



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE LICENCIATURA PLENA EM FÍSICA**

ANTONIO LUCAS ARAUJO MOREIRA

**DESENVOLVIMENTO DE SIMULAÇÃO DO PROBLEMA DOS N-CORPOS PARA
ENSINO DE GRAVITAÇÃO COM MAPLE**

**PATOS
2020**

ANTONIO LUCAS ARAUJO MOREIRA

**DESENVOLVIMENTO DE SIMULAÇÃO DO PROBLEMA DOS N-CORPOS PARA
ENSINO DE GRAVITAÇÃO COM MAPLE**

Artigo apresentado ao Departamento do Curso de Licenciatura Plena em Física da Universidade Estadual da Paraíba como requisito parcial à obtenção do título de Licenciado em Física.

Orientador: Prof. Dr. Rodrigo Cesar Fonseca da Silva

**PATOS
2020**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

M838d Moreira, Antonio Lucas Araujo.
Desenvolvimento de simulação do problema dos n-corpos para ensino de gravitação com Maple [manuscrito] / Antonio Lucas Araujo Moreira. - 2020.
27 p. : il. colorido.
Digitado.
Trabalho de Conclusão de Curso (Graduação em Física) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas , 2020.
"Orientação : Prof. Dr. Rodrigo Cesar Fonseca da Silva , Coordenação do Curso de Ciências Exatas - CCEA."
1. Ensino de física. 2. Gravitação. 3. Software Maple. 4. Simulação. I. Título

21. ed. CDD 530

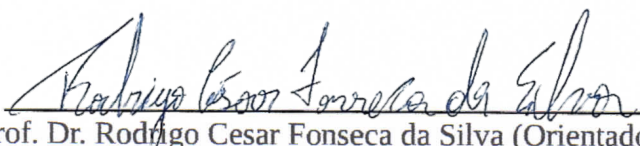
ANTONIO LUCAS ARAUJO MOREIRA

DESENVOLVIMENTO DE SIMULAÇÃO DO PROBLEMA DOS N-CORPOS PARA
ENSINO DE GRAVITAÇÃO COM MAPLE

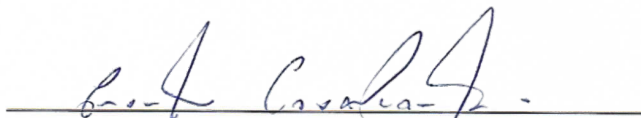
Artigo apresentado ao Departamento do Curso de Licenciatura Plena em Física da Universidade Estadual da Paraíba como requisito parcial à obtenção do título de Licenciado em Física.

Aprovado em: 06/11/2020.

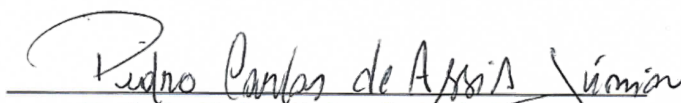
BANCA EXAMINADORA



Prof. Dr. Rodrigo Cesar Fonseca da Silva (Orientador)
Universidade Estadual da Paraíba (UEPB)



Prof. Dr. Everton Cavalcante
Universidade Estadual da Paraíba (UEPB)



Prof. Dr. Pedro Carlos de Assis Junior
Universidade Estadual da Paraíba (UEPB)

LISTA DE ILUSTRAÇÕES

Figura 1 – Vetores \mathbf{u} e $-\mathbf{u}$	8
Figura 2 – Vetor representado com coordenadas cartesianas.....	9
Figura 3 – Adição de vetores.....	10
Figura 4 – Projeção de \mathbf{v} sobre \mathbf{u}	10
Figura 5 – Órbita elíptica.....	16
Figura 6 – Comando usado para gerar posições aleatórias.....	18
Figura 7 – Condições e comandos usados para a exibição.....	18
Figura 8 – Segunda fase da rotina.....	18
Figura 9 – Gráfico de forças.....	19
Figura 10 – Gráfico de vetores posição.....	19
Figura 11 – Sistema Terra-Lua.....	19
Figura 12 – Animação com 4 corpos.....	20

SUMÁRIO

1 INTRODUÇÃO	6
2 VETORES	8
3 COMPUTAÇÃO ALGÉBRICA COM MAPLE	12
3.1 Breve histórico da computação algébrica e do Maple	12
3.2 Entrada	12
3.3 Comandos básicos	13
3.4 Ferramentas gráficas do Maple	14
3.5 Trabalhando com vetores no Maple	14
4 GRAVITAÇÃO	14
5 OBJETIVO	16
6 METODOLOGIA	17
7 RESULTADOS	17
7.1 Força gravitacional	17
7.2 Problema dos n-corpos	19
8 CONCLUSÃO	20
REFERÊNCIAS	21
APÊNDICE A – CÓDIGO DA ROTINA FORÇA GRAVITACIONAL	24
APÊNDICE B – CÓDIGO DA ROTINA PROBLEMA DOS N-CORPOS	27

DESENVOLVIMENTO DE SIMULAÇÃO DO PROBLEMA DOS N-CORPOS PARA ENSINO DE GRAVITAÇÃO COM MAPLE

DEVELOPMENT OF GRAVITATION-TEACHING TARGETED N-BODY PROBLEM SIMULATION WITH MAPLE

Antonio Lucas Araujo Moreira¹
Prof. Dr. Rodrigo Cesar Fonseca da Silva²

RESUMO

Os computadores são ferramentas de importância fundamental para o desenvolvimento científico que têm ganhado relevância também para a educação. Uma das principais formas de utilizar os recursos computacionais no ensino de física é criar simulações de fenômenos físicos. Partindo disso, utilizou-se o *software* de computação algébrica e numérica Maple para simular o problema dos n-corpos da gravitação considerando massas pontuais e poucos corpos. Dado que a gravidade é descrita através de vetores, foi feita uma revisão dos conceitos básicos de geometria analítica necessários para criar um modelo de sistema gravitacional. Também foi feito um levantamento dos recursos e características do software utilizado para criar a simulação. Como resultado preliminar, foi desenvolvida uma sub-rotina de cálculo de forças gravitacionais instantâneas e gerado um gráfico de pontos e vetores. Tomando esta sub-rotina como ponto de partida, o cálculo da simulação envolveu uma aplicação direta da lei da gravitação universal de Newton a intervalos de tempo fixos para gerar uma animação. Analisou-se a eficácia da simulação aplicando os dados orbitais do sistema Terra-Lua à rotina e verificou-se que o período da órbita simulada foi muito próximo do período real.

Palavras-chave: Ensino de Física. Gravitação. Software Maple. Simulação.

ABSTRACT

Computers are essentially important tools for scientific advancement which are getting relevant to education also. Designing simulations of physical phenomena is among the most used ways of incorporating computers into physics teaching. Thereby symbolic and numeric computation software Maple was employed to simulate the n-body problem from gravitation regarding point masses and few bodies. Since gravity is described using vector notation, some basic concepts of analytic geometry needed to model a gravitational system have been reviewed. Additionally knowledge about the tools and features of the software used to make the simulation was gathered. As a preliminary result, a subroutine to calculate instantaneous gravitational forces and generate a points and vectors plot was devised. Starting from that subroutine, the method used to simulate an n-body gravitational system consisted in a direct fixed time step application of Newton's law of universal gravitation to create an animation. Simulation effectiveness was measured by feeding the orbital data of the Earth-Moon system onto the software routine and checking that the simulated revolution period was very close to the actual period.

Keywords: Physics teaching. Gravitation. Maple Software. Simulation.

1 Concluinte do curso de Licenciatura Plena em Física pela Universidade Estadual da Paraíba Campus VII, toinho.alam@gmail.com

2 Professor titular e coordenador do curso de Licenciatura Plena em Física na Universidade Estadual da Paraíba Campus VII, rodrigo.fonsecafisico@gmail.com

1 INTRODUÇÃO

Historicamente, o desenvolvimento tecnológico e o desenvolvimento científico costumavam ser amplamente distintos, dado que muitas vezes a engenharia avançou, através da tentativa e erro, décadas antes de a ciência conseguir explicar seus princípios de funcionamento (MOURA; ROZADOS; CAREGNATO, 2006). Foi assim com Newton, que formulou os princípios da mecânica muito depois de terem aparecido inovações técnicas na mecânica e na balística, e com Faraday, exímio inventor, e Maxwell, dotado da capacidade matemática necessária para padronizar e unificar o eletromagnetismo por meio de quatro simples equações.

A produção científica, conseqüentemente, mostra-se refém das ferramentas tecnológicas disponíveis em cada período. Em contrapartida, Corsatto e Hoffman (2015) discutem que, à medida que as tênues fronteiras entre ciência e tecnologia podem ser distinguidas, atualmente, o conhecimento é que dá forma à inovação tecnológica. Na idade contemporânea, no entanto, o computador se mostra uma ferramenta-chave na construção do conhecimento científico, por sua capacidade avançada de cálculo e vasto número de aplicações, resultando em descobertas na Matemática, como o Teorema das Quatro Cores (APPEL; HAKEN, 1977), na Astrofísica, como a recente primeira imagem de um buraco negro (ASTRÔNOMOS..., 2019), além de cálculos de trajetórias de foguetes para exploração espacial e automação de satélites e sondas espaciais, e ainda na Física de Partículas, na Biologia e na Química (VALDES-PEREZ, 1995). Há também, paralelamente, plataformas virtuais de contribuição pública através da internet, como é o caso do Zooniverse (2020), voltado para a caracterização de objetos no espaço profundo, que requer o esforço conjunto de um grande número de pessoas, além da coleta de dados para melhorar a automatização desse processo através de *machine-learning*.

