

UNIVERSIDADE DE BRASÍLIA  
INSTITUTO DE FÍSICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM FÍSICA

**Equações diferenciais funcionais dependentes do caminho:  
A dinâmica de Synge para duas cargas pontuais  
interagindo com seus campos eletromagnéticos**

Rodrigo Ribeiro da Silva  
*Orientador:* Prof. Dr. Annibal Dias de Figueiredo Neto

Brasília  
2022

## **Agradecimentos**

Agradeço, primeiramente, a Deus pela vida e pelas oportunidades que me foram apresentadas. Ao professor Annibal, pela orientação, paciência, disponibilidade de sanar minhas dúvidas e a motivação pelo trabalho desenvolvido. A minha mãe Lara, meu pai Luiz e a minha avó Cleuzeni que são os meus pilares e agradeço por tudo que eles já fizeram por mim, pois o que eu sou hoje é o reflexo do amor incondicional e da dedicação deles. A minha irmã Ana pelo carinho e pela sua alegria contagiante. A Lydiane pela amizade e incrível esposa que tenho. Amorosa e companheira em todos os momentos.

A Capes pelo apoio financeiro.

## Resumo

Os significados físicos e alguns aspectos matemáticos do problema proposto por Synge [1], será desenvolvido ao longo deste trabalho. O problema de Synge é determinar a dinâmica de duas cargas pontuais elétricas interagindo por meio de seus campos eletromagnéticos, sem levar em consideração os termos de radiação devidos às auto forças em cada carga pontual. Em particular, discutimos o problema de como considerar as condições iniciais compatíveis com um sistema isolado das forças externas. Este problema decorre da existência de restrições intertemporais para as trajetórias das cargas, gerando assim equações relativísticas de Newton para as cargas. Tais equações não formam um sistema de Equações Diferenciais Ordinárias, mas sim um sistema de Equações Diferenciais Funcionais, cuja dificuldade para a obtenção de soluções é bem maior, quando comparado com sistemas de Equações Diferenciais Ordinárias. A elaboração de um algoritmo, fundamentado apenas Equações Diferenciais Ordinárias para a obtenção de soluções aproximadas para o problema de Synge, foi desenvolvido e permitiu a construção de um algoritmo numérico utilizando métodos tradicionais de integração para sistemas de E.D.O.s. Finalmente, usamos este algoritmo para obter aproximações para as soluções quase circulares que são previstas no problema de Synge.

## Abstract

The physical meanings and some mathematical aspects of the problem proposed by Synge [1] will be developed throughout this work. Synge's problem is to determine the dynamics of two electric point charges interacting through their electromagnetic fields, without taking into account the radiation terms due to the self-forces on each point charge. In particular, we discuss the problem of how to consider initial conditions compatible with a system isolated from external forces. This problem stems from the existence of intertemporal restrictions for the trajectories of the charges, thus generating Newton's relativistic equations for the charges. Such equations do not form a system of Ordinary Differential Equations, but a system of Functional Differential Equations, whose difficulty to obtain solutions is much greater when compared to systems of Ordinary Differential Equations. The elaboration of an algorithm, based only on Ordinary Differential Equations to obtain approximate solutions to the Synge problem, was developed and allowed the construction of a numerical algorithm using traditional integration methods for ODEs systems. Finally, we use this algorithm to obtain approximations to the quasi-circular solutions that are predicted in Synge's problem.

