necessário fazer uma projeção em um plano bidimensional. Esta técnica é chamada de perspectiva da imagem.

Para visualização de uma cena em perspectiva, utilizamos uma rotina do OpenGL, dada por

¹ API - *Application Programming Interface* é um conjunto de rotinas e padrões que pode ser utilizado por aplicativos que não objetivam envolver-se em detalhes da implementação do software, mas apenas usar seus serviços

gluPerspective(cdv,aspecto,perto,longe);

onde cdv é o ângulo do campo de visão da cena em questão, *aspecto* é a razão entre a largura e altura, *perto* e *longe* são os limites de distância mínimo e máximo de visualização para montagem da cena. Também utilizamos uma função do OpenGL para definir os parâmetros da câmera em

onde *posCam* é a posição da câmera, *olhaPara* é o ponto que a câmera está apontada e *normal* é o vetor normal que indica o que é o referencial de cima da cena. Um exemplo de projeção e posição de câmera pode ser visto na Figura 21.

Figura 21 – Ilustração da visualização da cena, usando perspectiva e câmera.



Fonte: O Autor.

3.4 IMERSÃO EM REALIDADE VIRTUAL

Realidade Virtual é uma tecnologia com objetivo de criar uma interface imersiva entre um software gráfico e seu usuário. Para que a imersão ocorra é utilizada uma junção entre técnicas computacionais, como perspectiva e sobreposição de imagens, aliadas a manipulação do ambiente pelo usuário, de maneira a tornar o ambiente sintético o mais próximo da realidade.

Dentre as aplicações destacam-se o uso para a indústria aeroespacial na construção de simuladores de voo, na indústria automobilística para construção de simuladores utilizados por pilotos de corrida, em jogos para torná-los mais realistas, na ciência é utilizada na visualização e manipulação de modelos de uma infinidade de tipos de dinâmica molecular, medicina, habilitação de motoristas e neste trabalho, para imersão em simulações de dinâmica de galáxias.

Existem diversas formas de imersão, as duas mais comuns são: a utilização de múltiplos projetores com óculos passivos, como os utilizados nos filmes de cinema; a utilização de HMD (*Head Mounted Display*), vistos nas Figuras 22 e 23, que são os óculos de realidade virtual compostos por um conjunto de lentes biconvexas e uma tela ligada a um sistema eletrônico. A segunda opção foi adotada neste trabalho.

Figura 22 – Ilustração do uso de HMD.



Fonte: <veditto.com/what-is-virtual-reality-vr-how-does-vr-work-and-the-best-vr-gadgets>.



Figura 23 – Exemplo de utilização de HMD comercial.

Fonte: O Autor.

4 RESULTADOS

A partir deste ponto, com as revisões feitas no Capítulo 2 - Conceitos de Física e no Capítulo 3 - Conceitos de Computação Gráfica, o leitor possui conhecimento necessário para compreender os resultados aqui tratados. Na Seção 4.1 tratamos do desenvolvimento de um ambiente imersivo utilizado neste trabalho, composto pelo código GraVity e seu HMD GraVityVR. Na Seção 4.2 apresentamos todo o processo de criação dos modelos de galáxia, a montagem do grid com 72 simulações de colisão. Na seção 4.3 temos uma análise gráfica qualitativa das simulações, comparando as saídas de softwares de visualização com as do código GraVity e uma análise de sua performance das simulações.

4.1 O AMBIENTE IMERSIVO

No início deste mestrado tínhamos um objetivo claro: desenvolver um novo código de visualização de modelos provenientes do resultado de simulações de N-Corpos gravitacional. Este código deveria ser utilizado em conjunto com HMD comercial que possibilitaria ao usuário a sensação imersiva no ambiente virtual. Os protótipos de RV utilizados neste trabalho, conforme a Tabela 3, utilizaram diferentes tecnologias. Neste Capítulo daremos ênfase apenas ao Projeto 4. O acompanhamento das versões anteriores é abordado nos Apêndices A, B e C.

GraVityVR	HMD	Início	Estágio
1	Oculus Rift	Agosto de 2015	Descontinuado
2	OSVR	Junho de 2016	Descontinuado
3	Cardboard	Agosto de 2016	Descontinuado
4	OpenSource	Novembro de 2016	Projeto atual

Tabela 3 – Protótipos de RV e seu estado de desenvolvimento.

Fonte: O Autor.

Por razões alheias ao nosso controle, fomos impedidos de importar dois protótipos de HMD comerciais, tanto *Oculus Rift* como o *OSVR*, por isso, em diferentes estágios de desenvolvimento, tivemos de recomeçar o desenvolvimento utilizando outra plataforma.

4.1.1 O CÓDIGO GRAVITY

O código **GraVity** (*Immersive environment of Gas and Stars dynamics using Virtual Reality*) tem como objetivo visualizar as posições de partículas de uma simulação de N-Corpos. GraVity pode ser utilizado com o HMD construído neste trabalho ou em Desktop. GraVity permite criar um ambiente que será utilizado em realidade virtual, dando liberdade de visualizar e realçar diferentes componentes dos *snapshots*.

O código tem objetivo semelhante ao do GLnemo²¹ (LAMBERT, 2012) e do Gadget File Viewer², visualizar modelos tridimensionais de simulações. GraVity oferece a vantagem de possibilitar que o usuário se movimente dentro da simulação, alterando a sua posição na cena com três graus de liberdade em tempo de execução.

4.1.1.1 O DESENVOLVIMENTO

O Sistema Operacional (SO) adotado no desenvolvimento é o Linux, pelo fato de possuir código fonte aberto e possuir uma forte comunidade de desenvolvedores, o que permite que as bibliotecas tirem maior proveito de sua arquitetura e recursos do kernel. Outro fator importante é a variedade de compiladores e bibliotecas. Por fim, o Linux é um SO que permite maior personalização que outros sistemas pagos e de código fonte fechado.

Na escolha da linguagem de desenvolvimento foi priorizada a performance ao invés de produtividade, por isso, escolhemos a linguagem C++, cujos executáveis são compilados, em oposição a linguagens interpretadas e de desenvolvimento mais rápido como Python.

A biblioteca gráfica utilizada é o OpenGL, que, por ser multiplataforma pode permitir no futuro a portabilidade de versões para outros SO. Outro motivo é que o OpenGL é uma biblioteca madura e compatível com qualquer dispositivo gráfico de hoje. A fim de agilizar o desenvolvimento, foi adotada a biblioteca SDL2 ³ que funciona como uma camada sobre o OpenGL. O SDL2 tem o objetivo de simplificar o acesso a recursos de entrada e saída E/S entre o SO e o OpenGL, servindo como interface para dispositivos como teclado, mouse e joystick, além de fornecer acesso a recursos gráficos do SO como fontes True Type e acesso a imagens em diferentes formatos.

O motor gráfico utiliza recursos da GPU com múltiplas threads para acelerar a renderização da cena, que é construída utilizando planos de corte e transparência para evitar que partículas fiquem saturadas à medida que nos aproximamos ou nos afastamos

¹ Site de GLnemo2 < https://projets.lam.fr/projects/glnemo2>.

² Site do Gadget File Viewer < http://astro.dur.ac.uk/~jch/gadgetviewer/index.html>.

³ SDL2 - *Simple DirectMedia Layer 2* é uma biblioteca multiplataforma (Windows, Linux e Mac), multimídia, escrito em Linguagem C e distribuído com código aberto sobre a licença *zlib*.

das mesmas. Todas as partículas têm o mesmo tamanho e suas cores variam apenas com o tipo da partícula (estrelas, gás, etc).

4.1.1.2 INTERFACE COM USUÁRIO

GraVity utiliza uma interface simplista onde o usuário passa via linha de comando uma lista de *snapshots* e quais componentes deseja visualizar e estes são carregados. Um exemplo de parâmetros é dado a seguir:

eduschiavo@linux-b7q2:~> gravity --help

gravity version 0.2.13

in=	single snapshot name or snapshot list file
select=	disk,gas,halo,bulge,bh
vr=	on enable split scren, off desktop mode (default)

gravity in=snapshot_001 select=disk,gas vr=off

this will load only disk and gas particles from snapshot_001 and show particles in desktop mode

gravity in=snapshot_list select=disk,gas vr=on

this will load only disk and gas particles from snapshot list file and show particles in vr mode

Nas Figuras 24 e 25 temos exemplos de visualização utilizando modo Desktop e modo VR. O modo VR exibe a mesma imagem mas com ponto de vista ligeiramente deslocado, de forma a tentar reproduzir o ângulo de visada e o campo de visão de cada olho⁴.

⁴ Atualmente esse percentual de imagens repetidas é definido em arquivo de configuração e permite alteração em relação à sobreposição de imagens.

Figura 24 – Exemplo de visualização de galáxias em processo de colisão em modo Desktop.

Fonte: O Autor.

Figura 25 – Exemplo de visualização com tela dividida de galáxias em processo de colisão em modo RV.



Fonte: O Autor.

No motor gráfico do código GraVity todas as partículas são exibidas como esferas de raio unitário no SRO⁵, seu tamanho e a intensidade da cor variam à medida que se aumenta ou diminui a distância em função de um kernel que define os planos de corte e opacidade das partículas.

O kernel é essencial para evitar que partículas saturem a imagem gerada, escondendo alguma estrutura, como pode ser visto nas Figuras 35 até 38 de barra nos modelos na seção 4.3.1. Para o caso das partículas de gás, usamos uma intensidade *Alpha* que resulta no grau de transparência justamente para que sejam sobrepostas criando o efeito de nuvem.