Com a expandida acessibilidade do computador pessoal à população, representada pela difusão de *smartphones* e notebooks no âmbito familiar e escolar, propõe-se uma nova finalidade para os computadores: auxiliar no ensino. O desejo de aplicá-los para fins didáticos data de antes do lançamento dos primeiros computadores pessoais. Alfred Bork, físico estadunidense, defende desde o início da Revolução Digital que o computador proporciona um modelo de aprendizagem interativo e autônomo, em substituição ao modelo tradicional através de aulas palestradas e livros-textos, pois cada aluno aprende de maneiras e em ritmos diferentes. Para esse fim, evidentemente, não basta incorporar apresentações de slides e pesquisas virtuais. Esses são complementos que tão só reiteram o modelo de ensino tradicional. Analise-se uma previsão feita por Bork em 1979, quando computadores não eram populares e a própria internet engatinhava:

“[...] podemos esperar mais ênfase em currículos individualizados nos quais o currículo não é amarrado a limites de tempo fixos, com cursos começando sempre no início do semestre, mas em vez disso adaptados individualmente aos alunos. Não há por quê, dado o ambiente informatizado discutido neste artigo, um aluno não possa começar e terminar um curso a qualquer momento, sujeito a quaisquer limites desejados.” (BORK, 1979, p. 10, tradução do autor)³

Na concepção de Bork, a informática possui até mesmo potencial para mudar radicalmente o padrão de ensino. Observa-se agora o cumprimento dessa previsão através da popularização de plataformas de ensino virtuais, como o Udemy (2020), com cursos que podem ser seguidos no ritmo que o aluno preferir. A plataforma, todavia, mantém-se fora do escopo do Ensino Básico e Superior, contando com cursos de qualificação profissional e desenvolvimento pessoal.

³ “[...], we can expect more emphasis on self-paced curricula in which curriculum is not tied in with fixed time constraints with courses beginning only at the beginning of the semester, but rather is adapted to individual students. There is no reason, given the computer environment discussed in this paper, why a student cannot start a course at any time and finish a course at any time, subject to any desired constraints.” (BORK, 1979, p. 10)

Anos depois, começou-se a caracterizar e categorizar de que forma os computadores estavam sendo empregados, na prática, nesse começo de sua popularização. Assim, Jong et al (1992), em pesquisa realizada no contexto do ensino superior da Holanda, descrevem três principais modalidades de utilização do computador na educação: (1) como plataforma de exposição do conteúdo; (2) como plataforma de aplicação de exames; e (3) como recurso para simulações. Também foi analisado que cada área do conhecimento dá ênfases diferentes a cada uma dessas modalidades. As ciências humanas tendem a preferir as modalidades 1 e 2, ao passo que as ciências exatas e da natureza tendem pela modalidade 3, que corresponde à simulação, o tema deste trabalho.

Antes de comentar as características da simulação, pontue-se uma característica essencial das ciências exatas, particularmente da Física, que demonstra a importância dessa ferramenta. Frequentemente há, nessas áreas do conhecimento, a necessidade de criar representações matemáticas para os objetos de estudo. Força, velocidade e posição, por exemplo, são representados por meio de vetores. Grandezas escalares, como massa e tempo, são representadas por números reais. Decorre daí que a descrição de um estado físico se dá por meio desses objetos matemáticos, pois eles possibilitam prever o comportamento de um sistema físico caracterizado pelos parâmetros correspondentes a eles.

Além disso, a representação e descrição de sistemas físicos é, em si, ferramenta de ensino e propagação da ciência, pois só é possível ao estudante compreender os conceitos que lhe são ensinados por meio de representações, sejam verbais ou visuais. Imagens e animações devem ser associadas às descrições escrita e falada para facilitar o processo de apropriação mental do conteúdo realizado pelo aluno. Neste sentido, Medeiros e Medeiros (2002) enfatizam que a simulação não é um subsídio perfeito a ser usado sem que antes sejam definidos os seus limites enquanto representação da realidade: “Não existe dúvida de que a aprendizagem na interação com seres humanos, com um professor real, é uma necessidade indispensável em muitos casos” (MEDEIROS; MEDEIROS, 2002, p. 84). As imagens estimulam a intuição humana para que se compreenda melhor os fenômenos físicos apresentados.

A simulação convenientemente alia a essa forma de representação visual dos modelos físicos o fator interatividade, enfatizado pelos modelos de educação construtivistas (FILHO, 2010). Sobre isso, Jong et al (1992) dissertam que a “simulação propicia um contexto onde os estudantes se envolvem na manipulação ativa de variáveis e parâmetros no modelo da simulação, sendo também convidados a construir o conhecimento através de processos como postular e testar hipóteses”⁴, após destacarem que a simulação é popular devido à sua consonância com o modelo construtivista de ensino-aprendizagem enquanto proporcionadora de autonomia no aprendizado. Outros estudos também defendem a eficácia da simulação nas etapas de exposição do conteúdo e de compreensão intuitiva por parte dos alunos de conceitos físicos e matemáticos abstratos (FIOLHAIS; TRINDADE, 2003; NASCIMENTO, 2014).

Para analisar a performance da simulação no estudo de física, a gravitação mostra-se uma interessante área de estudos. Dentre os que advogam pelo uso de simuladores no ensino, Pires e Veit (2006) apontam que a gravitação é um conteúdo muitas vezes deixado de lado por ser considerado de menor importância que outros conteúdos, não ser muito cobrado em vestibulares e estar quase sempre no final das grades curriculares, o que resulta em não haver tempo para estudá-lo. Trata-se, pelo contrário, de um assunto muito importante na História da Física — uma das quatro forças fundamentais — e presente no cotidiano dos alunos. Dessa forma, o trabalho deles demonstra que a simular a força gravitacional é uma excelente maneira de aumentar a carga horária de astronomia. Além de ser utilizável extraclasse, essas simulações também engajam os alunos no aprendizado.

4 “Simulation offers an environment in which learners engage in active manipulation of variables and parameters in the simulation model and are also invited to construct knowledge through processes such as stating and testing hypotheses.” (JONG et al, 1992)

Para além dos propósitos relacionados à simulação, Silva (2002) aponta também a necessidade de valorizar o ensino de gravitação no Ensino Médio, devido à sua relevância social cultural e histórica expressada por meio das tecnologias espaciais presentes em nosso cotidiano, e que não ensiná-la constitui uma descaracterização da mecânica newtoniana, por terem-se elas desenvolvido sincronicamente e vinculadas em essência. O papel da simulação nessa tarefa pode fazer-se ainda mais relevante ao notar-se que falta ao currículo do ensino médio discutir as condições e métodos científicos, pois a divulgação mais acessível à população faz a ciência parecer um produto acabado (ALMEIDA, 1998 apud SILVA, 2002).

Tendo em vista essas considerações, conclui-se que o computador é o centro de uma revolução na ciência e no ensino, ao passo que o uso de simulações, oriundo da investigação científica, começa a fazer parte também do processo de ensino. O uso dos computadores, que além de servir para processamento de grandes bancos de dados, quase sempre se dá por meio de simulações. A etapa de testagem de hipóteses que tentam prever o comportamento dos sistemas físicos depende disso. Dessa forma, incorporar esse tipo de procedimento ao ensino só deixa a educação mais rica, ao aproximar o processo cognitivo da aprendizagem ao verdadeiro processo de investigação científica.

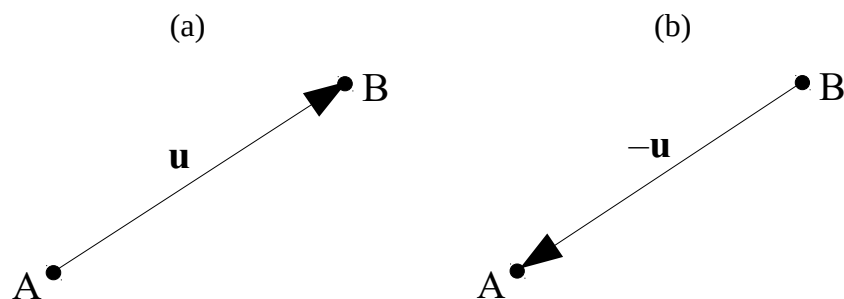
Para cumprir esse papel, existem vários *softwares* de computação algébrica, que, além de cálculos, permitem criar representações gráficas de eixos coordenados, pontos, vetores, funções e formas geométricas. Dentre esses *softwares*, o Maple se destaca por sua intuitividade na escrita e representação de fórmulas e versatilidade nas aplicações. Valer-se-á dele neste trabalho para descrever e calcular as forças gravitacionais entre corpos massivos no espaço, com o objetivo de simular um sistema planetário simples, e exibir os resultados graficamente a fim de despertar a intuição do aluno à compreensão desse tipo de sistema físico. A intenção é que o estudante que tenha contato com o material experimente sistematicamente mudar parâmetros para explorar diversas configurações de corpos e escalas de tempo e tamanho.

Para compreender melhor o escopo deste trabalho, as seguintes subseções contêm uma fundamentação teórica complementar e uma breve exposição das ferramentas utilizadas para criar um modelo matemático aproximativo de um sistema gravitacional.

2 VETORES

Um vetor é a família de segmentos de reta orientados que possuem mesmo módulo (tamanho absoluto), direção e sentido (CORRÊA, 2006). Essa definição é suficiente no contexto deste trabalho, mas uma definição mais abrangente poderia defini-lo como uma entidade matemática que satisfaz as propriedades do produto vetorial. O conceito de vetor contrapõe-se ao conceito de escalar, que geralmente denota um número real.

Figura 1 – Vetores \mathbf{u} e $-\mathbf{u}$



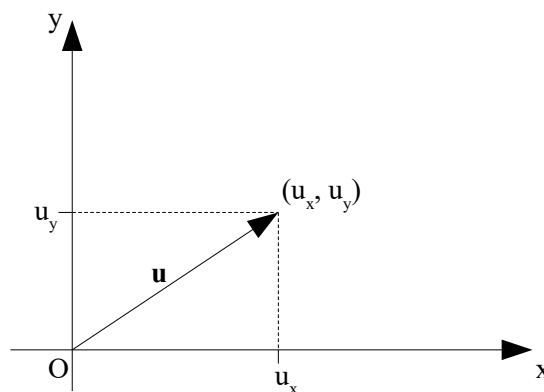
Fonte: autoria própria.