# Sumário

## Sumário

## Lista de Figuras

## Lista de Tabelas

## Introdução

## I Campos eletromagnéticos retardados 10

1.1 Aproximação instantânea para as acelerações . . . . . 16

1.2 Aproximação para os tempos retardados . . . . . 17

## II Campos eletromagnéticos retardados e avançados 20

2.1 Aproximação instantânea para o campo simétrico . . . . . 21

2.2 A existência de soluções periódicas circulares . . . . . 22

## III Implementação do método iterativo para obter aproximações sucessivas das equações diferenciais instantâneas 25

3.1 Trajetórias circulares coulombianas . . . . . 26

3.1.1 Campos Eletromagnéticos Retardados . . . . . 26

3.1.2 Campos eletromagnéticos simétricos . . . . . 34

3.2 Trajetórias não circulares coulombianas . . . . . 37

3.2.1 Campos Eletromagnéticos Retardados . . . . . 37

3.2.2 Campos eletromagnéticos simétricos . . . . . 41

## IV Conclusão 45

4.1 Conclusões e considerações finais . . . . . 45

## Referências Bibliográficas 47

## Apêndice A Equações de movimento no eixo cartesiano 50

## Apêndice B Transformação De Coordenadas 76

2.1 Vetor do centro de massa e relativo . . . . . 76

2.2 Coordenadas polares . . . . . 79

## Apêndice C Transformação De Escala 84

3.1 Momento angular . . . . . 84

## Apêndice D Condições Iniciais 86

4.1 Condições iniciais para o sistema dinâmico . . . . . 86



# Lista de Figuras

3.1.1 Trajetórias planares em coordenadas cartesianas $(x, y)$ para os vetores $\mathbf{r}_1$ , $\mathbf{r}_2$ e $\mathbf{R}$ , onde integramos numericamente (para a mesma condição inicial) os quatro primeiros sistemas de equações diferenciais da sequência (1.2.47) considerando $\eta = 1$ . . . . .	27
3.1.2 Trajetórias planares em coordenadas cartesianas $(x, y)$ para os vetores $\mathbf{r}_1$ , $\mathbf{r}_2$ e $\mathbf{R}$ , onde integramos numericamente (para a mesma condição inicial) os quatro primeiros sistemas de equações diferenciais da sequência (1.2.47) considerando $\eta = 2$ . . . . .	28
3.1.3 Comparação dos erros da aproximação $S_3$ com relação a $S_0$ , $S_1$ e $S_2$ para $\eta$ igual a 1, 2 e 100 considerando o raio inicial igual a $r_0 = 50$ . . . . .	29
3.1.4 Comparação da trajetória relativa das partículas, obtido por integração numérica dos sistemas de equações diferenciais $S_0$ , $S_1$ , $S_2$ e $S_3$ para $\eta = 1$ . Na figura (3.1.4a) podemos observar a mudança da trajetória relativa nas sucessivas aproximações dadas por (1.2.47). Na figura (3.1.4b) o sistema $S_2$ e o sistema $S_3$ possuem trajetórias idênticas. Os vetores são mostrados em coordenadas cartesianas planas $(x, y)$ . . . . .	30
3.1.5 Comparação da trajetória relativa das partículas, obtido por integração numérica dos sistemas de equações diferenciais $S_0$ , $S_1$ , $S_2$ e $S_3$ para $\eta = 2$ . Como observado em (3.1.4) temos que na figura (3.1.5a) a mudança da trajetória relativa nas sucessivas aproximações dadas por (1.2.47). Na figura (3.1.5b) o sistema $S_2$ e o sistema $S_3$ possuem trajetórias idênticas. Os vetores são mostrados em coordenadas cartesianas planas $(x, y)$ . . . . .	30
3.1.6 Comparação da trajetória do centro de massa das partículas, obtido por integração numérica dos sistemas de equações diferenciais $S_0$ , $S_1$ , $S_2$ e $S_3$ para $\eta = 2$ . Assim como nos casos apresentados nas figuras (3.1.4) e (3.1.5), temos na figura (3.1.6a) a mudança da trajetória do centro de massa nas sucessivas aproximações dadas por (1.2.47). Na figura (3.1.6b) o sistema $S_2$ e o sistema $S_3$ possuem trajetórias idênticas. Os vetores são mostrados em coordenadas cartesianas planas $(x, y)$ . A trajetória do centro de massa para o caso $\eta = 1$ não é apresentado como exemplo, pois o centro de massa mantém-se parado. . . . .	31
3.1.7 Comparação da trajetória do centro de massa das partículas, obtido por integração numérica dos sistemas de equações diferenciais $S_0$ , $S_1$ , $S_2$ e $S_3$ para $\eta = 100$ . Os vetores são mostrados em coordenadas cartesianas planas $(x, y)$ . . . . .	31
3.1.8 Comparação das distâncias relativas ( $ r $ ), obtidas por integração numérica dos sistemas de equações diferenciais $S_0$ , $S_1$ , $S_2$ e $S_3$ para $\eta$ igual a 1, 2, e 1836. . . . .	32