A navegação no modelo é feita utilizando um Joystick de *Nintendo Wii*, onde a posição no modelo é controlada pelo direcional digital, no plano xz, e a posição para onde a câmera aponta, por sua vez, é controlada pelo analógico do Joystick girando ao redor

 $[\]overline{}^{5}$ Para mais informações sobre sistemas de coordenadas veja a Seção 3.2.

dos eixos x e y, ou ainda através de um giroscópio no HMD. O avanço ou retrocesso do modelo é feito através dos botões A e B do Joystick, ou ainda da barra de espaço para reprodução contínua.

4.1.2 CONSTRUÇÃO DO HMD OPENSOURCE

A visualização de simulações geralmente é feita através da saída padrão, o monitor. O usuário define seu ponto de vista, quais são as componentes presentes no modelo que necessita visualizar, como: gás, halo, bojo, disco e, então, reproduz a simulação de forma análoga a um filme, avançando, retrocedendo, ou mesmo alterando a velocidade de reprodução para ver atentamente algum detalhe.

Essa visualização tradicional é funcional, porém a reprodução em RV permitirá uma melhor interpretação da imagem exibida, devido ao maior grau de liberdade. Por isso o objetivo deste trabalho é criar um ambiente de visualização computacional que proporcione uma experiência imersiva em modelos resultantes de simulações de sistemas planetários, aglomerados de estrelas, galáxias e até grandes volumes do universo em simulações de formação e evolução de grandes estruturas cósmicas. O uso da imersão em RV poderá auxiliar na análise morfológica destas estruturas. A impressão do corpo do protótipo pode ser vista na Figura 26, os modelos tridimensionais e seus pares impressos podem ser vistos na Figura 27.



Figura 26 – Impressão do corpo do protótipo baseado no Zendome.

Fonte: O Autor.

Figura 27 – Os modelos 3D e impressos do HMD: a) o separador do corpo de lentes; b) o suporte da lente; c) o conjunto de suporte das lentes; d) o suporte do LCD; e) os correspondentes impressos.



Fonte dos itens **a**), **b**), **c**), **d**): Carsten Fulland <<u>https://www.thingiverse.com/thing</u>: 43655>; Item **e**): O Autor.

Este protótipo foi construído a partir de uma versão modificada do OpenSource "ZENDOME Stereoscopic 3D NEXUS 7⁶. A principal modificação tem o objetivo de substituir o smartfone por uma tela de LCD.

O conjunto óptico eletrônico desenvolvido para este projeto (HMD), recebeu o nome de GraVityVR. O corpo do HMD foi acoplado a uma tela LCD de 7 polegadas com resolução HD (1280x720) que é controlada pelo controlador VS Display Technology model VS-TY2662-V1⁷, que podem ser vistos na Figura 28.

Durante o desenvolvimento dos protótipos foi detectado que a resolução exibida no HMD era 1080i, ou seja, 1920x1080 interpolada. Resoluções interpoladas são muito utilizadas como uma forma de enviar imagens mais detalhadas em menores larguras de banda, ou seja, trocava blocos de 540 linhas alternadamente, em vez da imagem inteira. Essa interpolação poderia causar desconforto ao usuário e, por este motivo, o módulo

⁶ O modelo do ZENDOME pode ser obtido em <https://www.thingiverse.com/thing:43655>.

 $^{^{7} {\}rm \ Mais\ informações\ sobre\ a\ controladora\ < http://www.vslcd.com/Specification/VS-TY2662-V1.pdf>.}$



Figura 28 – Protótipo desmontado, sem o conjunto de lentes.

Fonte: O Autor.

de kernel que controla a impressão dos pontos na tela foi reescrito de forma a trabalhar com resoluções progressivas e, além disso, a resolução foi alterada para 720p ou 1280x720 progressivo. O resultante do protótipo GraVityVR exibindo uma simulação em 720p pode ser visto na Figura 29.

Figura 29 – Tela de LCD de 7 polegadas, exibindo um frame de simulação com um par de galáxias em interação.



Fonte: O Autor.

Nas Figuras 30 e 31 temos um teste em bancada do protótipo controlado por uma Raspberry Pi 3, rodando uma distribuição OpenSUSE Tumbleweed para arquitetura aarch64. Apesar da Raspberry Pi 3 ser um mini computador quadcore, a falta de memória e de uma GPU dedicada limitaram de forma significativa a performance, assim, este equipamento foi utilizado apenas para testes.





Fonte: O Autor.

Figura 31 – Imagem sendo exibida nos testes iniciais em bancada com o protótipo.

Fonte: O Autor.

4.2 O GRID DE SIMULAÇÕES

A proposta inicial destas simulações foi estender o trabalho proposto por Beghetto Junior (2015), ampliando o objetivo inicial de estudar formação de anel polar, adicionando também a evolução morfológica dos modelos em relação à sua classificação. Para isso foram incluídas novas inclinações não testadas pelo autor no trabalho original.

Este trabalho foi desenvolvido com o uso de recursos do Cluster Hipercubo (Projetos FINEP 01.10.0661-00, FAPESP 2011/13250-0, FAPESP 2013/17247-9 e 2014/10489-0) do Instituto de Pesquisa & Desenvolvimento (IP&D) da Universidade do Vale do Paraíba (UNIVAP).

4.2.1 MODELOS

Para criarmos os modelos de galáxias utilizados para montar o grid, utilizamos o código *MakeNewDisk* abordado na seção 2.3.1.1 que pode criar uma variedade de opções de modelos de galáxias de forma parametrizada. As formas como *MakeNewDisk* realiza as distribuições de partículas ao longo do Halo, Bojo, Disco de Estrelas e Gás são abordadas nas Seções 2.3.1.1.1, 2.3.1.1.2, 2.3.1.1.3 respectivamente. Os parâmetros utilizados para criação dos modelos foram gentilmente cedidos pelo autor Dino Beghetto Junior e são resumidos a seguir.

- Modelo A (galáxia hospedeira) Galáxia discoidal, tomando como base os parâmetros gerais da Via-Láctea. Este modelo foi concebido originalmente para se tornar a galáxia hospedeira de uma eventual PRG.
- Modelo B (galáxia doadora) Galáxia com 50% da massa do modelo A. Disco de gás mais estendido, o que pode facilitar a captura desse material pela hospedeira para formar o anel do sistema.
- Modelo C (galáxia doadora) Similar ao modelo B, mas com 20% da massa do modelo A.
- Modelo D (galáxia doadora) Similar ao modelo B, mas com 10% da massa do modelo A.
- Modelo E (galáxia doadora) Similar ao modelo D, mas com 40% a mais na fração de gás do modelo D.

Na Figura 32 temos as condições iniciais do Modelo A, plotados por diferentes ferramentas.

Figura 32 – Plotagem do disco de estrelas e gás do modelo A. a) Gadget Viewer 1.0.7; b) GLnemo2 1.10.0; c) GraVity 0.2.12.



Fonte: O Autor.

A nomenclatura das simulações segue a convenção sim-M1x α M2y β -velV-qQ, onde M1x α é o modelo principal rotacionado em torno de seu eixo x em α graus, M2y β é o modelo secundário rotacionado em torno de seu eixo y em β graus, e velV é a velocidade inicial, calculada a partir de sua velocidade de escape, onde vel0 é a velocidade de escape da órbita da galáxia secundária com relação à principal. Por exemplo, vel10 é uma simulação em que a galáxia secundária é lançada com velocidade 10% superior a velocidade de escape e vel20 é acrescido de 20%. A distância de pericentro q=8, 12 kpc.

Adotamos M1 (principal) como o modelo **A** rotacionado em $\alpha = 90$ graus com respeito ao eixo x para todas as simulações. Para M2 escolhemos utilizar apenas os modelos **C**, **D** e **E**, pois o modelo **B** não trouxe os resultados desejados no trabalho original. Para inclinação dos modelos M2 adotamos $\beta = 0, 30, 60, 90$ graus com respeito ao eixo y. As velocidades de escape foram adotadas velV= 0, 10, 20. Para distância de pericentro q da órbitas das galáxias foram escolhidos os valores de 8 e 12 kpc. A evolução temporal adotada para estas simulações foi um período de 3 bilhões de anos. Um exemplo com arquivo de condições iniciais é visto na Figura 33.

As combinações de todos os parâmetros possíveis, na Figura 34, resultaram em um grid com total de 72 simulações.

Figura 33 – Plotagem com estrelas e gás do arquivo de condições iniciais do modelo sim-Ax90Cy0-vel0-q8. a) Gadget Viewer 1.0.7; b) GLnemo2 1.10.0; c) GraVity 0.2.12.



Fonte: O Autor.

Figura 34 – Diagrama com todas as combinações de simulações possíveis.



Fonte: O Autor.

As órbitas das galáxias não são keplerianas. Para seu cálculo de posição inicial e velocidades foi utilizado o código *PotDinFric*, que foi escrito em linguagem C pelo professor Rodrigues (1999) com colaborações de seus alunos ao longo dos anos. O objetivo do código é obter as coordenadas iniciais de velocidade e posição da galáxia secundária com relação à principal. E seguindo o trabalho de (BEGHETTO JUNIOR, 2015), a velocidade no início
de cada simulação foi deduzida negligenciando-se os efeitos causados por fricção dinâmica, por meio de integração temporal reversa.