Os vetores são usados na representação de grandezas físicas que possuem módulo, direção e sentido. Grandezas que possuem apenas módulo são chamadas grandezas escalares, pois podem ser representadas apenas por um escalar e sua unidade de medida. Eles podem ser representados por setas entre dois pontos, apontando de um ao outro. Neste trabalho, a denominação de um vetor será feita por uma letra em negrito, enquanto pontos serão denominados por uma letra maiúscula. Por exemplo, na figura 1-a, temos o vetor \mathbf{u} , representado como uma seta que vai do ponto A ao ponto B. Associadamente, considerando os mesmos pontos A e B, o vetor que parte de B e vai até A possui mesmos módulo e direção do vetor \mathbf{u} , mas sentido oposto (figura 1-b). Este vetor é denominado $-\mathbf{u}$. Assim, \mathbf{u} e $-\mathbf{u}$ são chamados opostos.

Podemos representar vetores usando um sistema de coordenadas. O tipo mais comum são as coordenadas cartesianas, definidas por um ponto, chamado ponto de origem, ou simplesmente origem, escrito $(0, 0)$ em duas dimensões, e eixos perpendiculares, eixo- x e eixo- y . Ao longo de cada eixo, traçam-se retas perpendiculares a distâncias regulares e, em cada intersecção entre essas retas, define-se um ponto (x, y) , onde x e y denotam a distância das retas ao eixo- x e eixo- y , respectivamente.

Desenhando um vetor \mathbf{u} com as mesmas magnitude, direção e sentido que na figura 1-b, partindo da origem, percebemos que ele projeta sobre a direção horizontal um segmento de tamanho u_x e na direção vertical, um segmento de tamanho u_y , que são chamados componentes do vetor (figura 2).

Figura 2 – Vetor representado com coordenadas cartesianas



Fonte: autoria própria.

Assim, podemos escrever o vetor \mathbf{u} como $\mathbf{u} = (u_x, u_y)$ pois ele é representado como um segmento orientado partindo da origem do sistema de coordenadas até o ponto (u_x, u_y) . Onde u_x é a componente horizontal de \mathbf{u} e u_y , sua componente vertical. O vetor não precisa ser representado na origem, mas as suas componentes são mais facilmente visualizadas e entendidas nesse ponto.

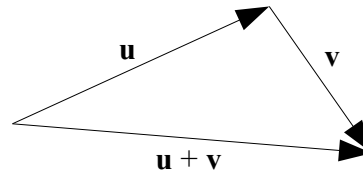
O módulo (ou norma) de um vetor “é a distância da origem à extremidade de um segmento orientado que o represente.” (CORRÊA, 2006). Pode-se calcular essa distância aplicando o Teorema de Pitágoras ao triângulo formado pela origem, a extremidade do vetor e a projeção da extremidade sobre o eixo horizontal à altura da origem. No exemplo da figura 2, esse triângulo é formado pelos pontos $(0,0)$, $(u_x, 0)$ e (u_x, u_y) .

O vetor $\mathbf{0} = (0,0)$ é o vetor nulo. Ele possui módulo igual a zero, sua direção e sentido são indeterminados e qualquer ponto do espaço é a sua representação.

Dois vetores podem ser somados, e o resultado dessa adição é obtido representando o final do primeiro no início do segundo e determinando o segmento orientado que parte da ori-

gem do primeiro vetor e termina na extremidade do segundo vetor (figura 3). Usando coordenadas cartesianas, dados os vetores $\mathbf{u} = (a, b)$ e $\mathbf{v} = (c, d)$, a soma desses dois vetores é dada pelo vetor $\mathbf{u} + \mathbf{v} = (a + c, b + d)$.

Figura 3 – Adição de vetores



Fonte: autoria própria.

A adição do vetor \mathbf{u} com o oposto de \mathbf{v} , $\mathbf{u} + (-\mathbf{v})$ pode ser representada também por $\mathbf{u} - \mathbf{v}$, chamada subtração de vetores. Repare-se que a operação $\mathbf{a} - \mathbf{a}$ resulta num vetor nulo. O vetor $\mathbf{u} - \mathbf{v}$ pode ser representado graficamente como um segmento orientado que parte da *extremidade* de \mathbf{v} e termina na *extremidade* de \mathbf{u} . Contraste-se esse vetor com o vetor soma $\mathbf{u} + \mathbf{v}$, que parte da *origem* de \mathbf{u} e termina na *extremidade* de \mathbf{v} .

A operação de multiplicação por escalar resulta num vetor de mesma direção e sentido, mas módulo multiplicado pelo escalar em questão. Essa operação é escrita como o escalar (digamos m) seguido do vetor multiplicado (digamos \mathbf{u}): $m\mathbf{u}$. Ao multiplicar um vetor por -1 , o resultado é o vetor oposto: $(-1)\mathbf{u} = -\mathbf{u}$.

Determina-se o produto escalar de dois vetores fazendo a soma dos produtos de suas componentes, conforme se vê na equação 1.

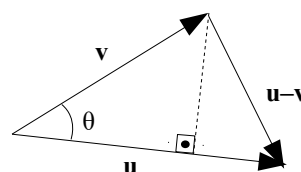
$$\mathbf{u} \cdot \mathbf{v} = u_x v_x + u_y v_y \quad (1)$$

O produto escalar de vetores tem as propriedades comutativa e distributiva. É possível calcular o módulo de um vetor \mathbf{u} , escrito $\|\mathbf{u}\|$, através de seu produto escalar consigo próprio, conforme equação 2.

$$\mathbf{u} \cdot \mathbf{u} = u_x^2 + u_y^2 = \|\mathbf{u}\|^2 \quad (2)$$

Um vetor unitário é aquele cujo módulo é 1. Dividindo-se um dado vetor \mathbf{u} pelo seu módulo $\|\mathbf{u}\|$, obtém-se o vetor unitário $\hat{\mathbf{u}}$ na sua direção, também chamado versor, representado com um acento circunflexo. Os eixos das coordenadas cartesianas são representados por vetores unitários chamados $\hat{\mathbf{i}}$, para o eixo-x, $\hat{\mathbf{j}}$ para o eixo-y e $\hat{\mathbf{k}}$ para o eixo-z em três dimensões. Usando esses vetores unitários, existe outra forma de se representar dado vetor $\mathbf{u} = (u_x, u_y, u_z)$, que é $\mathbf{u} = u_x \hat{\mathbf{i}} + u_y \hat{\mathbf{j}} + u_z \hat{\mathbf{k}}$.

Figura 4 – Projeção de \mathbf{v} sobre \mathbf{u}



Fonte: autoria própria.

Observa-se, ao representar dois vetores desenhado com mesma origem, que entre eles há um ângulo, que aqui será denotado por θ . A distância da origem de \mathbf{u} até a sua intersecção com a linha tracejada partindo da extremidade de \mathbf{v} , perpendicular a \mathbf{u} é igual ao produto do módulo de \mathbf{v} com o cosseno de θ . Essa é a projeção ortogonal de \mathbf{v} sobre \mathbf{u} . Aplicando a lei dos cossenos ao triângulo formado pelos vetores \mathbf{u} e \mathbf{v} na figura 4, deduz-se uma relação entre o produto escalar e a projeção de um vetor sobre o outro (equação 3).

$$\begin{aligned} \|\mathbf{u} - \mathbf{v}\|^2 &= \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\|\mathbf{u}\|\|\mathbf{v}\|\cos\theta \\ (\mathbf{u} - \mathbf{v})(\mathbf{u} - \mathbf{v}) &= \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\|\mathbf{u}\|\|\mathbf{v}\|\cos\theta \\ \mathbf{u} \cdot \mathbf{u} - \mathbf{u} \cdot \mathbf{v} - \mathbf{v} \cdot \mathbf{u} + \mathbf{v} \cdot \mathbf{v} &= \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\|\mathbf{u}\|\|\mathbf{v}\|\cos\theta \\ \|\mathbf{u}\|^2 - 2\mathbf{u} \cdot \mathbf{v} + \|\mathbf{v}\|^2 &= \|\mathbf{u}\|^2 + \|\mathbf{v}\|^2 - 2\|\mathbf{u}\|\|\mathbf{v}\|\cos\theta \\ \mathbf{u} \cdot \mathbf{v} &= \|\mathbf{u}\|\|\mathbf{v}\|\cos\theta \end{aligned} \quad (3)$$

Assim, podemos definir o produto escalar dos vetores \mathbf{u} e \mathbf{v} como o produto do módulo de \mathbf{u} com a projeção ortogonal de \mathbf{v} sobre \mathbf{u} . A principal consequência dessa relação é que se pode calcular o ângulo entre dois vetores usando o produto escalar (equação 4).

$$\cos\theta = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} \quad (4)$$

Como o cosseno de 90° é igual a zero, se o produto escalar $\mathbf{u} \cdot \mathbf{v}$ for igual a zero, então \mathbf{u} e \mathbf{v} são ditos vetores ortogonais. Se $\cos\theta = 1$, \mathbf{u} e \mathbf{v} são denominados vetores paralelos. Por fim, se $\cos\theta = -1$, \mathbf{u} e \mathbf{v} são chamados vetores antiparalelos.

Outra operação com vetores importante é o produto vetorial. Ele é definido entre dois vetores situados em três dimensões $\mathbf{u} = (u_x, u_y, u_z)$ e $\mathbf{v} = (v_x, v_y, v_z)$ como o determinante de uma matriz conforme indicado na equação 5, onde $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ e $\hat{\mathbf{k}}$ são os vetores unitários do espaço cartesiano. O produto vetorial é representado por $\mathbf{u} \times \mathbf{v}$.

$$\mathbf{u} \times \mathbf{v} = \det \begin{pmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ u_x & u_y & u_z \\ v_x & v_y & v_z \end{pmatrix} \quad (5)$$

Diferentemente do produto escalar, a ordem dos vetores na operação resulta no seu oposto, ou seja, $\mathbf{v} \times \mathbf{u} = -\mathbf{u} \times \mathbf{v}$, pela que é chamada propriedade anticomutativa. O produto vetorial é um vetor ortogonal a ambos os seus operandos, com sentido determinado pela regra da mão direita, onde os dedos se curvam do multiplicando ao multiplicador e o polegar mostra o sentido do produto. Ele possui módulo igual ao produto dos módulos dos vetores vezes o seno do ângulo entre eles (equação 6).

$$\|\mathbf{u} \times \mathbf{v}\| = \|\mathbf{u}\|\|\mathbf{v}\|\sin\theta \quad (6)$$

Como esta também é a fórmula da área de um paralelogramo de lados iguais aos módulos dos vetores, o produto vetorial pode ser usado para calcular essa área. Além disso, como um triângulo é sempre a metade de um paralelogramo, é possível calcular a área do triângulo determinado por dois vetores conforme visto na equação 7.

$$A = \frac{\|\mathbf{u} \times \mathbf{v}\|}{2} \quad (7)$$

3 COMPUTAÇÃO ALGÉBRICA COM MAPLE

Nesta seção serão abordados tópicos sobre a utilização do Maple, enfatizando as ferramentas relevantes para a realização de uma simulação de sistema gravitacional.