3.1.9	Na figura à esquerda, mostramos a evolução da distância relativa entre as partículas $ r $ em função do tempo $t$ para diferentes valores de $\eta$ . Na figura à direita, plotamos o tempo de singularidade (obtido numericamente) em função de $\eta$ (Pontos) e seu respectivo ajuste linear (Linha preta tracejada).	33
3.1.10	Trajetórias relativas planares em coordenadas cartesianas $(x, y)$ para $\eta = 1$ e $\eta = 2$ , como mostrado nos gráficos (3.1.10a) e (3.1.10b), onde integramos numericamente (para a mesma condição inicial) os dois primeiros sistemas de equações diferenciais da sequencia (1.2.47). Nos gráficos (3.1.10c) e (3.1.10d) temos a comparação das distâncias relativas considerando os sistemas de equações diferenciais $S_0$ e $S_1$ .	34
3.1.11	Trajetória do centro de massa das partículas, obtido por integração numérica dos sistemas de equações diferenciais para $\eta = 2$ e $\eta = 100$ . Observamos que o padrão da trajetória muda conforme o valor de $\eta$ , porém as trajetórias parecem oscilar entre um valor máximo e um valor mínimo, nunca decaindo para uma singularidade. Os vetores são mostrados em coordenadas cartesianas planas $(x, y)$ . A trajetória do centro de massa para o caso $\eta = 1$ não é apresentada como exemplo, pois o centro de massa mantém-se parado.	35
3.1.12	Trajetórias planares em coordenadas cartesianas $(x, y)$ para os vetores $\mathbf{r}_1$ , $\mathbf{r}_2$ e $\mathbf{R}$ , onde integramos numericamente (para a mesma condição inicial) considerando $\eta = 2$ e $\eta = 100$ .	35
3.1.13	Comparação das distâncias relativas, obtidas por integração numérica dos sistemas de equações diferenciais considerando somente o campo retardado e o campo simétrico, para $\eta$ igual a 1, 2 e 100 considerando o raio inicial igual a $r_0 = 50$ .	36
3.1.14	Comparação das distâncias relativas, obtidas por integração numérica dos sistemas de equações diferenciais considerando somente o campo retardado e o campo simétrico, para $\eta$ igual a 2 e 1836 considerando o raio inicial igual a $r_0 = 18789$ .	36
3.2.15	Trajetórias planares em coordenadas cartesianas $(x, y)$ para os vetores $\mathbf{r}_1$ , $\mathbf{r}_2$ e $\mathbf{R}$ para o sistema $S_2$ da sequencia (1.2.47) com a condição inicial não circular considerando $\eta = 1$ , $\eta = 2$ e $\eta = 100$ .	38
3.2.16	Comparação da trajetória relativa das partículas, obtido por integração numérica dos sistemas de equações diferenciais $S_0$ , $S_1$ , $S_2$ e $S_3$ para $\eta = 2$ para uma condição inicial não circular.	38
3.2.17	Trajetória relativa das partículas para $\eta = 1$ , $\eta = 2$ e $\eta = 100$ para uma condição inicial não circular.	39
3.2.18	Trajetória do centro de massa das partículas para $\eta = 2$ e $\eta = 100$ para uma condição inicial circular.	40
3.2.19	Distância relativa para $\eta$ igual a 1, 2, e 100 para uma condição inicial não circular.	40
3.2.20	Trajetórias planares em coordenadas cartesianas $(x, y)$ para os vetores $\mathbf{r}_1$ , $\mathbf{r}_2$ e $\mathbf{R}$ para o sistema $S_1$ da sequencia (1.2.47) com a condição inicial não circular considerando $\eta = 1$ , $\eta = 2$ e $\eta = 100$ para o campo simétrico.	41
3.2.21	Trajetórias relativas para o sistema $S_1$ da sequencia (1.2.47) com a condição inicial não circular considerando $\eta = 1$ , $\eta = 2$ e $\eta = 100$ para o campo simétrico.	42
3.2.22	Trajetórias do centro de massa para o sistema $S_1$ da sequencia (1.2.47) com a condição inicial não circular considerando $\eta = 2$ e $\eta = 100$ para o campo simétrico.	43



3.2.23	Comparação das distâncias relativas, obtidas por integração numérica dos sistemas de equações diferenciais considerando somente o campo retardado e o campo simétrico, para $\eta$ igual a 1, 2, e 100 para uma condição inicial não circular. . . . .	43
3.2.24	Distâncias relativas, obtidas por integração numérica dos sistemas de equações diferenciais considerando o campo simétrico, para $\eta$ igual a 2 e 100 para uma condição inicial circular. . . . .	44