Para criarmos os arquivos de condições iniciais utilizados nas simulações, utilizamos recursos do UNSIO - Universal N-body Snapshot Input Output, desenvolvido por Lambert (2014). O $UNSIO^8$ é uma API para intercâmbio de diferentes formatos de snapshots desenvolvida sobre licença CeCILL2⁹.

Para rotacionarmos os modelos utilizamos o código uns_rotate , que foi desenvolvido pelo professor Irapuan Rodrigues e por Beghetto Junior (2015), conforme o exemplo para rotacionar o modelo A em 90 graus em respeito ao eixo x, como segue:

```
# README-EDU-uns rotate.txt
```

```
# data: 15-09-2016
```

Dados da simulação

```
# Modelos de entrada:
```

in:GADGET-IC2A

```
# out:GADGET-IC2A-xrot90.0
```

Gerado com

```
uns_rotate in=GADGET-IC2A \
    out=GADGET-IC2A-xrot90.0 \
    select=all \
    xrot=90 \
    type=gadget2
```

Para juntarmos os dois modelos no arquivo de condições iniciais utilizamos o uns_stack , conforme o exemplo no qual geramos o arquivo de condições da simulação sim-Ax90Cy0-vel0 para q = 8kpc, dado por:

- # README-EDU-parametrosusados.txt
- # data: 15-09-2016
- # Dados da simulação
- # Modelos de entrada:
- # in1:GADGET-IC2C-yrot0.0
- # in2:GADGET-IC2A-xrot90.0
- # nome do job edu_sim-Ax90Cy0-vel0

⁸ O UNSIO pode ser obtido em <https://projets.lam.fr/projects/unsio>.

⁹ CeCILL2 - é uma licença de software desenvolvida pelas agências francesas Commissariat à l'Énergie Atomique, Centre national de la recherche scientifique e Institut national de recherche en informatique et en automatique o nome é formado a partir do acrônimo do nome das instituições Cea Cnrs Inria Logiciel Libre V2.

```
# Gerado com
```

```
uns_stack in1=../GADGET-IC2C-yrot0.0 \
    in2=../GADGET-IC2A-xrot90.0 \
    out=GADGET-IC2-Ax90Cy0-vel0 \
    deltar=253.231,-78.8249,0.0 \
    deltav=-185.364,73.8154,0.0 \
    zerocm=true
```

As simulações foram feitas utilizando o código Gadget GA laxies with Dark matter and Gas int Erac T (SPRINGEL, 2005) na versão 2.0.7, distribuído sob a licença GPL¹⁰. Cada simulação foi executada em 20 processos MPI, o script de job padrão usado no torque é dado a seguir:

```
#!/bin/bash
# Arquivo: run torque.sh
# Job name
#PBS -N edu sim-Ax90Cy0-vel0
    Queue:
#
#PBS -q dinamica
#
   Mandar e-mail em caso de a (abort), b (begin), and e (end)
#PBS -m abe
#PBS -M eduschiavo@gmail.com
    Junta STDOUT e STDERR no mesmo arquivo
#
#PBS -j oe
#PBS -l nodes=1:ppn=20
# Entra no diretório onde voce submeteu o job
cd $PBS O WORKDIR
# Lista os nodos usados
pbsdsh uname -n
# Carrego os módulos necessários
module load shared gcc openmpi/gcc
# Imprime a data e hora do início do job
date
# Executa a simulação usando MPI
time mpirun -machinefile $PBS NODEFILE -np 20 \
     /home/eduschiavo/simulation/bin/Gadget2 OPENMPI GCC GADGET-PARAMETERS
# Imprime a data e hora do fim do job
date
```

¹⁰ GPL - General Public License é utilizada por projetos de software livre e de código aberto.

4.3 ANÁLISE DOS RESULTADOS

Na Seção 4.3.1 são abordados os resultados comuns a todas as simulações. Na Seção 4.3.2 são abordados os resultados das simulações do tipo AC. Na Seção 4.3.3 são abordados os resultados das simulações do tipo AD e AE. Na Seção 4.3.4 são tratadas as análises de performance das simulações.

4.3.1 RESULTADOS COMUNS A TODAS SIMULAÇÕES

O primeiro passo para a realização de uma simulação de colisão de galáxias é o teste de estabilidade dos modelos isolados. Os parâmetros dos modelos A, C, D e E que foram construídos baseados em parâmetros da Via Láctea. As galáxias dos modelos A e C evoluem para SBb com 0.22 bilhões de anos de evolução temporal, as galáxias D e E em torno de 0.51 bilhões de anos. Importante ressaltar que nesse ponto ambas as galáxias primária e secundária ainda não colidiram, por isso a definição de sua estrutura ocorre independente de distância de pericentro, inclinação relativa ou de colisão.

A título de comparação visualizamos os *snapshots* em 3 programas distintos, GadgedViewer, GLnemo2 e GraVity. Nas Figuras 35 até 38, como visto na Seção 4.2.1, foi feita a plotagem dos modelos desta vez já com barra característica.

Figura 35 – Comparativo do frame das componentes disco de estrelas e gás do Modelo A após 0,21 bilhões de anos: a) Gadget Viewer 1.0.7; b) GLnemo2 1.10.0; c) GraVity 0.2.12.



Fonte: O Autor.

Figura 36 – Comparativo do frame das componentes disco de estrelas e gás do Modelo C após 0,21 bilhões de anos: a) Gadget Viewer 1.0.7; b) GLnemo2 1.10.0; c) GraVity 0.2.12.



Fonte: O Autor.

Figura 37 – Comparativo do frame das componentes disco de estrelas e gás do Modelo D após 0,55 bilhões de anos: a) Gadget Viewer 1.0.7; b) GLnemo2 1.10.0; c) GraVity 0.2.12.



Fonte: O Autor.

Figura 38 – Comparativo do frame das componentes disco de estrelas e gás do Modelo E após 0,55 bilhões de anos: a) Gadget Viewer 1.0.7; b) GLnemo2 1.10.0; c) GraVity 0.2.12.



Fonte: O Autor.

4.3.2 SIMULAÇÕES AC

Com exceção das simulações Ax90Cy90, na Figura 39 em que primária e secundária se distorcem pouco ao longo de sua evolução, em todas as outras simulações do tipo AC, a secundária se deforma abrindo longos e definidos braços de maré, vistos na Figura 40. Ambas perdem matéria para o meio no choque e um braço da secundária se conecta no núcleo da primária que então começa a acretar matéria formando uma ponte de ligação entre as duas. Adotamos como início e fim da colisão os momentos em que os discos se tocam e que se separam. Na tabela 4 são dados os tempos em que se iniciam e terminam o processo de colisão para estas simulações. Na figura 41 temos o processo de colisão das simulações com distância de pericentro de 12 kpc.

Figura 39 – Plotagem das componentes de disco de estrelas da simulação sim-Ax90Cy0vel0-q12 em 1,125 bilhões de anos de evolução: a) vista lateral dos modelos após colisão; a) vista superior após colisão com rotação de 90 graus em direção ao eixo x; c) vista lateral após colisão com rotação de 90 graus em direção ao eixo y.



Fonte: O Autor.

Nas simulações do tipo Vel0, o braço de ligação fornece matéria para a primária

Figura 40 – Plotagem das componentes de disco de estrelas e gás na evolução da secundária para SBc após a colisão: a) sim-Ax90Cy0-vel0-q8 em 1,07 bilhões de anos de evolução; b) sim-Ax90Cy30-vel10-q8 em 0,97 bilhões de anos de evolução; d) sim-Ax90Cy60-vel20-q8 em 0,97 bilhões de anos de evolução.



Fonte: O Autor.

Tabela 4 – Período de colisão dos modelos AC em bilhões de anos.

Simulação	Período
sim-Ax90Cy β -Vel0	0,92 - 1,02
sim-Ax90Cy β -Vel10	0,87 - 0,97
sim-Ax90Cy β -Vel20	0,82 - 0,92

Fonte: O Autor.

Figura 41 – Plotagem das componentes de disco de estrelas e gás durante o processo de colisão: a) sim-Ax90Cy0-vel0-q12 em 0,99bilhões de anos de evolução;
b) sim-Ax90Cy30-vel10-q12 em 0,93 bilhões de anos de evolução; d) sim-Ax90Cy60-vel20-q12 em 0,89 bilhões de anos de evolução.





próximo ao núcleo de seu disco, que logo em seguida começa a ejetar matéria pelo lado oposto. Neste ponto se inicia a formação de uma estrutura semelhante a uma alça difusa de estrelas e gás que atravessa o núcleo da galáxia até o lado oposto que se encontra a doadora. A inclinação desta estrutura em forma de alça que varia conforme a inclinação relativa dos modelos. Em todas as simulações deste tipo a secundária volta e colide mais uma vez com a primária iniciando o processo de merger.

No caso de simulações do tipo Vel10 e Vel20, o braço de ligação também fornece matéria para a primária próximo ao núcleo de seu disco, que logo em seguida começa a ejetar matéria pelo lado oposto. Neste ponto se inicia a formação de uma estrutura difusa de estrelas e gás do lado oposto a doadora. Nestas simulações não chega a ocorrer a segunda colisão¹¹.