3.1 Breve histórico da computação algébrica e do Maple

Nas décadas de 60 e 70, surgiram os primeiros sistemas computacionais algébricos, que, ao contrário dos primeiros sistemas de computação eletrônica, utilizam definições exatas e formais de objetos e operações matemáticas em vez de utilizar apenas operações de ponto flutuante, dando mais exatidão e facilidade para manipular expressões algébricas. Esses sistemas processam as entradas de maneira mais parecida com como um ser humano as processaria, munidos da capacidade superior de cálculo da máquina.

O Maple foi desenvolvido pela Universidade de Waterloo, no Canadá, em colaboração com o Instituto ETH, na Suíça (MARIANI, 2005), com o objetivo de facilitar a manipulação e visualização de objetos matemáticos, podendo ser usado para resolução e exploração de problemas matemáticos de uma ampla gama de áreas. Possui uma interface intuitiva e permite escrever em notação matemática usual, de forma prática e controlável. Por isso, pode ser usado como ferramenta expositiva aplicável a diversas áreas do conhecimento, não só da matemática, mas também da física e da engenharia.

3.2 Entrada

Em suas últimas versões, o Maple aceita duas formas de entrada. A primeira, com a qual o usuário tem contato ao começar a utilizá-lo, é mais intuitiva e livre. É possível digitar expressões matemáticas e comandos e apertar Enter para ver o resultado. As expressões, escritas em forma de notação matemática, podem ser ajustadas enquanto se digita, por exemplo, as frações aparecem como o numerador sobre o denominador separados por uma barra horizontal, podendo ser expandidos sem necessidade do uso de parêntesis, apenas movendo o cursor. Também é possível escolher, numa paleta, certas operações e objetos matemáticos que só seriam acessíveis através de comandos de texto no programa.

A segunda forma de entrada é a tradicional, que as versões mais antigas do programa usavam. As expressões matemáticas, operações e alguns objetos são digitados de maneira mais comum na computação. Cada linha se inicia com “>” e cada comando é finalizado com “;” (ponto e vírgula), para exibir o resultado, ou “:” (dois pontos), para ocultar o resultado, que fica apenas armazenado na memória (poupando processamento gráfico).

Na linguagem do Maple, o símbolo “=” (igual) é usado para digitar equações, enquanto “:=” é usado para atribuir um nome ou variável a uma expressão, objeto ou resultado para que este possa ser utilizado em um comando posterior. O Maple distingue maiúsculas de minúsculas nos nomes dos comandos e ao nomear objetos.

Pode-se acrescentar comentários usando o símbolo “#” (jogo-da-velha) antes de um texto a ser ignorado no processamento.

Há, no Maple, algumas constantes globais, que não podem ser usadas para nomear outros objetos. É o caso de “I”, reservado para a raiz quadrada de -1 , “Pi”, reservado para o número π (pi) e “infinity”, reservado para a entidade matemática ∞ (infinito).

Os resultados das expressões não são aproximados. Portanto, se tiver sido digitada uma fração, o resultado aparece em forma de fração. Se tiverem sido entrados dados em formato de *float*, que é como se denominam os números decimais na computação, os resultados ficam aproximados. É possível, porém, converter entre os tipos numéricos usando o comando “convert”.

3.3 Comandos básicos

Todo comando é digitado com seus argumentos entre parêntesis, separados por vírgula e em ordem: “comando(argumento 1, argumento 2, ...)”. A página de ajuda do Maple, acessada no programa através da tecla F1, contém uma lista de todos os comandos e informações acerca de como devem ser utilizados e exemplos de uso.

Algumas operações matemáticas são acessíveis por comandos do programa, como a raiz quadrada, pelo comando “sqrt” e as funções trigonométricas: “sin” para seno, “cos” para cosseno, “tan” para tangente e assim por diante.

Entre os comandos básicos, há o comando “plot”, que constrói o gráfico de uma função de uma variável e o comando “plot3d”, que constrói o gráfico de uma função de duas variáveis (em 3D). É possível ajustar vários parâmetros de exibição para esses comandos. Os parâmetros dos comandos são escritos após os argumentos.

Às vezes, os parâmetros ou argumentos são listas ou conjuntos. As listas e conjuntos, assim como outros objetos, podem ser salvos com um nome. Uma lista é definida entre colchetes, e seus elementos são ordenados, podendo ter elementos repetidos, por exemplo, o comando “L := [1, 2, 1, x, Pi, a];” define uma lista atribuída ao nome “L”, possuindo seis elementos em uma sequência específica, que são números e letras misturados, mas as letras são, na verdade, variáveis. Já um conjunto é definido entre chaves, seus elementos não são ordenados, e não pode ter elementos repetidos, porque o conjunto os conta como um só, por exemplo, o comando “conj:={1, 2, 1, x, Pi, a}” cria o conjunto {1, 2, Pi, a, x}, pois o Maple exibe os números primeiro, do menor para o maior e as letras em ordem alfabética, além de ter somente cinco elementos, pois o “1” foi repetido na definição, não sendo contado novamente. Pontos e vetores são escritos no Maple na forma de listas. É possível ainda criar uma lista com o comando “array”, que pode ser definido antes de seus elementos, possibilitando que eles sejam acrescentados a ele posteriormente. O seu primeiro argumento é o intervalo para os números correspondentes a cada elemento na lista (possibilitando que a contagem inicie-se em qualquer outro número, em vez de 1, que é o padrão para listas), e o segundo é uma lista com alguns ou todos os elementos da lista.

O comando “op” retorna um ou mais operandos do seu argumento, que pode ser uma operação, uma lista ou uma equação: “op(L);” retorna todos os elementos da lista “L” (do parágrafo anterior), enquanto que “op(4, L);” retorna somente “x”, que é o 4º elemento da lista. O comando “numelems” retorna o número de elementos do argumento: “numelems(L);” retorna “6”, que é a quantidade de elementos de “L”.

É de costume iniciar um programa com o comando “restart”, que apaga todas as constantes e objetos definidos anteriormente, começando o programa do zero sempre que for reexecutado e evitando erros.

O Maple aceita condicionais do tipo “if-else” e *loops* do tipo “for-while”, presentes nas linguagens de programação. Já o comando “seq” é parecido com o *loop* “for-while”, realizando o mesmo tipo de tarefa, porém de forma mais específica. Ele retorna o resultado de uma função, repetido para vários valores de uma variável local. Por exemplo: “seq(4*i, i=0..5);” retorna os resultados da multiplicação do número quatro por todos os números inteiros de 0 a 5, de forma inclusiva, em sequência: “0, 4, 8, 12, 16, 20”. Semelhante ao comando “seq”, há também o comando “add”, que soma os termos de uma sequência, possuindo os mesmos argumentos de “seq”. Se usássemos “add”, no lugar de “seq”, no exemplo anterior, obteríamos a soma dos elementos da sequência gerada, “60”.

Muitos comandos estão disponíveis apenas em ambientes restritos. Para acessá-los, devem-se carregar seus respectivos pacotes através do comando “with”, podendo ser carregados todos os comandos do pacote ou só alguns, listados nos argumentos do comando “with”. Um comando dentro de um pacote também pode ser acessado escrevendo o nome do pacote

seguido do nome do comando entre colchetes, sem necessidade de carregar o pacote ou o comando. Um exemplo de comando de pacote que foi utilizado neste trabalho é o comando “Generate”, presente no pacote “RandomTools”, que gera um número aleatório a partir de alguns parâmetros definidos pelo usuário. Alguns comandos gráficos também só são acessíveis no ambiente de seus pacotes.

3.4 Ferramentas gráficas do Maple

Os principais pacotes gráficos do Maple são “plots” e “plottools”. O pacote “plots” tem várias ferramentas básicas de gráficos. Dentre elas, o comando “display”, que permite exibir vários gráficos e desenhos juntos, podendo exibi-los em sequência, como uma animação, usando o parâmetro “insequence”. Em “plots”, há também os comandos “arrow”, que permite desenhar uma seta, normalmente representando um vetor, bidimensional ou tridimensional, e o comando “implicitplot”, que permite desenhar gráficos de lugares geométricos indescritíveis por funções, como elipses e hipérbolas, com uma equação que descreve uma relação entre as variáveis.

Já o pacote “plottools” contém ferramentas de desenho para várias formas geométricas, tais como elipses, polígonos, esferas, sólidos geométricos e arcos. Além de ferramentas para manipulação de imagens, como refletir e rotacionar. Os comandos do pacote “plottools” geram resultados que devem ser postos no comando “display” do pacote “plots” para ser exibidos.

O parâmetro “scaling” do comando “display”, quando definido para “constrained” serve para deixar os eixos dos gráficos na mesma proporção. Sem ele, um gráfico pode parecer achatado, pois por padrão, o Maple estende cada eixo separadamente para preencher todo o quadro. Esse parâmetro não é muito útil quando os eixos representam grandezas diferentes, mas na representação dum espaço métrico, onde as coordenadas representam distâncias de magnitudes similares, ele deixa o gráfico mais realista.

3.5 Trabalhando com vetores no Maple

O principal pacote para trabalhar com vetores é o “VectorCalculus”. O pacote “LinearAlgebra” trabalha fundamentalmente com matrizes, mas também pode ser usado para manipular vetores. Esses pacotes possuem comandos para calcular a norma de um vetor (“Norm”), o produto escalar (“DotProduct”) e o produto vetorial (“CrossProduct”), além das operações de adição (usando +) e subtração de vetores (usando –).