# Lista de Tabelas

E.1 Tabela dos parâmetros das simulações. O parâmetro  $\alpha$  indica a proporção entre os campos calculados no tempo de retardo e no tempo avançado. Dado o valor alfa igual a meio o campo é calculado somente no tempo de retardo e quando alfa é zero existe a contribuição dos dois campos de forma idêntica. O valor de  $\eta$  indica a razão entre as massas das partículas e  $r_0$  a distância inicial entre as partículas. O campo vetorial  $H^n$  indica em qual ordem de convergência  $n$  estamos simulando as equações. Os parâmetros  $n$  e  $m$ , trabalham juntos para obter os valores das posições, velocidades e acelerações calculados durante as simulações. Sendo assim,  $n$  indica a divisão do círculo em  $n$  partes e  $m$  é o número de círculos que desejamos calcular, por fim temos os tempos de simulação. . . . . 89

# Introdução

O problema de duas cargas pontuais, formulado como um problema de dois corpos isolados, consiste em determinar a dinâmica das cargas em função de seus próprios campos eletromagnéticos, ou seja, considerar o campo eletromagnético gerado por fontes externas as duas cargas consideradas. A primeira formulação rigorosa deste problema foi feita por Synge [1], nela ele não considera as forças de reação de radiação de cargas pontuais [2, 3, 4, 5]. Discutiremos esse assunto posteriormente nesta introdução, mas, por enquanto, todos os esforços serão voltados para a formulação matemática de Synge.

O problema matemático de Synge consiste em acoplar as equações de Maxwell para duas cargas pontuais com as equações da mecânica relativística usando a força de Lorentz para cada carga. Matematicamente, as equações relativísticas de Newton para duas cargas pontuais aceleradas são dadas por:

$$\frac{d\mathbf{p}_i}{dt} = \mathbf{F}_{ji}, \quad \mathbf{p}_i = \frac{m_i \mathbf{v}_i}{\sqrt{1 - |\mathbf{v}_i|^2/c^2}}, \quad i = 1, 2, \quad (0.0.1)$$

em que  $\mathbf{p}_i$  é o momento relativístico da partícula  $i$  e

$$\mathbf{F}_{ji} = q_i \mathbf{E}_{ji} + q_i \mathbf{v}_i \times \mathbf{B}_{ji} \quad (i \neq j), \quad (0.0.2)$$

é a força de Lorentz na carga pontual  $i$  devido ao campo eletromagnético da carga pontual  $j$  e  $\mathbf{v}_i$  é a velocidade da partícula  $i$ . Aqui estamos considerando a carga pontual  $q_i$  como uma partícula com massa inercial  $m_i$  ( $i = 1, 2$ ) e os campos eletromagnéticos devem satisfazer as equações de Maxwell para uma distribuição de duas cargas pontuais, o que corresponde a um sistema de Equações Diferenciais Parciais dado por:

$$\begin{aligned} \nabla \cdot \mathbf{B}(\mathbf{r}, t) &= 0, & \nabla \times \mathbf{E}(\mathbf{r}, t) + \frac{\partial \mathbf{B}(\mathbf{r}, t)}{\partial t} &= 0, \\ \nabla \cdot \mathbf{E}(\mathbf{r}, t) &= \frac{\rho(\mathbf{r}, t)}{\epsilon_0}, & \nabla \times \mathbf{B}(\mathbf{r}, t) - \frac{1}{c^2} \frac{\partial \mathbf{E}(\mathbf{r}, t)}{\partial t} &= \mu_0 \mathbf{j}(\mathbf{r}, t), \end{aligned} \quad (0.0.3)$$

onde  $\mathbf{r} \in \mathbb{R}^3$  representa um vetor de posição no espaço e  $t \in \mathbb{R}$  um instante de tempo. Este sistema de equações de Maxwell é escrito no sistema de unidades MKS, com  $\epsilon_0$ ,  $\mu_0$  e  $c$  sendo permissividade elétrica, a susceptibilidade magnética e a velocidade da luz no vácuo, respectivamente. As densidades de carga  $\rho(\mathbf{r}, t)$  e as densidades de corrente  $\mathbf{j}(\mathbf{r}, t)$  são definidas como:

$$\rho(\mathbf{r}, t) = \sum_{i=1,2} q_i \delta(\mathbf{r} - \mathbf{r}_i), \quad \mathbf{j}(\mathbf{r}, t) = \sum_{i=1,2} q_i \mathbf{v}_i \delta(\mathbf{r} - \mathbf{r}_i), \quad (0.0.4)$$

onde  $\delta$  é a função delta de Dirac tridimensional e  $\mathbf{r}_i$  é o vetor de posição associado à carga  $q_i$ . Finalmente, os campos eletromagnéticos na equação (0.0.2) são calculados na posição da carga  $\mathbf{r}_i$  e no tempo instantâneo  $t$ , ou seja,

$$\mathbf{E}_{ji} = \mathbf{E}(\mathbf{r}_i, t), \quad \mathbf{B}_{ji} = \mathbf{B}(\mathbf{r}_i, t) \quad (0.0.5)$$