4.3.3 SIMULAÇÕES AD E AE

As simulações do tipo AD e AE apresentam resultados estruturalmente muito similares, por isso discutiremos apenas os resultados das AD. De forma semelhante as simulações do tipo AC, as secundárias se deformam criando longos braços de maré, vistas na Figura 42, ambas perdem matéria para o meio durante a colisão. Um braço de maré da secundária se conecta no núcleo da primária que então começa a acretar matéria formando uma ponte de ligação entre as duas. Durante o processo de acreção de matéria, essa ponte de ligação se desloca da parte superior do disco para borda do mesmo, e por fim acaba ligada à parte inferior.

Figura 42 – Plotagem das componentes de disco de estrelas e gás na evolução da secundária para SBc após a colisão: a) sim-Ax90Dy0-vel0-q8 em 1,01 bilhões de anos de evolução; b) sim-Ax90Dy30-vel10-q8 em 0,96 bilhões de anos de evolução; d) sim-Ax90Dy60-vel20-q8 em 0,96 bilhões de anos de evolução.



Fonte: O Autor.

¹¹ A segunda colisão e o processo de merge poderia ocorrer caso fosse aumentado o tempo de simulação, como o objetivo era apenas verificar a formação de estruturas, não foi adicionado mais tempo nas simulações.

Na tabela 5 são dados os tempos em que se iniciam e terminam o processo de colisão para estas simulações. Na figura 43 temos o processo de colisão das simulações com distância de pericentro de 8 kpc.

Tabela 5 – Período de colisão dos modelos AD em bilhões de anos.

$sim-Ax90Cy\beta-Vel0$	0,94 - 1,04
sim-Ax90Cy β -Vel10	0,88 - 0,98
sim-Ax90Cy β -Vel20	0,83 - 0,93

Figura 43 – Plotagem das componentes de disco de estrelas e gás durante o processo de colisão: a) sim-Ax90Dy0-vel0-q8 em 1,09 bilhões de anos de evolução; b) sim-Ax90Dy30-vel10-q8 em 1,04 bilhões de anos de evolução; d) sim-Ax90Dy60-vel20-q8 em 1,04 bilhões de anos de evolução.





Nas simulações do tipo Vel0, o braço de ligação fornece matéria para a primária próximo ao núcleo de seu disco, que logo em seguida começa a ejetar matéria pelo lado oposto. Neste ponto se inicia a formação de uma estrutura alça difusa de estrelas e gás que atravessa o núcleo da galáxia até o lado oposto ao que se encontra a doadora. A inclinação desta estrutura varia conforme a inclinação relativa dos modelos.

Para os casos de simulações do tipo Vel10 e Vel20, o braço de ligação também fornece matéria para a primária próximo ao núcleo de seu disco, que logo em seguida começa a ejetar matéria pelo lado oposto. Neste ponto se inicia a formação de uma estrutura difusa de estrelas e gás do lado oposto à doadora.

4.3.4 ANÁLISE DE PERFORMANCE DAS SIMULAÇÕES

As simulações foram rodadas no Cluster Hipercubo do Instituto de Pesquisa & Desenvolvimento (IP&D) da Universidade do Vale do Paraíba (UNIVAP), utilizando 4 nodos SuperMicro twin (node013 a node016), onde cada nodo possui 2 processadores Intel(R) Xeon(R) CPU E5-2670 v2 com 10 núcleos e 20 threads, 25MB de cache e frequência de 2,5 GHz, 80 GB de memória DDR3, redes gigabit (IPMI/lan) e Infiniband (fast) de 40Gbps.

O *Wall time* ou tempo de processamento nas tabelas 6 e 7 é o tempo de relógio entre o início e o fim das simulações, e o tempo de CPU nas tabelas 8 e 9 é a soma do tempo de execução de cada núcleo utilizado no processamento das simulações. Os tempos possuem o seguinte formato [*HH:MM:SS,SSS]:

- *HH: Tempo acumulado em horas com 2 ou mais dígitos, não fracionados em múltiplos de 24;
- MM: minutos com dois dígitos;
- SS,SSS: segundos com dois dígitos e fração de segundo com 3 dígitos em milissegundos.

Tabela 6 – Tempo de processamento (*Wall time*) em [*HH:MM:SS,SSS] das simulações com distância de pericentro de 8kpc.

yβ	Vel	Ax90C	Ax90D	Ax90E
	Vel-0	58:17:39,769	57:22:21,894	58:42:35,927
y0	Vel-10	61:11:00,656	60:59:24,684	64:42:54,437
	Vel-20	62:12:20,719	63:57:53,849	65:52:23,148
	Vel-0	61:47:20,980	63:51:00,203	62:05:47,666
y30	Vel-10	66:59:09,930	65:55:54,892	66:01:39,096
	Vel-20	67:13:10,622	63:02:01,331	64:44:21,031
	Vel-0	60:13:51,935	57:06:49,617	61:03:49,121
y60	Vel-10	62:43:29,007	64:02:56,901	63:45:58,243
	Vel-20	64:02:39,959	61:49:29,657	60:05:29,422
	Vel-0	59:39:49,879	61:29:46,876	63:15:58,602
y90	Vel-10	61:51:39,474	65:27:49,449	64:33:47,098
	Vel-20	64:59:48,876	63:38:12,465	65:50:35,008

Fonte: O Autor.

A execução das simulações com distância de pericentro de 8 kpc levaram entre 57:06:49,617 e 67:13:10,622 (2,3 e 2,8 dias).

Vel	Ax90C	Ax90D	Ax90E
Vel-0	64:35:12,471	65:49:34,481	62:42:51,914
Vel-10	68:58:50,057	64:58:57,494	67:34:26,172
Vel-20	67:15:28,532	66:32:01,070	67:45:25,388
Vel-0	62:32:35,701	60:26:16,638	62:04:53,023
Vel-10	66:15:19,487	63:04:25,318	64:00:15,120
Vel-20	64:39:59,779	65:00:39,559	66:53:14,952
Vel-0	60:41:58,890	61:46:07,000	61:34:18,085
Vel-10	66:55:22,788	65:25:29,078	64:32:36,894
Vel-20	67:04:44,729	68:13:08,795	66:37:07,057
Vel-0	63:58:56,782	67:33:09,363	68:07:24,666
Vel-10	71:36:00,497	71:00:05,722	73:22:40,603
Vel-20	72:33:39,288	67:36:41,225	61:51:05,578
	Vel- Vel-10 Vel-20 Vel-0 Vel-10 Vel-20 Vel-0 Vel-20 Vel-20 Vel-10 Vel-20	VelAx90CVel-064:35:12,471Vel-1068:58:50,057Vel-2067:15:28,532Vel-062:32:35,701Vel-066:15:19,487Vel-2064:39:59,779Vel-060:41:58,890Vel-1066:55:22,788Vel-2067:04:44,729Vel-063:58:56,782Vel-1071:36:00,497Vel-2072:33:39,288	VelAx90CAx90DVel-064:35:12,47165:49:34,481Vel-1068:58:50,05764:58:57,494Vel-2067:15:28,53266:32:01,070Vel-062:32:35,70160:26:16,638Vel-1066:15:19,48763:04:25,318Vel-2064:39:59,77965:00:39,559Vel-060:41:58,89061:46:07,000Vel-1066:55:22,78865:25:29,078Vel-2067:04:44,72968:13:08,795Vel-063:58:56,78267:33:09,363Vel-1071:36:00,49771:00:05,722Vel-2072:33:39,28867:36:41,225

Tabela 7 – Tempo de processamento (*Wall time*) em [*HH:MM:SS,SSS] das simulações com distância de pericentro de 12kpc.

Fonte: O Autor.

Já a execução das simulações com maior distância de pericentro, de 12 kpc, levaram entre 60:26:16,638 e 73:22:40,603 (2,5 e 3,1 dias). Apesar de ser prático utilizarmos o tempo de processamento, sempre que se necessita de tempo de processamento em algum cluster, esse tempo é pedido em tempo de CPU, pois contabiliza a soma do tempo de uso de cada núcleo utilizado, ou seja, para este caso aproximadamente 20 vezes o *walltime*, já que o código foi executado em 20 cores, por isso adotamos um detalhamento destes dados em tempo de CPU.

yХ	Vel	Ax90C	Ax90D	Ax90E
	Vel-0	1165:35:10,359	1146:55:53,660	1173:05:09,411
y0	Vel-10	1223:00:05,386	1219:20:45,520	1293:57:40,712
	Vel-20	1243:22:18,272	1278:47:10,854	1316:37:54,553
	Vel-0	1235:18:41,648	1276:09:35,023	1241:40:20,047
y30	Vel-10	1339:26:01,713	1317:51:08,100	1319:45:31,835
	Vel-20	1344:10:38,740	1259:51:56,826	1294:30:17,593
	Vel-0	1204:16:52,082	1141:49:16,233	1221:06:21,085
y60	Vel-10	1253:53:27,981	1280:21:51,428	1274:29:38,124
	Vel-20	1279:56:44,799	1235:43:40,630	1201:24:39,668
	Vel-0	1192:55:27,207	1229:20:11,229	1264:28:09,066
y90	Vel-10	1236:39:19,022	1308:55:21,498	1290:54:36,176
	Vel-20	1298:55:29,852	1272:25:59,916	1316:07:59,597

Tabela 8 – Tempo de CPU das simulações com distância de pericentro de 8kpc.

yХ	Vel	Ax90C	Ax90D	Ax90E
	Vel-0	1291:18:12,665	1315:52:02,338	1253:52:19,761
y0	Vel-10	1379:29:45,422	1299:23:47,312	1350:54:12,832
	Vel-20	1344:36:08,527	1330:18:31,899	1354:11:35,910
	Vel-0	1249:50:47,606	1208:11:27,302	1241:19:13,931
y30	Vel-10	1325:00:31,059	1260:47:39,089	1279:48:52,229
	Vel-20	1292:54:39,255	1298:58:32,840	1336:28:50,663
	Vel-0	1213:27:44,538	1234:39:38,839	1231:11:52,678
y60	Vel-10	1338:21:19,267	1308:02:10,728	1290:11:45,972
	Vel-20	1341:11:11,893	1363:54:27,829	1331:46:25,739
	Vel-0	1278:48:17,764	1350:40:44,079	1361:28:10,094
y90	Vel-10	1431:20:58,215	1419:08:14,019	1466:58:40,914
	Vel-20	1450:14:32,948	1351:07:15,171	1235:59:50,524

Tabela 9 – Tempo de CPU das simulações com distância de pericentro de 12kpc.