Em alguns casos, vetores podem ser descritos simplesmente como listas de números: $[x, y]$ (valor da abscissa primeiro e, em seguida, da ordenada) para duas dimensões ou $[x, y, z]$, para três. Mas podem ser criadas listas com mais elementos para vetores de dimensões superiores. O comando “Vector” também cria um vetor, recebendo uma lista como argumento. O vetor também pode ser definido entre os sinais de menor que e maior que: $\langle x, y \rangle$ ou $\langle x, y, z \rangle$. Algumas dessas maneiras de definir vetores não podem ser usadas em certos contextos, portanto, é preciso conhecer os argumentos do comando para saber qual pode ser utilizada.

4 GRAVITAÇÃO

A gravidade é objeto de estudo da ciência desde a antiguidade (MARTINS, 2006). Naquela época, a gravidade não era entendida como uma aceleração, ou uma força, mas como uma característica intrínseca dos corpos, correspondente a seu peso. No Ocidente, os filósofos gregos foram os primeiros a registrar suas formulações das leis do movimento, separando os

movimentos dos corpos próximos à superfície da Terra, naturalmente sob influência da gravidade, e dos corpos celestes.

No campo da dinâmica nas proximidades da Terra, onde ocorre o fenômeno da queda dos corpos massivos, Aristóteles (385 A.C.–322 A.C.) exerceu notável influência na história da ciência, e suas descrições do movimento foram sendo gradualmente refutadas e corrigidas ao longo da história. A partir do século XIV, estudiosos de Oxford e Paris chegaram, a partir de argumentos racionais baseados na cinemática aristotélica, a descrições matemáticas bastante semelhantes às conhecidas atualmente, embora mais rudimentares e sem o respaldo de experimentos, sem formalismo e sem definições claras de unidades de medida. Encontram-se aí conceitos preliminares de movimento uniformemente acelerado, momento linear (sob o título de quantidade de movimento, o ímpeto), e massa (entendida como quantidade de matéria) (MOODY, 1951).

No campo da astronomia, a descrição presente no *Almagesto* de Ptolomeu (90 d.C. – 168 d.C.), compilando séculos de estudos do movimento dos astros em um modelo geoestático, prevaleceu por toda a Idade Média, devido à sua precisão e concordância com as observações. Copérnico (1473–1543) foi o primeiro a contestar esse modelo com um modelo helioestático, porém sem respaldo empírico. Foram necessários mais alguns anos até que as observações precisas de Brahe (1546–1601), favorecessem o modelo de Copérnico através das modificações de Kepler (1571–1630), descritas por suas três leis (DAMASIO, 2011).

Finalmente, cientistas da Idade Moderna, munidos de demonstrações matemáticas em conjunto com verificações empíricas, chumbaram as teorias físicas clássicas sobre gravitação usadas até hoje. Galileu (1564–1642) descreveu que a ação gravitacional provoca uma aceleração constante nos objetos próximos à superfície terrestre, independentemente de sua massa. Anos depois, Newton (1643–1727) formalizou a descrição do movimento gravitacional, unificando a queda dos objetos conforme descrita por Galileu e o movimento dos planetas conforme descrito por Kepler em uma única lei da Gravitação Universal, escrita com notação moderna de vetores na equação 7.

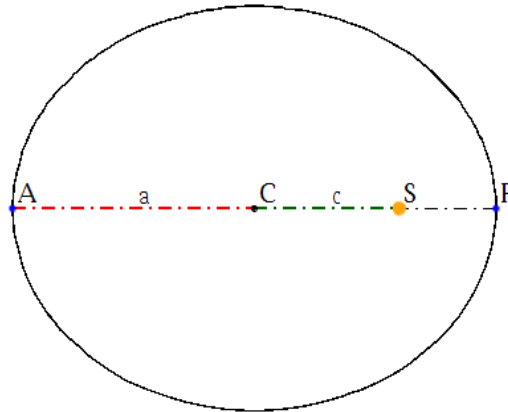
$$\mathbf{F}_g = \frac{G m_1 m_2}{d^2} \hat{\mathbf{d}} \quad (8)$$

Nesta equação, \mathbf{F}_g é a força gravitacional que um corpo de massa m_1 provoca sobre um corpo de massa m_2 , separado daquele por uma distância d na direção do vetor unitário $\hat{\mathbf{d}}$ e G é a constante gravitacional. Esta força representa um par ação e reação, ou seja, o corpo de massa m_2 causa uma força de igual intensidade e na mesma direção, porém em sentido oposto sobre o corpo de massa m_1 . Algumas soluções dessa equação são descritas através das leis de Kepler, que descrevem as trajetórias dos planetas orbitando o Sol, podendo também ser aplicadas para satélites orbitando planetas. Considera-se, no modelo de Kepler, que os corpos são acúmulos pontuais de massa, e que um dos corpos possui massa muito maior do que a de todos os outros, de forma que a gravidade de cada um dos outros não perturbe as órbitas dos outros (ROY, 2005). Dessa forma, cada trajetória é tratada de maneira independente.

A primeira lei de Kepler afirma que as trajetórias dos planetas são elipses com o Sol em um dos focos. A figura 5 é um esquema de uma órbita elíptica de Kepler. C é o centro da órbita, este ponto é apenas uma abstração geométrica, o Sol (ou o corpo dominante, se forem considerados outros tipos de sistemas) se localiza no ponto S , correspondendo a um dos focos da elipse. P e A são os pontos extremos da órbita, chamados apsides (substantivo feminino e paroxítono). P é o ponto da órbita mais próximo de S , chamado periapside, e A é o ponto mais distante, chamado apoapside. Dependendo do sistema, eles recebem denominações específicas: para o Sol, são o periélio e o afélio, para a Terra são o perigeu e o apogeu, e assim por diante. O segmento $c \equiv \overline{SC}$ é a distância entre o Sol e o centro da órbita e o segmento

$a \equiv \overline{CA} \equiv \overline{CP}$ é a metade do eixo maior da elipse, ou semieixo maior. A razão $e=c/a$ é a excentricidade da órbita. Geralmente a excentricidade é muito menor do que a que foi apresentada na figura, mas esta foi exagerada para melhor ilustrar os elementos da órbita. A elipse é simétrica em relação ao eixo maior, e a distância média (ou raio médio) entre o Sol e o planeta corresponde ao semieixo maior da elipse.

Figura 5 – Órbita elíptica



Fonte: autoria própria.

A segunda lei de Kepler afirma que as áreas varridas por uma linha entre o planeta e o Sol em intervalos de tempos iguais são iguais. Como consequência disso, a velocidade orbital é máxima na periapside e mínima na apoapside.

Por fim, a terceira lei relaciona os períodos das órbitas aos seus semieixos maiores através da relação descrita pela equação 9 (SWOKOWSKI, 1994), onde T é o período (tempo que um planeta leva para voltar ao mesmo ponto da órbita), a é o semieixo maior (correspondente ao raio médio), G é a constante gravitacional e M é a massa do Sol (ou corpo cuja massa é dominante no sistema).

$$\frac{T^2}{a^3} = \frac{4\pi^2}{GM} \quad (9)$$

Quando o sistema contém mais de dois corpos, cada corpo age um sobre o outro dois a dois e as soluções deixam de ser triviais. Tornam-se necessários métodos aproximativos para calcular as trajetórias de tais sistemas, um processo facilitado pelo uso de computadores. Este problema é conhecido como o problema dos n -corpos, e foi formulado pela primeira vez por Newton, com o objetivo de prever as posições e velocidades em qualquer instante, dadas uma posição, uma velocidade, e uma massa inicial para cada corpo. O problema dos n -corpos é subdividido em dois casos gerais: sistemas com poucos corpos, como sistemas planetários; e sistemas com muitos corpos, como uma galáxia. Neste trabalho não serão estudados sistemas com muitos corpos, que requerem procedimentos estatísticos específicos, nem serão consideradas as dimensões dos corpos tratados, tidas como desprezíveis conforme a escala do sistema que se pretende simular.

5 OBJETIVO

O objetivo deste trabalho é desenvolver uma ferramenta que possibilite, tendo como base um *software* de computação algébrica, o Maple, a apresentação didática e intuitiva das leis da gravitação e do movimento planetário, mostrando ao mesmo tempo a importância dos

softwares de computação gráfica nas representações de sistemas físicos. Para isso, buscou-se criar uma animação ilustrativa do problema dos n-corpos através de um método simples de aproximação no Maple.

6 METODOLOGIA

Esse trabalho foi desenvolvido através de encontros semanais com o orientador para avançar e definir gradualmente os limites e objetivos do trabalho, expandindo os resultados a cada encontro. Simultaneamente foi feito o estudo bibliográfico em artigos e livros sobre as ferramentas necessárias para a simulação: *softwares* de computação algébrica, particularmente o Maple, vetores, lei da gravitação universal e leis de Kepler. Este estudo foi complementado pela exploração no Maple através de tentativa e erro para descobrir as melhores formas de usá-las na construção dos simuladores.

O método de aproximação usado neste trabalho consiste em repetir em *loop*, um número arbitrário de vezes o seguinte procedimento: dados os parâmetros iniciais de posição, velocidade e massa, calcular, a partir da força gravitacional, uma próxima velocidade v , pela fórmula $v = v_0 + a \cdot \Delta t$, da definição de aceleração média, uma próxima posição s , pela fórmula $s = s_0 + v \cdot \Delta t$, da definição de velocidade média, e uma próxima força resultante pela lei da gravitação universal. Nestas fórmulas, v_0 é a velocidade inicial de cada passo, s_0 é a posição inicial, Δt é um intervalo de tempo muito pequeno, caso no qual a velocidade média é aproximadamente igual à velocidade instantânea e a é a aceleração média da força gravitacional agindo no passo de tempo, dada por $a = F_g / m$ onde, F_g é a força gravitacional resultante inicial sobre um corpo de massa m . Na próxima interação deste *loop*, os parâmetros calculados tornam-se os parâmetros iniciais, e isso se repete até o fim do *loop*.