O sistema de equações (0.0.1) - (0.0.5) representam a formulação básica do problema de duas cargas interagindo unicamente através de seus respectivos campos eletromagnéticos. Em uma primeira inspeção, isso constitui um problema matemático muito complexo de acoplamento de um sistema de E.D.O.s (equações de Newton) com um sistema de E.D.P.s (equações de Maxwell). Apesar de os problemas envolverem dificuldades técnicas oriundas das singularidades das densidades de carga, este é um problema bastante atípico do ponto de vista matemático e envolve questões muito sérias para estabelecer resultados para a existência de soluções através da imposição de alguma condição inicial [6], o que implicaria em conhecer as posições  $\mathbf{r}_i(t_0)$  e as velocidades  $\mathbf{v}_i(t_0)$  das partículas, bem como os campos  $\mathbf{E}(\mathbf{r}, t_0)$  e  $\mathbf{B}(\mathbf{r}, t_0)$  em um determinado instante  $t_0$ .

A origem desta dificuldade está relacionada ao fato de que o sistema composto pelas equações (0.0.1) - (0.0.5) tem uma restrição intertemporal. Essa restrição surge do fato de que as equações em (0.0.3) tem uma solução bem determinada, baseada nos potenciais de Liénard-Wiechert, para uma dada trajetória da carga pontual. Na verdade, os campos de Liénard-Wiechert retardados ou avançados são campos eletromagnéticos variáveis no tempo que satisfazem as equações de Maxwell para uma carga pontual em um movimento arbitrário descrito por seu vetor de posição  $\mathbf{r}_q(t)$  em função do tempo  $t$ .

Os respectivos campos elétricos e magnéticos retardados, calculados em alguma posição  $\mathbf{r}$  e no tempo  $t$ , são dados por [7, 8, 9]:

$$\mathbf{E}^-(\mathbf{r}, t) = \frac{q}{4\pi\epsilon_0} \left[ \frac{(1 - \beta^2)(\mathbf{n} - \boldsymbol{\beta})}{R^2|1 - \mathbf{n} \cdot \boldsymbol{\beta}|^3} + \frac{\mathbf{n} \times [(\mathbf{n} - \boldsymbol{\beta}) \times \mathbf{a}]}{c^2 R|1 - \mathbf{n} \cdot \boldsymbol{\beta}|^3} \right]_{t_r}, \quad (0.0.6)$$

$$\mathbf{B}^-(\mathbf{r}, t) = \frac{\mathbf{n}(t_r)}{c} \times \mathbf{E}^-(\mathbf{r}, t), \quad (0.0.7)$$

$$t_r = t - \frac{R}{c}, \quad (0.0.8)$$

onde  $\mathbf{R} = \mathbf{r} - \mathbf{r}_q(t)$ ,  $R = |\mathbf{R}|$ ,  $\mathbf{n} = \mathbf{R}/R$ ,  $\boldsymbol{\beta} = \mathbf{v}/c$ ,  $\mathbf{v} = d\mathbf{r}_q/dt$  e  $\mathbf{a} = d\mathbf{v}_q/dt$  são todos avaliados no tempo retardado  $t_r$ .

A derivação de campos avançados de Liénard-Wiechert pode ser obtida de maneira semelhante à derivação usual realizada em livros para campos retardados, entretanto, devemos considerar os campos elétricos e magnéticos avaliados no tempo avançado  $t_a$  [10]:

$$\mathbf{E}^+(\mathbf{r}, t) = \frac{q}{4\pi\epsilon_0} \left[ \frac{(1 - \beta^2)(\mathbf{n} + \boldsymbol{\beta})}{R^2|1 + \mathbf{n} \cdot \boldsymbol{\beta}|^3} + \frac{\mathbf{n} \times [(\mathbf{n} + \boldsymbol{\beta}) \times \mathbf{a}]}{c^2 R|1 + \mathbf{n} \cdot \boldsymbol{\beta}|^3} \right]_{t_a}, \quad (0.0.9)$$