Fonte: O Autor.

No histograma apresentado Figura 44 temos a distribuição do tempo das simulações em relação ao intervalo com detalhe na distância de pericentro dos modelos e na Figura 45 apresentamos o detalhamento da distribuição de tempo das 72 simulações.



Figura 44 – Histograma de tempo de CPU.

Fonte: O Autor.

Fonte: O Autor.





Este grid de simulações consumiu um tempo total (*walltime*) de 4636:22:06,619 e um tempo de CPU total de 92605:01:57,696. Por fim, com base nos resultados e analisando a disposição dos dados das Figuras e tabelas de tempos, pode-se afirmar que os resultados de performance obtidos corroboram com o que foi esperado para este grid.

Analisando os 3 tipos de modelos colididos AC, AD e AE, mas utilizando as mesmas distâncias de pericentro, é possível notar que os modelos que consomem menos tempo de CPU são os AD, seguidos de AC e por fim AE. A explicação para esse resultado se dá pelo fato de os modelos AD terem menos partículas de estrela no disco que os do modelo AC, ou seja, com menos partículas interagindo o esforço para cálculo de forças é menor que nos outros. Por fim os modelos AE possuem a mesma quantidade de partículas de disco que os AD, mas possuem uma maior fração de gás o que faz com que estes sejam os modelos com maior tempo de CPU, pois sabemos que os cálculo de força gravitacional consome menos tempo de CPU que a interação do gás, e este modelo possui 40% mais gás que o modelo AD.

5 CONSIDERAÇÕES FINAIS

No que tange a revisão bibliográfica necessária para o estudo de Dinâmica de Sistemas Estelares, este trabalho pode servir como um guia para futuros alunos que se adentrarem na área de simulação, especificamente nas simulações de N-Corpos gravitacional, pois revê desde os conceitos mais básicos de órbitas, gravitação, distribuição de partículas em modelos numéricos, algoritmos de cálculo de forças e integradores de tempo. Um artigo baseado no Capítulo 2 - Conceitos de Física está em andamento, com o objetivo de ser uma revisão deste processo de simulação.

Quanto à extensão do estudo de formação de anel polar (PRG) em simulações de colisão de galáxias, apesar de não ter encontrado formação de anel polar, serviu como no trabalho de Beghetto Junior (2015) para restringir um pouco mais os parâmetros necessários na formação de anel polar no cenário de acresção.

Nas discussões de Beghetto Junior (2015), o autor propõe que galáxias doadoras (secundárias) com menos de 50% de massa das hospedeiras (primárias) com distâncias de pericentro menores de 20 kpc poderiam ser o ideal, mas o encontrado não colaborou com o proposto. Nos resultados vemos claramente que quando a diferença de massa é maior que 50%, como ocorreu com os modelos AD e AE, foi justamente quando obtivemos os piores resultados, e tivemos poucas estruturas em forma de alça ou nenhum semelhante a anel, o que nos permite atualizar a tendência de formação de PRG para diferenças menores que 50%. Nos modelos AC obtivemos resultados de estrutura semelhante a uma alça ligada ao disco da principal, mas não obtivemos anel bem definido.

Um artigo já está em andamento sobre formação de PRG contendo tanto os dados de Beghetto Junior (2015) como os deste trabalho. Novas simulações agora abordando diferenças menores de massa estão sendo discutidas e um novo grid de simulações será feito em busca de melhores condições. Além disso o próximo trabalho será feito com formação estelar numa versão que não é pública do código GADGET-3.

Já para o viés computacional, este trabalho pode servir como referência para revisão dos conceitos relacionados a Computação Gráfica, Simulações em ambiente HPC e ainda análise de performance de simulações.

Em relação à análise morfológica com o uso do novo código GraVity, este permitiu a visualização da formação de barra nas galáxias durante as evoluções temporais, que posteriormente foram confirmados alterando as escalas de cor no GLnemo2.

Em relação a perspectivas futura, este código ainda necessita de amadurecimento e seu desenvolvimento vai continuar, um artigo sobre seu desenvolvimento também está em desenvolvimento. Nas próximas versões pretendemos focar principalmente na detecção automatizada de estruturas em modelos, o que facilitaria na análise de resultados de diversas simulações. Também temos como objetivo adicionar suporte a outros formatos de arquivos, pois no momento o software só suporta o formato do GADGET-2 e um formato próprio usado para *debug*. Ainda na questão de compatibilidade, pretendemos adicionar o suporte a visualização de partículas traçadoras ou partículas isoladas em potenciais, para auxiliar a visualização tridimensional por exemplo dos modelos simulados nos trabalhos de órbitas caóticas dos Apêndices D Artigos.

Quanto à novas abordagens no uso de tecnologias gráficas para o estudo de Dinâmica de Sistemas Estelares presumo que é algo que só tende a crescer, pois estamos em um momento em que o acesso a novas tecnologias de RV como os HMD comerciais finalmente está deixando de ser tão restrito e assim poderá ser utilizada de uma forma menos simplória como a tratada neste trabalho, onde tivemos de construir o nosso próprio protótipo de HMD.

REFERÊNCIAS

AARSETH, S. J. Dynamical evolution of clusters of galaxies, I. *Monthly Notices of the Royal Astronomical Society*, v. 126, p. 223, 1963. Citado 2 vezes nas páginas 44 e 47.

AARSETH, S. J. Direct n-body codes. In: _____. The Cambridge N-Body Lectures. Dordrecht: Springer Netherlands, 2008. p. 1–30. ISBN 978-1-4020-8431-7. Disponível em: https://doi.org/10.1007/978-1-4020-8431-7_1. Citado na página 38.

AARSETH, S. J.; HOYLE, F. An assessment of the present state of the N-body problem. Astrophysica Norvegica, v. 9, p. 313, 1964. Citado 2 vezes nas páginas 44 e 47.

AZEVEDO, E.; CONCI, A. Computação Gráfica - Geração de Imagens. [S.l.]: Elsevier, 2003. Citado na página 51.

BAILLARD, A. et al. The EFIGI catalogue of 4458 nearby galaxies with detailed morphology. *Astronomy & Astrophysics*, v. 532, p. A74, ago. 2011. Citado na página 26.

BARNES, J.; HUT, P. A hierarchical O(N log N) force-calculation algorithm. *Nature*, v. 324, p. 446–449, dez. 1986. Citado 2 vezes nas páginas 44 e 45.

BEGHETTO JUNIOR, D. Um estudo numérico sobre formação de galáxias com anel polar. Dissertação (Mestrado) — Instituto de Pesquia & Desenvolvimento, Universidade do Vale do Paraíba, São José dos Campos - SP, Brasil, 2015. Disponível em: <http://biblioteca.univap.br/dados/00001c/00001c4b.pdf>. Acesso em: 3 de janeiro de 2017. Citado 6 vezes nas páginas 28, 43, 69, 71, 72 e 85.

BINNEY, J.; TREMAINE, S. *Galactic Dynamics: Second Edition*. Princeton University Press, 2011. (Princeton Series in Astrophysics). ISBN 9781400828722. Disponível em: https://books.google.com.br/books?id=6mF4CKxlbLsC>. Citado na página 40.

BOURNAUD, F.; COMBES, F. Formation of polar ring galaxies. Astronomy and Astrophysics, v. 401, p. 817–833, abr. 2003. Citado na página 28.

DUBINSKI, J. A parallel tree code. *New Astronomy*, v. 1, p. 133–147, out. 1996. Citado 2 vezes nas páginas 45 e 46.

FINKELMAN, I.; FUNES, J. G.; BROSCH, N. Polar ring galaxies in the Galaxy Zoo. *Monthly Notices of the Royal Astronomical Society*, v. 422, p. 2386–2398, maio 2012. Citado na página 28.

HERNQUIST, L. An analytical model for spherical galaxies and bulges. *Astrophysical Journal*, v. 356, p. 359–364, jun. 1990. Citado 2 vezes nas páginas 41 e 42.

HERNQUIST, L.; KATZ, N. TREESPH - A unification of SPH with the hierarchical tree method. *Astrophysical Journal Supplement Series*, v. 70, p. 419–446, jun. 1989. Citado na página 47.

HUBBLE, E. P. Extragalactic nebulae. *Astrophysical Journal*, v. 64, dez. 1926. Citado 2 vezes nas páginas 9 e 24.