Para verificar a exatidão da simulação, ter-se-á por modelo o sistema Terra-Lua, para o qual podem-se obter os parâmetros orbitais necessários em Williams (2020) e verificar-se-á se a simulação forma uma órbita e se o período dessa órbita corresponde ao período de revolução da Lua ao redor da Terra. O intervalo de tempo utilizado nos cálculos foi definido arbitrariamente como 5000 segundos, considerando-se que este intervalo é de ordem inferior a 1 milésimo do período de revolução da Lua.

7 RESULTADOS

Usando os conceitos e ferramentas descritos nas seções anteriores, foram produzidas duas rotinas no Maple que aplicam e simulam leis físicas relacionadas à gravitação. A primeira rotina cria um gráfico de forças gravitacionais instantâneas sobre um dos corpos, representados por pontos. A segunda rotina produz uma animação ilustrativa do problema dos n-corpos em três dimensões, tendo como base a primeira rotina. Os códigos dessas rotinas estão nos apêndices.

7.1 Força gravitacional

Com esta rotina, pretende-se ilustrar as forças gravitacionais agindo sobre um, dentre vários corpos num único instante de tempo. A rotina se inicia com o comando “restart” e chamando os pacotes e comandos adicionais. O usuário define então as constantes e parâmetros que o programa utilizará para produzir os gráficos. Dentre eles, estão o número de dimensões, o número de corpos, suas posições e suas massas.

A primeira fase de execução é a preparação dos dados. Se o usuário não houver definido os parâmetros, o programa os cria aleatoriamente. Na figura 6, mostram-se os parâmetros utilizados nesta rotina para as posições aleatórias. Ele gera uma lista cujo número de termos é

igual ao número de dimensões, os números são em formato *float*, definidos no intervalo do limite do módulo das coordenadas, com o número de dígitos definido pela precisão dos valores de posição definida pelo usuário, usando o método uniforme (em contraponto ao logarítmico). Nesta rotina, a constante gravitacional foi definida como 1 para simplificar os cálculos.

Figura 6 – Comando usado para gerar posições aleatórias

```
r := Generate(list(float(range = -tmax .. tmax, digits = dp, method =
uniform), nd));
```

Fonte: autoria própria.

A segunda fase é a de cálculo das forças. Aqui se calcula a força gravitacional entre o corpo referencial e todos os outros (Figura 7), além do vetor força resultante.

Figura 7 – Condições e comandos usados para a exibição

```
#exibição dos gráficos
if np>0 and nd<=3 then
display(pp, tp, vp, ppo, scaling=constrained);
display(pp, vf, vfr, ppo, scaling=constrained);
elif nd<=3 then
plot([]);
elif nd>3 then textplot3d([0,0,0,"Por favor, defina um número de dimensões
menor ou igual a 3"], 'color'=blue);
end if;
```

Fonte: autoria própria.

A terceira fase é a preparação da exibição final dos resultados, incluindo a criação dos gráficos de pontos, vetores e texto. O gráfico do ponto de referência é feito de forma diferente, para ele se destacar entre os demais.

O gráfico de texto rotula os corpos com base na sua posição na lista de pontos, para identificá-los no gráfico, colocando o seu respectivo número acima e à esquerda do ponto correspondente à sua posição. Para o gráfico não ficar apinhado de números, foi estipulado um limite para o número de corpos que produz um gráfico com rótulos. Se o número de corpos ultrapassar esse limite, a rotina não cria um gráfico de texto.

Figura 8 – Segunda fase da rotina

```
#cálculo das forças gravitacionais entre o ponto de referência e os demais
Vf := array(1 .. np, [seq(G*mo*Mfix[i]*Vector(Vfix[i]-o)/Norm(Vector(Vfix[i]
-o))^3, i = 1 .. np-1)]);

#cálculo da força gravitacional resultante
Vf[np] := add(Vf[i], i = 1 .. np-1):
```

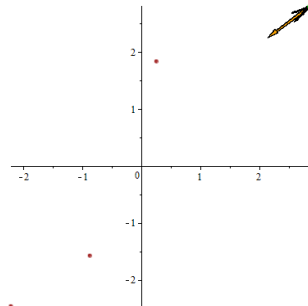
Fonte: autoria própria.

A quarta e última etapa é a exibição (Figura 8). Nesta etapa são criados dois gráficos: o primeiro contendo os gráficos dos pontos, o de vetores posição e de rótulos dos pontos, se houver. O segundo gráfico contém os gráficos de pontos, de vetores força e vetor força resultante, mas sem os rótulos. A rotina de ilustração das forças gravitacionais termina aqui. Nas figuras 9 e 10, um exemplo de execução do programa.

A aparência do gráfico depende da interação entre as constantes e parâmetros definidos inicialmente na rotina. O ato de manipulá-los para testar diferentes configurações é o fator

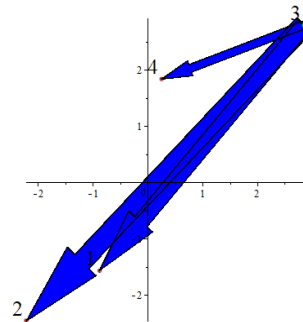
didático desse simulador, pois o aluno vai descobrindo como funciona a lei da gravitação com elevado grau de controle sobre o sistema.

Figura 9 – Gráfico de forças



Fonte: autoria própria.

Figura 10 – Gráfico de vetores posição



Fonte: autoria própria.

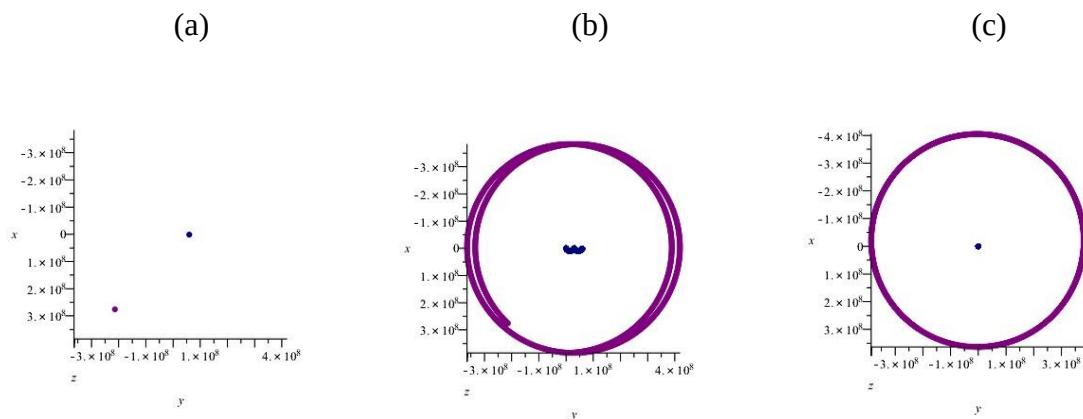
7.2 Problema dos n-corpos

Tendo como base o código escrito para a rotina “Força gravitacional” (seção 6.1), foi desenvolvida uma animação em três dimensões do problema dos n-corpos, onde cada quadro é gerado a partir do anterior, usando os parâmetros físicos de posição, velocidade, força e massa, que são armazenados em listas do programa. A lista de forças, particularmente, é calculada a partir de uma matriz antissimétrica de forças gravitacionais, porque cada corpo exerce uma força de igual intensidade, porém sentido contrário um sobre o outro.

Os quadros da animação são gerados através de um *loop* acrescentando a cada interação o próximo valor de velocidade, posição e força, nesta ordem. Cada interação cria um gráfico de pontos das posições dos corpos, com cores diferentes para cada corpo.

Para verificar a exatidão do método, valendo-se das informações orbitais e de massa do sistema Terra-Lua, fez-se uma animação de testes para ajustar parâmetros como a quantidade de quadros calculados e a escala de tempo dos quadros que permitissem uma animação fluida, mas de rápido processamento. O resultado final está apresentado na figura 11. A figura 11-a representa um dos 900 quadros da animação gerada, onde cada quadro representa 5 mil segundos, e a figura 11-b é uma sobreposição de todos os quadros, para formar um rastro. Observa-se que a Terra também se move ao redor de um ponto e ao mesmo tempo para a direita.

Figura 11 – Sistema Terra-Lua

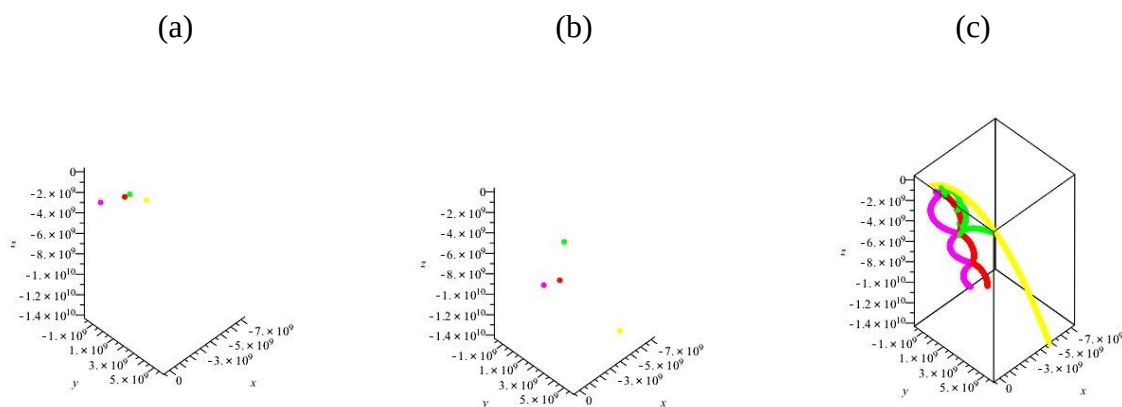


Fonte: autoria própria.

Podemos verificar, se acrescentarmos um termo oposto à posição da Terra no comando que gera o gráfico dos pontos, colocando assim a Terra na origem espaço cartesiano, que a Lua retorna ao ponto inicial aproximadamente aos 472 quadros (cada quadro correspondendo a 5 mil segundos). Isto resulta em um período de revolução de 27,3148 dias, um erro de 0,025% em relação ao valor real, de 27,3217 dias, demonstrando o sucesso da simulação. Na figura 11-c, temos a representação de todos os quadros com a Terra como ponto de referência, evidenciando que a órbita é elíptica (o apogeu, na parte superior, está a $4 \cdot 10^8$ m, e o perigeu, na parte inferior, a $3,6 \cdot 10^8$ m).