$$\mathbf{B}^+(\mathbf{r}, t) = \frac{\mathbf{n}(t_a)}{c} \times \mathbf{E}^+(\mathbf{r}, t), \quad (0.0.10)$$

$$t_a = t + \frac{R}{c}. \quad (0.0.11)$$

Os campos elétrico e magnético da partícula  $j$  atuando na partícula  $i$  são definidos da seguinte forma:

$$\mathbf{E}_{ji} = \left( \frac{1}{2} + \alpha \right) \mathbf{E}_{ji}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{E}_{ji}^+, \quad (0.0.12)$$

$$\mathbf{B}_{ji} = \left( \frac{1}{2} + \alpha \right) \mathbf{B}_{ji}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{B}_{ji}^+, \quad (0.0.13)$$

com  $i \neq j$ . Os sobrescritos "-" e "+" indicam se o campo eletromagnético deve ser avaliado em tempos retardados ou avançados, respectivamente. O parâmetro  $\alpha$  é apenas um truque para considerar combinações lineares entre campos retardados e avançados que representam diferentes soluções das equações de Maxwell. Vale a pena citar dois casos:

o primeiro é o cenário causal usual com  $\alpha = 1/2$ , onde há apenas campo retardado; o segundo é  $\alpha = 0$ , que corresponde à formulação Wheeler-Feynman do eletromagnetismo onde há uma contribuição simétrica proveniente de campos retardados e avançados [11, 12].

Apesar do problema da causalidade não está resolvida e a necessidade do uso dos campos avançados, proposto por Wheeler e Feynman, para a interpretação dos absorvedores, podemos utilizar tal proposição de construção de um sistema de duas cargas pontuais para descrever o método iterativo que permite obter aproximações sucessivas, por meio de E.D.O.s, de um conjunto de equações que contém termos com derivadas dependentes dos tempos retardados e avançados. O ganho técnico associado a resolução deste problema parece superar as questões mais físicas associadas a questão de obedecer ou não a causalidade.

Os campos elétricos e magnéticos apresentados acima são dados por:

$$\mathbf{E}_{ji}^- = \frac{q_j}{4\pi\epsilon_0} \mathbf{G}_{ji}^-(\mathbf{r}_i, \mathbf{r}_j^-, \mathbf{v}_j^-, \mathbf{a}_j^-), \quad \mathbf{B}_{ji}^- = \frac{q_j}{4\pi\epsilon_0 c} \mathbf{n}_{ji}^- \times \mathbf{G}_{ji}^-(\mathbf{r}_i, \mathbf{r}_j^-, \mathbf{v}_j^-, \mathbf{a}_j^-), \quad (0.0.14)$$

$$\mathbf{E}_{ji}^+ = \frac{q_j}{4\pi\epsilon_0} \mathbf{G}_{ji}^+(\mathbf{r}_i, \mathbf{r}_j^+, \mathbf{v}_j^+, \mathbf{a}_j^+), \quad \mathbf{B}_{ji}^+ = \frac{q_j}{4\pi\epsilon_0 c} \mathbf{n}_{ji}^+ \times \mathbf{G}_{ji}^+(\mathbf{r}_i, \mathbf{r}_j^+, \mathbf{v}_j^+, \mathbf{a}_j^+), \quad (0.0.15)$$

onde

$$\mathbf{G}_{ji}^- = \frac{(1 - \beta_j^{-2})(\mathbf{n}_{ji}^- - \beta_j^-)}{|\mathbf{r}_i - \mathbf{r}_j^-|^2 |1 - \mathbf{n}_{ji}^- \cdot \beta_j^-|^3} + \frac{\mathbf{n}_{ji}^- \times [(\mathbf{n}_{ji}^- - \beta_j^-) \times \mathbf{a}_j^-]}{c^2 |\mathbf{r}_i - \mathbf{r}_j^-| |1 - \mathbf{n}_{ji}^- \cdot \beta_j^-|^3}, \quad (0.0.16)$$

$$\mathbf{G}_{ji}^+ = \frac{(1 - \beta_j^{+2})(\mathbf{n}_{ji}^+ + \beta_j^+)}{|\mathbf{r}_i - \mathbf{r}_j^+|^2 |1 + \mathbf{n}_{ji}^+ \cdot \beta_j^+|^3} + \frac{\mathbf{n}_{ji}^+ \times [(\mathbf{n}_{ji}^+ + \beta_j^+) \times \mathbf{a}_j^+]}{c^2 |\mathbf{r}_i - \mathbf{r}_j^+| |1 + \mathbf{n}_{ji}^+ \cdot \beta_j^+|^3}, \quad (0.0.17)$$