HUBBLE, E. P. Realm of the Nebulae. [S.l.: s.n.], 1936. Citado na página 24.

JARRETT, T. H. et al. The 2MASS Large Galaxy Atlas. *The Astronomical Journal*, v. 125, p. 525–554, fev. 2003. Citado na página 27.

KEPLER, J. Astronomia nova ..., seu physica coelestis, tradita commentariis de motibus stellae martis. [S.l.: s.n.], 1609. Citado 4 vezes nas páginas 30, 31, 32 e 33.

KEPLER, J. Ioannis Keppleri harmonices mundi libri V: quorum primus harmonicus
... quartus metaphysicus, psychologicus et astrologicus geometricus ... secundus
architectonicus ... tertius proprie ... quintus astronomicus & metaphysicus ... : appendix
habet comparationem huius operis cum harmonices Cl. Ptolemaei libro III cumque Roberti
de Fluctibus ... speculationibus harmonicis, operi de macrocosmo & microcosmo insertis.
[S.l.: s.n.], 1619. Citado 3 vezes nas páginas 30, 31 e 34.

KEPLER, S.; SARAIVA, M. Dynamics of Galaxies. [S.l.: s.n.], 2013. ISBN 9788578611873. Citado 2 vezes nas páginas 22 e 35.

LAMBERT, J. C. Glnemo2: An Interactive 3D Visualization Program for N-body Data. In: Ballester, P.; Egret, D.; Lorente, N. P. F. (Ed.). Astronomical Data Analysis Software and Systems XXI. [S.l.: s.n.], 2012. (Astronomical Society of the Pacific Conference Series, v. 461), p. 897. Citado 2 vezes nas páginas 22 e 62.

LAMBERT, J. C. UNSIO: Universal Nbody Snapshot Input Output. In: Manset, N.; Forshay, P. (Ed.). Astronomical Data Analysis Software and Systems XXIII. [S.l.: s.n.], 2014. (Astronomical Society of the Pacific Conference Series, v. 485), p. 429. Citado na página 72.

LEMES, P. F. *Espectroscopia e imageamento da galáxia com anel polar AM2020-504.* Dissertação (Mestrado) — Universidade do Vale do Paraíba, Instituto de Pesquisa e Desenvolvimento, São José dos Campos, 2010. Citado na página 28.

LéPINE, J. A Via Láctea: Nossa Ilha no Universo. [S.l.: s.n.], 2008. ISBN 978-85-314-1056-7. Citado na página 25.

MCCABE, T. J. A Complexity Measure. *IEEE Transactions on Software Engineering*, v. 2, p. 308–320, dez. 1976. Citado na página 91.

MOORE, G. E. Cramming more components onto integrated circuits. *Electronics*, v. 38, n. 8, p. 114–117, 1965. Disponível em: http://www.intel.com/technology/mooreslaw/ index.htm>. Citado na página 21.

NAVARRO, J. F.; FRENK, C. S.; WHITE, S. D. M. The Structure of Cold Dark Matter Halos. *Astrophysical Journal*, v. 462, p. 563, maio 1996. Citado na página 42.

NEWTON, I. Philosophiae Naturalis Principia Mathematica. Auctore Js. Newton. [S.l.: s.n.], 1687. Citado na página 35.

QUINN, T. et al. Time stepping N-body simulations. ArXiv Astrophysics e-prints, out. 1997. Citado 2 vezes nas páginas 48 e 49.

RESHETNIKOV, V. et al. AM 1934-563: a giant spiral polar-ring galaxy in a triplet. *Astronomy and Astrophysics*, v. 446, p. 447–458, fev. 2006. Citado na página 28.

RODRIGUES, I. Interações de sistemas estelares. Tese (Doutorado) — Instituto de Física, Universidade Federal do Rio Grande do Sul, Porto Alegre - RS, Brasil, 1999. Disponível em: http://www1.univap.br/irapuan/Tese/tese-irapa.pdf>. Acesso em: 3 de janeiro de 2017. Citado 3 vezes nas páginas 40, 43 e 71.

SLATER, S. "what is our galaxy's halo made of, and how was it formed? (intermediate)". 2015. Disponível em: http://curious.astro.cornell.edu/about-us/95-the-universe/galaxies/general-questions/511-what-is-our-galaxy-s-halo-made-of-and-how-was-it-formed-intermediate>. Acesso em: 27 de fevereiro de 2018. Citado na página 25.

SPRINGEL, V. The cosmological simulation code GADGET-2. *Monthly Notices of the Royal Astronomical Society*, v. 364, p. 1105–1134, dez. 2005. Citado 3 vezes nas páginas 46, 49 e 73.

SPRINGEL, V.; MATTEO, T. D.; HERNQUIST, L. Modelling feedback from stars and black holes in galaxy mergers. *Monthly Notices of the Royal Astronomical Society*, v. 361, p. 776–794, ago. 2005. Citado 2 vezes nas páginas 41 e 42.

SPRINGEL, V.; WHITE, S. D. M. Tidal tails in cold dark matter cosmologies. *Monthly Notices of the Royal Astronomical Society*, v. 307, p. 162–178, jul. 1999. Citado na página 42.

SPRINGEL, V.; YOSHIDA, N.; WHITE, S. D. M. GADGET: a code for collisionless and gasdynamical cosmological simulations. *New Astronomy*, v. 6, p. 79–117, abr. 2001. Citado 3 vezes nas páginas 40, 43 e 46.

TEAM, N. S. Wmap - content of the universe. 2014. Disponível em: <https://map.gsfc.nasa.gov/universe/uni_matter.html>. Acesso em: 01 de fevereiro de 2018. Citado na página 42.

WHITMORE, B. C. et al. New observations and a photographic atlas of polar-ring galaxies. *Astronomical Journal*, v. 100, p. 1489–1522, nov. 1990. Citado 3 vezes nas páginas 28, 29 e 30.

Apêndices

APÊNDICE A – PROJETO 1 - VR BASEADO NO OCULUS RIFT

Este projeto teve seu esboço no segundo semestre de 2015, com a proposta de ser desenvolvido durante o mestrado. O objetivo inicial era ambicioso: permitir ao usuário a interação com os modelos utilizando um conjunto de HMD e sensores que captassem os movimentos das mãos, que fariam a manipulação nestes modelos.

Apesar de hoje termos fácil acesso a material que permitiria o desenvolvimento em linguagens muito produtivas como Python ou Java, foi preferido o foco na performance, por isso, as linguagens adotadas para o desenvolvimento deste ambiente de imersão foram o C e C++, ambos no padrão ANSI 11.

Como plataforma de computação massivamente paralela, CUDA¹ da Nvidia foi adotada por possuir ampla literatura e grande quantidade de exemplos em sua SDK² disponíveis de forma gratuita. Poderiam ser adotadas outras plataformas, como OpenCL³ ou mesmo Intel MPSS⁴, mas, um maior conhecimento prévio e a disponibilidade de equipamentos com hardware da Nvidia levaram à escolha do padrão.

O protótipo foi inicialmente idealizado para fazer uso de arquitetura modular, ou seja, foi subdividido em três módulos. Com o objetivo de diminuir a **complexidade ciclomática**⁵ (MCCABE, 1976) do software, foi subdividido nos módulos:

- Entrada;
- Processamento;
- Renderização.

A.1 MÓDULO DE ENTRADA

Este módulo tem um requisito principal: ser pequeno e leve, pois o objetivo é possibilitar ser carregado junto ao corpo do usuário, em um bolso ou colete como os de

¹ CUDA - Compute Unified Device Architecture pode ser obtido em <https://developer.nvidia.com/ cuda-downloads>.

² SDK - Software Development Kit.

³ OpenCL - *Open Computing Language* é uma arquitetura que permite o desenvolvimento de códigos que funcionam de forma heterogênea em CPUs e GPUs sem necessidade de grande conhecimento sobre o hardware disponível, desenvolvido pelo Khronos Group, que também mantém o OpenGL.

⁴ Intel MPSS ou Intel Manycore Platform Software Stack é um conjunto de softwares necessários para utilização do host de processamento Intel Xeon Phi.

⁵ Complexidade ciclomática é uma métrica utilizada para determinar a complexidade de um código, contando a quantidade de caminhos de execução independentes que se pode tomar.

cinegrafistas, evitando uma grande quantidade de fios, permitindo maior ergonomia ao usuário. Por este motivo foi escolhido como hardware base a Raspberry Pi 2 modelo B.

A Raspberry Pi 2 vista na Figura 46 é um computador, de tamanho reduzido, construído com processador quadcore que utiliza arquitetura ARM^6 de 32 bits. Os processadores $RISC^7$ são muito utilizados em sistemas embarcados devido ao seu tamanho reduzido e à simplicidade de seus núcleos de processamento, com isso atingindo maior eficiência por ciclo de clock, realizando tarefas mais simples em ciclos mais curtos. Ainda devido a essa arquitetura, possui baixo consumo de energia e dissipa pouco calor, sendo ideal para o uso proposto.

Figura 46 – Raspberry Pi 2 modelo B, utilizada nos Projetos 1 e 2 (em tamanho real).





Como sistema operacional embarcado escolhemos o Linux na sua Distribuição nativa Raspbian⁸. Derivada do Debian, o Raspbian é desenvolvido especificamente para utilização na Raspberry. Novamente a escolha dessa distribuição foi feita com base em experiência prévia em outros projetos, devido a sua estabilidade, disponibilidade de repositórios e software e atualização frequente.