Tendo como base as dimensões dos parâmetros da Terra e da Lua, criaram-se intervalos para geração de distâncias, velocidades e massas aleatórias e gerou-se uma nova animação, com 4 corpos, para verificar a complexidade do movimento quando mais corpos aparecem no sistema (Figura 12). Através de alguns testes variando esses intervalos, entendeu-se que se essas dimensões forem muito pequenas, o movimento é muito lento e os corpos quase não saem do lugar. Do contrário, o movimento é rápido e os corpos logo se espalham pelo espaço. Os parâmetros foram ajustados para ficar num ponto de equilíbrio entre esses dois extremos e gerar um movimento visualmente interessante.

Figura 12 – Animação com 4 corpos



Fonte: autoria própria.

8 CONCLUSÃO

Conforme se vê nos resultados, o Maple se destaca como uma ferramenta didática, com capacidade para criar gráficos representativos de modelos físicos, despertando a curiosidade e a vontade de explorar as diferentes possibilidades de parâmetros do sistema trabalhado. Há nisto um potencial para aplicação dessas animações em sala de aula, possibilitando aos alunos o controle dos parâmetros que geraram os gráficos. Dessa forma, eles podem aprender conceitos de gravitação, geometria analítica e linguagem de computação. Pode-se sugerir que descubram um ponto de equilíbrio entre as situações às quais a simulação não se aplica adequadamente (corpos muito massivos ou se movendo muito rápido).

O conhecimento sobre vetores foi fundamental para escrever o código das simulações, dado que o sistema físico descrito depende fundamentalmente deles, e a sua aplicação incorreta resultaria em falhas nos resultados. Da mesma forma, o conhecimento sobre o sistema físico e o ambiente do computador também foram integrados de forma interdisciplinar. Verificou-se que, no caso da simulação do sistema Terra-Lua, a rotina desenvolvida sucedeu em corresponder à realidade. Podem-se buscar outros sistemas que podem ser representados acuradamente com este método utilizando procedimentos semelhantes.

Por fim, este trabalho partiu de uma mudança de paradigma na educação que está ocorrendo nos últimos anos, onde os computadores são mais utilizados como ferramenta didática e o aluno possui mais controle sobre o próprio aprendizado. A informática não deve servir apenas para a administração da escola e a apresentação de slides. Cabe ao professor levar algo a mais, tirando proveito de seus tão variados recursos.

REFERÊNCIAS

APPEL, K.; HAKEN, W. The Solution of the Four-Color-Map Problem. **Scientific American**, New York, v. 237, n. 4, p. 108-121, out. 1977. Disponível em:

<<https://www.scientificamerican.com/article/the-solution-of-the-four-color-map/>>. Acesso em: 25 ago. 2020.

ASTRÔNOMOS apresentam a primeira imagem de um buraco negro já registrada. **G1**, Rio de Janeiro, 10 abr. 2019. Disponível em:

<<https://g1.globo.com/ciencia-e-saude/noticia/2019/04/10/astronomos-apresentam-a-primeira-imagem-de-um-buraco-negro-ja-registrada.ghtml>>. Acesso em: 25 ago. 2020.

BORK, A. Interactive learning: Millikan Lecture, American Association of Physics Teachers, London, Ontario, June, 1978. **American Journal of Physics**. College Park, MD, v. 47, n. 1, 1979. Disponível em: <<https://aapt.scitation.org/doi/abs/10.1119/1.11684>>. Acesso em: 13 ago. 2020.

CORRÊA, P. S. Q. **Álgebra Linear e Geometria Analítica**. Rio de Janeiro: Interciência, 2006.

CORSATTO, C. A.; HOFFMANN, W. A. M. A Produção do Conhecimento Científica, Tecnológico e Organizacional no Contexto da Geração de Inovação e sua relação com a Ciência da Informação. In: ENCONTRO NACIONAL DE PESQUISA EM CIÊNCIA DA INFORMAÇÃO, 16., 2015, João Pessoa. **Anais...** João Pessoa: UFPB, 2015. Disponível em: <<http://www.ufpb.br/evento/index.php/enancib2015/enancib2015/paper/view/2804/1164>>. Acesso em: 3 ago. 2020.

DAMASIO, Felipe. O início da revolução científica: questões acerca de Copérnico e os epiciclos, Kepler e as órbitas elípticas. **Rev. Bras. Ensino Fís.**, São Paulo, v. 33, n. 3, p. 1-6, set. 2011. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172011000300020&lng=en&nrm=iso>. Acesso em: 12 out. 2020.
<http://dx.doi.org/10.1590/S1806-11172011000300020>.

FILHO, G. F. de S. **Simulações Computacionais para o Ensino de Física Básica: Uma Discussão sobre Produção e Uso**. Rio de Janeiro: UFRJ, 2010. Disponível em: <https://www.if.ufrj.br/~pef/producao_academica/dissertacoes/2010_Geraldo_Felipe/dissertacao_Geraldo_Felipe.pdf>. Acesso em: 27 ago. 2020.

FIOLHAIS, C.; TRINDADE, J. Física no computador: o computador como uma ferramenta no ensino e na aprendizagem das ciências físicas. **Rev. Bras. Ensino Fís.**, São Paulo, v. 25, n. 3, p. 259-272, Set. 2003. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172003000300002&lng=en&nrm=iso>. Acesso em: 13 ago. 2020.

JONG, T. de; ANDEL, J. van; LEIBLUM, M.; MIRANDE, M. Computer Assisted Learning in Higher Education in the Netherlands: a Review of Findings. **Computers Educ.**, Oxford, v. 19, n. 4, p. 381-386, 1992. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/036013159290085J?via%3Dihub>>. Acesso em: 15 ago. 2020.

MARIANI, V. C. Maple: **Fundamentos e Aplicações**. Rio de Janeiro: LTC, 2005.

MARTINS, R de A. A maçã de Newton: história, lendas e tolices. In: SILVA, C. C. (Org.). **Estudos de história e filosofia das ciências**: subsídios para aplicação no ensino. São Paulo: Livraria da Física, 2006. cap. 9.

MEDEIROS, A.; MEDEIROS, C. F. de. Possibilidades e Limitações das Simulações Computacionais no Ensino da Física. **Rev. Bras. Ensino Fís.**, São Paulo, v. 24, n. 2, p. 77-86, jun. 2002. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172002000200002&lng=en&nrm=iso>. Acesso em: 17 ago. 2020.

MOODY, E. A. Laws of Motion in Medieval Physics. **The Scientific Monthly**, New York, v. 72, n. 1, p. 18-23, jan. 1951. Disponível em: <<http://www.jstor.org/stable/19879>>. Acesso em: 12 out 2020.

MOURA, A. M. M. de; ROZADOS, H. B. F; CAREGNATO, S. E. Interações entre Ciência e Tecnologia: análise da produção intelectual dos pesquisadores-inventores da primeira cartapatente da UFRGS. **Enc. Bibli: R. eletrônica de Bibl. Ci. Inform.**, Florianópolis, n 22, 2º sem. 2006. Disponível em: <<https://periodicos.ufsc.br/index.php/eb/article/download/1518-2924.2006v11n22p1/360>>. Acesso em: 3 ago. 2020.

NASCIMENTO, M. L. F. et al. Laboratório virtual de Física: Gravitação. **Blucher Physics Proceedings**, v. 1, n. 2, p. 59-60, 2014. Disponível em: <https://www.researchgate.net/profile/Ramon_Prado/publication/301391932_Laboratorio_virtual_de_Fisica_Gravitacao/links/5846f0bb08aeda69682273d2/Laboratorio-virtual-de-Fisica-Gravitacao.pdf>. Acesso em: 20 ago. 2020.

PIRES, M. A.; VEIT, E. A. Tecnologias de Informação e Comunicação para ampliar e motivar o aprendizado de Física no Ensino Médio. **Rev. Bras. Ensino Fís.**, São Paulo, v. 28, n. 2, p. 241-248, jun. 2006. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1806-11172006000200015&lng=en&nrm=iso>. Acesso em: 20 ago. 2020.

ROY, A. E. **Orbital Motion**. 4 ed. Bristol: IoP, 2005. p. 4.

SILVA, H. C. da. **Discursos escolares sobre gravitação newtoniana**: textos e imagens na física do ensino médio. Campinas: [s.n.], 2002.

SWOKOWSKI, E. W.; FLORES, V. R. L. F.; MORENO, M. Q. **Cálculo de geometria analítica**. Tradução de Alfredo Alves de Faria. Revisão técnica Antonio Pertence Júnior. 2. ed. São Paulo: Makron, 1994. v. 2. c. 15.

UDEMY, 2020. Página inicial. Disponível em: <<https://www.udemy.com/>>. Acesso em: 27 ago. 2020.

VALDES-PEREZ, R. E. Some recent human-computer discoveries in science and what accounts for them. **AI Magazine**, Menlo Park, CA, v. 16, n. 3, p. 37, outono 1995. Disponível em: <<https://aaai.org/ojs/index.php/aimagazine/article/view/1147/1065>>. Acesso em: 25 ago. 2020.

WILLIAMS, R. D. **Moon Fact Sheet**. Washington, D.C.: NASA. 13 jan. 2020. Disponível em: <<https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html>>. Acesso em: 8 out. 2020.

ZOONIVERSE, 2020. Página inicial. Disponível em: <<https://www.zooniverse.org/>>. Acesso em: 27 ago. 2020.