$$\mathbf{n}_{ji}^- = \frac{\mathbf{r}_i - \mathbf{r}_j^-}{|\mathbf{r}_i - \mathbf{r}_j^-|}, \quad (0.0.18)$$

$$\mathbf{n}_{ji}^+ = \frac{\mathbf{r}_i - \mathbf{r}_j^+}{|\mathbf{r}_i - \mathbf{r}_j^+|}, \quad (0.0.19)$$

com  $\beta_j^- = \mathbf{v}_j^-/c$  e  $\beta_j^+ = \mathbf{v}_j^+/c$ . As variáveis dinâmicas avaliadas em seus respectivos os tempos são definidos como:

$$\mathbf{r}_i = \mathbf{r}_i(t), \quad \mathbf{v}_i = \mathbf{v}_i(t), \quad \mathbf{a}_i = \mathbf{a}_i(t), \quad (0.0.20)$$

$$\mathbf{r}_j^- = \mathbf{r}_j(t_r^j), \quad \mathbf{v}_j^- = \mathbf{v}_j(t_r^j), \quad \mathbf{a}_j^- = \mathbf{a}_j(t_r^j), \quad (0.0.21)$$

$$\mathbf{r}_j^+ = \mathbf{r}_j(t_a^j), \quad \mathbf{v}_j^+ = \mathbf{v}_j(t_a^j), \quad \mathbf{a}_j^+ = \mathbf{a}_j(t_a^j), \quad (0.0.22)$$

e os tempos de retardo e avançado são obtidos das soluções das seguintes equações:

$$t_r^j = t - \frac{|\mathbf{r}_i - \mathbf{r}_j(t_r^j)|}{c}, \quad t_a^j = t + \frac{|\mathbf{r}_i - \mathbf{r}_j(t_a^j)|}{c}. \quad (0.0.23)$$

Realizando uma transformação de escala nas unidades de espaço e de tempo definidas como:

$$\mathbf{r}_i \rightarrow L\mathbf{r}_i \quad (i = 1, 2) \quad \text{e} \quad t \rightarrow Tt, \quad (0.0.24)$$

a mesma transformação leva às seguintes transformações nas velocidades e acelerações:

$$\mathbf{v}_i \rightarrow c\mathbf{v}_i, \quad \mathbf{a}_i \rightarrow \frac{c}{T}\mathbf{a}_i, \quad (0.0.25)$$

e cabe ainda a seguinte definição:

$$\frac{L}{T} \equiv c. \quad (0.0.26)$$

A relação acima implica  $c = 1$  nas novas unidades de escala de espaço e tempo. Para realizar o reescalonamento no sistema (0.0.2) veremos inicialmente como ocorre a transformação de escala no termo  $\mathbf{G}_{ji}^-$ , pois uma vez que entendemos como se dá a transformação deste termo podemos aplicar em  $\mathbf{G}_{ji}^+$  e realizar o reescalonamento de todos os termos do sistema de equações em (0.0.2). Sendo assim, temos:

$$\mathbf{G}_{ji}^+ = \frac{\left(1 - |\mathbf{v}_j^-|^2/c^2\right) (\mathbf{n}_{ji}^- - \mathbf{v}_j^-/c)}{|\mathbf{r}_i - \mathbf{r}_j^-|^2 |1 - \mathbf{n}_{ji}^- \cdot \mathbf{v}_j^-/c|^3} + \frac{\mathbf{n}_{ji}^- \times [(\mathbf{n}_{ji}^- - \mathbf{v}_j^-/c) \times \mathbf{a}_j^-]}{c^2 |\mathbf{r}_i - \mathbf{r}_j^-| |1 - \mathbf{n}_{ji}^- \cdot \mathbf{v}_j^-/c|^3}.$$