O avanço e retrocesso da reprodução da simulação, bem como a movimentação dentro do ambiente imersivo são feitos utilizando um joystick *"Motion Plus com Nunchuk Wii"*, da Nintendo conforme a Figura 47, que se comunica através de um adaptador

⁶ ARM - (Advanced RISC Machine) é uma família de processadores que utiliza arquitetura RISC desenvolvida e licenciada pela empresa ARM Holdings.

⁷ RISC - (*Reduced Istruction Set Computer*) é uma arquitetura de processadores com conjunto reduzido de instruções.

⁸ O Raspbian pode ser obtido em <https://www.raspberrypi.org/downloads/raspbian/>.

Bluetooth⁹ fazendo uso de um módulo de kernel compilado. Este modelo de Joystick foi escolhido por possuir uma série de sensores de posição em relação a eixos e acelerômetros.



Figura 47 – Motion Plus com Nunchuk Wii, utilizada nos Projetos 1 e 2.

Fonte: <http://bit.ly/2zAB9tY>.

Os dados da posição da câmera (ponto de vista do usuário) e o controle de tempo da simulação são enviados via rede Wifi¹⁰ para o Módulo de Processamento utilizando protocolo de comunicação UDP¹¹, de forma semelhante ao streaming. Esta forma de comunicação foi escolhida por não necessitar de confirmação de recebimento de pacotes o tempo todo, ou seja, continua funcionando mesmo que ocorra uma eventual perda de pacotes durante a transmissão.

A.2 MÓDULO DE PROCESSAMENTO

Este módulo opera utilizando como base o hardware Intel x64 com arquitetura *Multi Core*¹², ou seja, um computador com processador Intel que possui diversos núcleos de processamento. Este módulo é responsável por abrir e processar os diversos arquivos

⁹ Bluetooth é um padrão de technologia sem fio utilizado para troca de dados em curtas distâncias.

¹⁰ Wifi ou Wi-Fi é uma tecnologia sem fio desenvolvida para comunicação de rede baseada no padrão IEEE 802.11. Wi-Fi é uma marca registrada da Wi-Fi Alliance.

 $^{^{11}\,}$ UDP - User Datagram Protocol é um protocolo de comunicação não orientado a conexão.

¹² Multi Core é a arquitetura utilizada na construção de processadores, como o de computadores pessoais, que tem dois ou mais núcleos de processamento (*cores*) no interior de um único processador.

de snapshots, que são os arquivos descritivos com dados de cada componente de uma simulação (estrela, gás, buraco negro, etc) em cada timestep da simulação. No momento, os arquivos de *snapshot* suportados são de formato proprietário.

Apesar de cada arquivo de snapshot conter muitas informações, para reprodução são utilizados apenas: posição em 3 componentes (x, y, z), massa, que é utilizada para definir a intensidade da cor de estrelas na simulação e a densidade do gás, que é utilizada para definir o coeficiente da componente Alpha (transparência) do OpenGL.

Os dados de posicionamento e ponto de vista, bem como controle do tempo de execução da simulação obtidos pelo Módulo de Entrada, são utilizados para gerar a cena que adota um espaço tridimensional. O processamento nesse módulo é massivo e faz uso de múltiplas threads¹³ com diversas linhas de execução, de modo a acelerar o processamento da construção da cena como um todo.

A.3 MÓDULO DE RENDERIZAÇÃO

Este módulo é responsável pelo código que faz a geração e renderização da imagem visualizada pelo usuário e novamente necessita de um processamento massivo, que ocorre em tempo de execução e, portanto, necessita da arquitetura *Many Core*¹⁴ para processar e renderizar as imagens em diversos blocos de threads. O HMD escolhido foi o "Oculus Rift Beta" conforme a Figura 48

No estágio atual de desenvolvimento do nosso sistema, a visualização pode ser feita através da saída padrão, utilizando monitor ou utilizando a visualização estereoscópica com dois projetores fazendo a sobreposição de imagem. Em ambos os casos a renderização é feita utilizando o OpenGL, "Nvidia GameWorks" e "Oculus PC SDK Beta".

A.4 RESULTADOS PRELIMINARES

Mesmo neste estágio inicial de desenvolvimento, foi possível gerar imagens estereoscópicas utilizando o *Oculus PC SDK 0.6.0.1 Beta*¹⁵ e o *Nvidia GameWorks*¹⁶, ambos disponíveis apenas para ambiente Windows (durante o então período de desenvolvimento).

 $^{^{13}\,}$ Thread é uma forma de um processo dividir a si mesmo em mais de uma tarefa que são executadas concorrencialmente.

¹⁴ Processadores na arquitetura Many Core são processadores especificamente projetados para executar tarefas de alta performance, com um alto nível de paralelismo.

¹⁵ O Oculus PC SDK pode ser obtido em https://developer.oculus.com/downloads/>.

¹⁶ O Nvidia GameWorks pode ser obtido em <https://developer.nvidia.com/what-is-gameworks>.



Figura 48 – Oculus Rift Development Kit, considerado para o Projeto 1.

Fonte: <http://bit.ly/2yqP44C>.

Nas figuras a seguir temos alguns resultados preliminares provenientes de testes, com geração de imagem estereoscópica em projeção nas Figuras 49 e 50.





Fonte: O Autor.



Figura 50 – Teste de geração de imagem estereoscópica projetada, utilizado no Projeto 1.

Fonte: O Autor.

Apesar de termos um início de testes funcional, por um problema com as políticas de vendas do fabricante, não foi permitida a importação do HMD em estágio de desenvolvimento para o Brasil e, por isso, em junho de 2016 fomos obrigados a desistir de utilizar o *Oculus Rift Beta* e o desenvolvimento desta versão do projeto foi descontinuada.

APÊNDICE B – PROJETO 2 - VR BASEADO NO OSVR

Iniciamos o desenvolvimento desta nova versão do projeto em junho de 2016, com características muito semelhantes ao projeto anterior (mais informações no Apêndice A), aproveitando a bagagem de conhecimentos adquiridos previamente e tivemos de fazer algumas adequações ao projeto no Módulo de Renderização.

B.1 MÓDULO DE RENDERIZAÇÃO

Removemos os *Oculus PC SDK Beta* utilizados no projeto anterior, e por isso abandonamos o desenvolvimento em plataforma Windows e adotamos o Linux na Distribuição OpenSUSE¹. Começamos a utilizar nativamente o OpenGL que posteriormente permitiria que o código fosse portado para outros Sistemas Operacionais. Por fim, em substituição ao HMD *Oculus Rift* escolhemos o *OSVR* - *Open Source Virtual Reality* conforme a Figura 51.



Figura 51 – OSVR Development Kit, considerado para o Projeto 2.

Fonte: <http://bit.ly/2Aupmg8>.

¹ A distribuição OpenSUSE, para uso em computadores pessoais, pode ser obtida em <https://software. opensuse.org>.

B.2 RESULTADOS PRELIMINARES

Neste projeto demos início aos testes com OpenGL e de forma semelhante ao projeto anterior, devido a políticas do fabricante fomos obrigados a desistir de utilizar o OSVR em julho de 2016, pois não foi permitida a importação do HMD para o Brasil, que estava em estágio inicial de desenvolvimento, por tal foi descontinuado.

APÊNDICE C – PROJETO 3 - VR BASEADO NO CARDBOARD

Iniciamos o desenvolvimento desta nova versão do projeto em agosto de 2016, com características bem distintas dos projetos anteriores. O objetivo era construir o HMD a partir do $Cardboard^1$. Para utilizar essa proposta de HMD tivemos de unificar os três módulos.

C.1 CONSTRUÇÃO DO HMD

Este HMD foi construído a partir do modelo OpenSource "VR cardboard glasses", que está disponível para download em https://www.thingiverse.com/thing:752547>.

As Figura 52 a 54 acompanham o processo de impressão do HMD.

Figura 52 – Suporte das lentes do protótipo baseado no Cardboard.



Fonte: a) Kuutio3D <https://www.thingiverse.com/thing:752547>; b) O Autor; c) O Autor.

¹ O *Cardboard* é um projeto da Google com objetivo de criar um HMD imersivo de baixo custo acoplado a um smartfone com Sistema Operacional Android.



Figura 53 – Modelo do corpo do suporte das lentes do protótipo baseado no Cardboard.

Fonte: a) Kuutio3D <https://www.thingiverse.com/thing:752547>; b) O Autor; c) O Autor.

Figura 54 – Corpo do protótipo baseado no Cardboard.



Fonte: a) Kuutio3D <https://www.thingiverse.com/thing:752547>; b) O Autor.

C.2 MÓDULO UNIFICADO

Contrariando todos os preceitos anteriores, tentamos integrar todos os módulos em App (Aplicativo mobile) que seria embarcado em smartfone baseado em Android.

Todas as arquiteturas utilizadas nos projetos anteriores teriam de ser convertidas para Arm, que é o hardware utilizado na maioria dos modelos de smartfones disponíveis no mercado. A linguagem de desenvolvimento teria de ser alterada para Java e manteríamos apenas o OpenGL remanescente dos projetos anteriores.

C.3 RESULTADOS PRELIMINARES

Os primeiros testes com o smartfone podem ser vistos na fígura 55.

Figura 55 – Montagem e testes do protótipo baseado no Cardboard.



Fonte: O Autor.