APÊNDICE A – CÓDIGO DA ROTINA FORÇA GRAVITACIONAL

```

restart;
with(plots):
with(LinearAlgebra):
with(VectorCalculus, Vector, VectorSpace, PlotVector):
with(Statistics, Mean):
with(RandomTools, Generate):

G := 1: #Constante gravitacional
dp := 15: #Quantidade de dígitos das coordenadas dos pontos
dm := 5: #Quantidade de dígitos das massas dos pontos
nd := 2: #Número de dimensões
np := 4: #Número total de pontos
tmax := 4: #Coordenada máxima dos pontos
mmin := 1: #Massa mínima
mmax := 5: #Massa máxima
Vpre := []: #Posições predefinidas
Mpre := []: #Massas predefinidas (respectivas em relação às posições)
vt := 3: #Número da lista de pontos correspondente ao ponto de referência em relação ao qual
as forças e posições serão mostradas

#escolha do comando de texto dependendo do número de dimensões
if nd=2 or nd=1 then a:='pointplot';b:='textplot'
elif nd=3 then a:='pointplot3d';b:='textplot3d'fi:

#decisão da execução ou não dos comandos dependendo do número de dimensões
if nd<=3 then

#decisão da execução ou não dos comandos dependendo do número de pontos
if np>0 then

#geração de ponto de referência aleatório, caso não tenha sido escolhido
if vt = NULL or vt = 0 or vt > np then
vt := Generate(integer(range = 1 .. np)) end if:

#corte de pontos adicionais, caso tenham sido dados mais que o número de pontos escolhido
if numelems(Vpre) > np then
V := array(1 .. np, subs([0 = -0., 0. = -0.], [seq(Vpre[i], i = 1 .. np)])) else
V := array(1 .. np, subs([0 = -0., 0. = -0.], [op(Vpre)])) end if:

#geração de pontos aleatórios, caso tenham sido dados menos que o número de pontos
escolhidos
if numelems(Vpre) < np then
for i from numelems(Vpre)+1 to np do
r := Generate(list(float(range = -tmax .. tmax, digits = dp, method = uniform), nd));
while evalb(`in`(r, [seq(V[i], i = 1 .. np)])) = true do
r := Generate(list(float(range = -tmax .. tmax, digits = dp, method = uniform), nd)) end do:
if nd = 1 then V[i] := [r[1], 0] else
V[i] := r end if end do end if:

```

```

o := V[vt]:
V2 := [seq(convert(V[i],float,dp), i = 1 .. np)];

#corte de massas adicionais
if numelems(Mpre) > np then
M := array(1 .. np, [seq(Mpre[i], i = 1 .. np)]) else
M := array(1 .. np, [op(Mpre)]) end if:

#geração de massas aleatórias
if numelems(Mpre) < np then
for i from numelems(Mpre)+1 to np do
r := Generate(integer(range = 10^(dm-1) .. 10^dm-1))*10^Generate(integer(range =
trunc(log[10](mmin)) .. trunc(log[10](mmax))))/10^dm:
while `or`(r < mmin, r > mmax) do
r := Generate(integer(range = 10^(dm-1) .. 10^dm-1))*10^Generate(integer(range =
trunc(log[10](mmin)) .. trunc(log[10](mmax))))/10^(dm-1) end do:
M[i] := r end do end if:
mo := M[vt]:
M2 := [seq(convert(M[i],float,dm), i = 1 .. np)];

#exclusão do ponto de referência da lista de pontos (ele foi registrado como "o"). Isso é
necessário para o cálculo das forças
for i from vt to np-1 do
V[i] := V[i+1]:
M[i] := M[i+1] end do:
Vfix := array(1 .. np-1, [seq(V[i], i = 1 .. np-1)]):
Mfix := array(1 .. np-1, [seq(M[i], i = 1 .. np-1)]):

#cálculo das forças gravitacionais entre o ponto de referência e os demais
Vf := array(1 .. np, [seq(G*mo*Mfix[i]*Vector(Vfix[i]-o)/Norm(Vector(Vfix[i]-o))^3, i = 1 ..
np-1)]):

#cálculo da força gravitacional resultante
Vf[np] := add(Vf[i], i = 1 .. np-1):

#criação dos gráficos dos pontos
pp := seq(a(Vfix[i], symbol = solidcircle, color = brown, symbolsize = 10), i = 1 .. np-1):

#criação do gráfico do ponto de referência
ppo := a(o, symbol = solidcircle, color = green):

#escolha do ponto onde ficarão as origens dos vetores
Sv := VectorSpace(o):

#etiquetagem dos pontos, com base no número de pontos e dimensões
if `or`(`and`(nd = 1, np > 2), `and`(`or`(nd = 2, nd = 3), np > 8)) then
tp := NULL; vp := NULL else
tp := seq(b([op(V2[i]), i], 'align' = {'above', 'left'}, 'font' = ["times", 20]), i = 1 .. np):

#criação dos gráficos dos vetores posição

```

```

vp := seq(PlotVector(Sv:-Vector(Vfix[i]-o), color = blue), i = 1 .. np-1) end if;

#criação dos gráficos dos vetores força
vf := seq(PlotVector(Sv:-Vector(convert(Vf[i], list)), color = red), i = 1 .. np-1);

#criação do gráfico do vetor força resultante
if np <= 2 then vfr := NULL else vfr := PlotVector(Sv:-Vector(convert(Vf[np], list)), color =
"Orange") end if;

end if;

#exibição dos pontos
V2;M2;
end if;

#exibição dos gráficos
if np>0 and nd<=3 then
display(pp,tp,vp,ppo,scaling=constrained);
display(pp,vf,vfr,ppo,scaling=constrained);
elif nd<=3 then
plot([]);
elif nd>3 then textplot3d([0,0,0,"Por favor, defina um número de dimensões menor ou igual a
3"],'color'=blue);
end if;

```

APÊNDICE B – CÓDIGO DA ROTINA PROBLEMA DOS N-CORPOS

```

restart :
with(plots) :
with(VectorCalculus, Norm) :
with(Statistics, Mean) :
with(RandomTools, Generate) :

G := 6.674184·10-11 : #Constante gravitacional

nc := 4 : #Número de corpos (puntiformes) (somente para geração aleatória)
mmin := 1·1022 :
#Massa mínima (somente para geração aleatória), é necessário uma massa mínima porque
teoricamente a massa gerada poderia ser zero
mmax := 1·1025 : #Massa máxima (somente para geração aleatória)
rmax := 4·108 :
#Distância máxima em relação à origem dos eixos de coordenadas para as posições
iniciais dos corpos (geração aleatória)
vmax := 1·103 : #Velocidade inicial máxima para uma coordenada individual
tempo := 1000 : #quanto tempo dura a simulação em número de frames gerados
dimt := 5000 : # duração de um frame, em segundos
frameskip := 3 :
# para ver um movimento mais acelerado, porém o número de cálculos é multiplicado por
este número

Corpos := [ ] :
#Lista de corpos no formato [x, y, z], separados por vírgula e encapsulados por mais um
par de colchetes, onde x, y e z são as coordenadas cartesianas dos corpos
Velocidades := [ ] :
#Lista de velocidades iniciais no formato [vx, vy, vz] para os respectivos corpos da lista
anterior
Massas := [ ] :
#Lista de massas para os corpos indicados, caso se queira deixar em aberto somente as
posições

if numelems(Corpos) = 0 then
Corpos := array(1..nc, [ ] ) :
for i from 1 to nc do
Corpos[i] := Generate(list(float(range=-rmax..rmax), 3)) :
od:
fi:
if numelems(Velocidades) = 0 then
Velocidades := array(1..nc, [ ] ) :
for i from 1 to nc do
Velocidades[i] := Generate(list(float(range=-vmax..vmax), 3))
od:
fi:
if numelems(Massas) = 0 then
Massas := Generate(list(float(range = mmin ..mmax, method = logarithmic), nc)) :
fi:
op(Corpos) : op(Velocidades) : Massas :

Forçasx := Matrix(nc, nc, shape = antisymmetric) :
Forçasy := Matrix(nc, nc, shape = antisymmetric) :
Forçasz := Matrix(nc, nc, shape = antisymmetric) :
for i from 1 to nc - 1 do
for j from i + 1 to nc do
magn :=  $\frac{G \cdot \text{Massas}[i] \cdot \text{Massas}[j]}{(\text{Norm}(\text{op}(\text{Corpos}[i] - \text{Corpos}[j])))^3}$ ;
Forçasx[i, j] := magn·op(1, Corpos[i] - Corpos[j]);
# a coluna j da matriz contém todas as forças exercidas sobre o corpo j
Forçasy[i, j] := magn·op(2, Corpos[i] - Corpos[j]);
Forçasz[i, j] := magn·op(3, Corpos[i] - Corpos[j]);
od:
od:
Forças := [seq([add(Forçasx[i, j], i = 1..nc), add(Forçasy[i, j], i = 1..nc), add(Forçasz[i, j], i = 1..nc)], j = 1..nc)]:

frames[1] := display(pointplot3d(Corpos, color = ColorTools:-HueSpread("Red", nc), symbol = sphere)) :

for k from 2 to tempo do
for l from 1 to frameskip do
Velocidades := [seq(Velocidades[i] + dimt· $\frac{\text{Forças}[i]}{\text{Massas}[i]}$ , i = 1..nc)];
Corpos := [seq(Corpos[i] + dimt·Velocidades[i], i = 1..nc)];
for i from 1 to nc - 1 do
for j from i + 1 to nc do
magn :=  $\frac{G \cdot \text{Massas}[i] \cdot \text{Massas}[j]}{(\text{Norm}(\text{op}(\text{Corpos}[i] - \text{Corpos}[j])))^3}$ ;
Forçasx[i, j] := magn·op(1, Corpos[i] - Corpos[j]);
# a coluna j da matriz contém todas as forças exercidas sobre o corpo j
Forçasy[i, j] := magn·op(2, Corpos[i] - Corpos[j]);
Forçasz[i, j] := magn·op(3, Corpos[i] - Corpos[j]);
od:
od:
Forças := [seq([add(Forçasx[i, j], i = 1..nc), add(Forçasy[i, j], i = 1..nc), add(Forçasz[i, j], i = 1..nc)], j = 1..nc)]:
od:
frames[k] := display(pointplot3d(Corpos, color = ColorTools:-HueSpread("Red", nc), symbol = sphere)) :
od:
display(seq(frames[i], i = 1..tempo), axes = frame, scaling = constrained, labels = [x, y, z], insequence = true);

```