Tivemos problemas com o armazenamento dos modelos resultantes de simulação, pois, em média, cada modelo tem entre 10 e 30 Megabytes e uma simulação inteira possui de 100 a 300 modelos destes, ou seja, precisaríamos de pelo menos 1 Gigabyte de espaço para armazenar os modelos. Outro problema encontrado é que a memória era escassa, ou seja, nem todos dispositivos (smartfones) seriam capazes de rodar o App, então novamente em outubro de 2016 este projeto foi abandonado.

APÊNDICE D – ARTIGOS PUBLICADOS

Issue 3 (July)

PROGRESS IN PHYSICS

Volume 13 (2017)

Using the SALI Method to Distinguish Chaotic and Regular Orbits in Barred Galaxies with the LP-VIcode Program

Lucas Antonio Caritá^{1,2,3}, Irapuan Rodrigues², Ivânio Puerari³ and Luiz Eduardo Camargo Aranha Schiavo²

¹Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, IFSP, São José dos Campos, Brasil.

²Universidade do Vale do Paraíba, UNIVAP, São José dos Campos, Brasil.

³Instituto Nacional de Astrofísica, Óptica y Electrónica, INAOE, Puebla, México.

The Smaller Alignment Index (SALI) is a new mathematical tool for chaos detection in the phase space of Hamiltonian Dynamical Systems. With temporal behavior very specific to movements ordered or chaotic, the SALI method is very efficient in distinguishing between chaotic and regular movements. In this work, this method will be applied in the study of stellar orbits immersed in a gravitational potential of barred galaxies, once the motion of a test particle, in a rotating barred galaxy model is given by a Hamiltonian function. Using an analytical potential representative of a galaxy with bar (two degrees of freedom), we integrate some orbits and apply SALI in order to verify their stabilities. In this paper, we will discuss a few cases illustrating the trajectories of chaotic and regular orbits accompanied by the graph containing the behavior of SALI. All calculations and integrations were performed with the LP-VIcode program.

1 Introduction

One of the schemes more used to classify galaxies according to their morphology was proposed by Edwin Powell Hubble. Basically, the Hubble fork separates galaxies in two types: regular spirals (S) and barred spirals (SB). The galaxy bar, spiral arms and even galactic rings are structures that can be interpreted as disturbance to axisymmetric potential of the galactic disk.

In this work, we study the nature of some orbits immersed in analytical potentials with two degrees of freedom representing barred galaxies. In order to do this, we applied the Smaller Alignment Index (SALI) [9–13], which is a mathematical tool for distinguishing regular and chaotic motions in the phase space of Hamiltonian Dynamical Systems in analytical gravitational potentials. It is possible because the motion of a test particle in a rotating barred galaxy model is given by a Hamiltonian function.

The orbits integration and the SALI calculation were performed using the LP-VIcode program [2]. The LP-VIcode is a fully operational code in Fortran 77 that calculates efficiently 10 chaos indicators for dynamic systems, regardless of the number of dimensions, where SALI is one of them. To construct our barred galaxies models, two different sets of parameters were extracted from the paper of Manos and Athanassoula [5].

The main purpose of this paper is to show some regular and chaotic orbits, where the stability study was done using the SALI method. Such orbits were taken immersed in a mathematical model for the gravitational potential that simulates a barred galaxy in a system with two degrees of freedom.

2 Methodology

2.1 The SALI method

Considering a Hamiltonian flow (N degrees of freedom), an

orbit in the 2*N*-dimensional phase space with initial condition $P(0) = (x_1(0), \dots, x_{2N}(0))$ and two different initial deviation vectors from the initial point $P(0), w_1(t)$ and $w_2(t)$, we define the Smaller Alignment Index (SALI) by:

$$SALI(t) = \min\{\|\widehat{w}_1(t) + \widehat{w}_2(t)\|, \|\widehat{w}_1(t) - \widehat{w}_2(t)\|\}$$
(1)

where $\widehat{w}_{i}(t) = w_{i}(t) / ||w_{i}(t)||$ for $i \in \{1, 2\}$.

In the case of chaotic orbits, SALI(t) falls exponentially to zero as follows:

$$SALI(t) \propto e^{-(L_1 - L_2)t}$$
(2)

where L_1 and L_2 are the biggest Lyapunov Exponents.

When the behavior is ordered, SALI oscillates in non-zero values, that is:

$$SALI(t) \approx constant > 0, t \longrightarrow \infty$$
. (3)

Therefore, there is a clear distinction between orderly and chaotic behavior using this method.

2.2 Gravitational potential of a barred galaxy

We apply the SALI method in the study of stellar orbits immersed in a gravitational potential of barred galaxies, once the movement of a test particle in a rotating three-dimensional model of a barred galaxy is given by the Hamiltonian:

$$H(x, y, z, p_x, p_y, p_z) = = (p_x^2 + p_y^2 + p_z^2) + \Phi_T(x, y, z) + \Omega_b(xp_y - yp_x)$$
(4)

where the bar rotates around *z*; *x* and *y* contain respectively the major and minor axes of the galactic bar, Φ_T is the gravitational potential (which will be described later), and Ω_b represents the standard angular velocity of the bar. New Astronomy 60 (2018) 48-60

Contents lists available at ScienceDirect

New Astronomy

journal homepage: www.elsevier.com/locate/newast

The Smaller Alignment Index (SALI) applied in a study of stellar orbits in barred galaxies potential models using the LP-VIcode

Lucas Antonio Caritá^{*,a,b,c}, Irapuan Rodrigues^b, Ivânio Puerari^c, Luiz Eduardo Camargo Aranha Schiavo^b

^a Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, São José dos Campos, Brasil

 $^{\rm b}$ Universidade do Vale do Paraíba, São José dos Campos, Brasil

^c Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, México

ARTICLE INFO

Keywords: Chaos Galaxies: general Galaxies: kinematics and dynamics Galaxies: spiral

ABSTRACT

The Smaller Alignment Index (SALI) is a mathematical tool, not yet conventional, for chaos detection in the phase space of Hamiltonian Dynamical Systems. The SALI values has temporal behaviors very specific to ordered or chaotic motions, what makes the distinction between order and chaos easily observable in these systems. In this paper, this method will be applied to the stability study of stellar orbits immersed in gravitational potential of barred galaxies, since the motion of a test particle in a rotating barred galaxy model is given by a Hamiltonian function. Extracting four parameter sets from the Manos and Athanassoula (2011) work and elaborating a different initial conditions set for each case, we were able to introduce another point of view of their stability study for two degrees of freedom. We have also introduced two new extreme models that corroborates with the conclusions that more axisymmetric bars create an environment with less chaos and that more massive bars create an environment with more chaos. Separate studies were carried out for prograde and retrograde orbits that showed that the retrograde orbits seem more conducive to chaos. To perform all the orbits integrations we used the LP-VIcode program.

et al., 2002; 2003b; Bountis et al., 2012).

Bars are non-axisymmetric structures that can be mathematically modelled by the ellipsoidal density distribution proposed by

Ferrers (1877). It has been extensively used in many works (e.g.: Manos

and Athanassoula, 2011; Athanassoula et al., 1983; Pfenniger, 1984;

Patsis, 2002; Patsis et al., 2003a; Skokos et al., 2002a; 2002b; Patsis

decades. Athanassoula et al. (1983) integrated orbits in a rigid potential

composed of a prolate heterogeneous bar in an axysimmetric back-

ground. Varying the parameter set of the model, they studied the main

families of periodic orbits and non-periodic orbits. It was shown that

inside corotation, there are two main prograde families, one confined to

small radii, and the second aligned with the bar, as well as a retrograde

family. They also noted that more massive or more eccentric bars create more irregular motions, affecting the building of self-consistent bars.

Outside corotation, analogs are found for the two main prograde fa-

milies, as well as a -1/1 resonant family. In turn, Pfenniger (1984),

focused on the dynamics in a 3D model. Here we highlight the im-

portance of the appendix of this publication, where the author pre-

sented a very useful polynomial form for the Ferrers potential.

Let us describe some studies related to galactic bars made in the last

1. Introduction

Galaxies can be classified according to the scheme proposed by Hubble (1926). This scheme separates the spiral galaxies into two types: regular spiral galaxies (S) and barred spiral galaxies (SB). Indeed, about 65% of disk galaxies show bar-like structures (Eskridge et al., 2000; Sheth et al., 2003). Between them, the characteristic of their bars vary considerably, from faint weak bars to prominent, strong and massive bars. In this work, we will focus on the study of gravitational potentials generated by a barred galaxy structure.

The classic way to study these barred galactic potentials is to observe the structure of the orbits that are supported by them. Stellar orbits supported by a galactic potential are the basic constituents of any galactic structure. Therefore, the study of the stellar orbits characteristics is of vital importance for the understanding of the formation and evolution of these structures. Also, the analysis of periodic orbits and their stability provides valuable information about galaxy structure. There is a bond between stable periodic orbits and regular motion, once they are surrounded by tori of quasi-periodic motion. Unstable periodic orbits, by their side, originates chaos (Manos and Athanassoula, 2011).

http://dx.doi.org/10.1016/j.newast.2017.10.005

Received 20 May 2017; Received in revised form 16 August 2017; Accepted 20 October 2017 Available online 21 October 2017

1384-1076/ © 2017 Elsevier B.V. All rights reserved.





CrossMark

^{*} Corresponding author. *E-mail address*: prof.carita@ifsp.edu.br (L.A. Caritá).