Aplicando a transformação de escala obtemos:

$$\begin{aligned} & \frac{\left(1 - \frac{c^2 |\mathbf{v}_j^-|^2}{c^2}\right) \left(\mathbf{n}_{ji}^- - \frac{c\mathbf{v}_j^-}{c}\right)}{L^2 |\mathbf{r}_i - \mathbf{r}_j^-|^2 |1 - \mathbf{n}_{ji}^- \cdot \frac{c\mathbf{v}_j^-}{c}|^3} + \frac{\mathbf{n}_{ji}^- \times \left[\left(\mathbf{n}_{ji}^- - \frac{c\mathbf{v}_j^-}{c}\right) \times \frac{c\mathbf{a}_j^-}{T}\right]}{c^2 L |\mathbf{r}_i - \mathbf{r}_j^-| |1 - \mathbf{n}_{ji}^- \cdot \frac{c\mathbf{v}_j^-}{c}|^3}, \\ & \frac{\left(1 - |\mathbf{v}_j^-|^2\right) (\mathbf{n}_{ji}^- - \mathbf{v}_j^-)}{L^2 |\mathbf{r}_j - \mathbf{r}_j^-|^2 |1 - \mathbf{n}_{ji}^- \cdot \mathbf{v}_j^-|^3} + \frac{\mathbf{n}_{ji}^- \times [(\mathbf{n}_{ji}^- - \mathbf{v}_j^-) \times \mathbf{a}_j^-]}{cLT |\mathbf{r}_i - \mathbf{r}_j^-| |1 - \mathbf{n}_{ji}^- \cdot \mathbf{v}_j^-|^3}, \\ & \frac{\left(1 - |\mathbf{v}_j^-|^2\right) (\mathbf{n}_{ji}^- - \mathbf{v}_j^-)}{L^2 |\mathbf{r}_j - \mathbf{r}_j^-|^2 |1 - \mathbf{n}_{ji}^- \cdot \mathbf{v}_j^-|^3} + \frac{\mathbf{n}_{ji}^- \times [(\mathbf{n}_{ji}^- - \mathbf{v}_j^-) \times \mathbf{a}_j^-]}{L^2 |\mathbf{r}_i - \mathbf{r}_j^-| |1 - \mathbf{n}_{ji}^- \cdot \mathbf{v}_j^-|^3}, \\ & \frac{1}{L^2} \left[ \frac{\left(1 - |\mathbf{v}_j^-|^2\right) (\mathbf{n}_{ji}^- - \mathbf{v}_j^-)}{|\mathbf{r}_i - \mathbf{r}_j^-|^2 |1 - \mathbf{n}_{ji}^- \cdot \mathbf{v}_j^-|^3} + \frac{\mathbf{n}_{ji}^- \times [(\mathbf{n}_{ji}^- - \mathbf{v}_j^-) \times \mathbf{a}_j^-]}{|\mathbf{r}_i - \mathbf{r}_j^-| |1 - \mathbf{n}_{ji}^- \cdot \mathbf{v}_j^-|^3} \right]. \end{aligned}$$

Portanto o reescalonamento do termo  $\mathbf{G}_{ij}^-$  é:

$$\mathbf{G}_{ji}^- \rightarrow \frac{1}{L^2} \mathbf{G}_{ji}^-, \quad (0.0.27)$$

e consequentemente:

$$\mathbf{G}_{ji}^+ \rightarrow \frac{1}{L^2} \mathbf{G}_{ji}^+. \quad (0.0.28)$$

Aplicando tais transformações à equação (0.0.1), segue que:

$$\begin{aligned} \frac{c}{T} m_i \frac{d(\gamma_i \mathbf{v}_i)}{dt} &= \frac{q_i q_j}{4\pi\epsilon_0 L^2} \left[ \left(\frac{1}{2} + \alpha\right) \mathbf{G}_{ji}^- + \left(\frac{1}{2} - \alpha\right) \mathbf{G}_{ji}^+ \right] \\ &+ \frac{q_i q_j}{4\pi\epsilon_0 L^2} \mathbf{v}_i \times \left[ \left(\frac{1}{2} + \alpha\right) \mathbf{n}_{ji}^- \times \mathbf{G}_{ji}^- + \left(\frac{1}{2} - \alpha\right) \mathbf{n}_{ji}^+ \times \mathbf{G}_{ji}^+ \right]. \end{aligned} \quad (0.0.29)$$

Utilizando a equação (0.0.29) acima podemos escrever a equação reescalada para a

partícula 1 e 2 da seguinte maneira:

$$\begin{aligned} \frac{c}{T} m_1 \frac{d(\gamma_1 \mathbf{v}_1)}{dt} &= \frac{q_1 q_2}{4\pi\epsilon_0 L^2} \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{G}_{21}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{G}_{21}^+ \right] \\ &\quad + \frac{q_1 q_2}{4\pi\epsilon_0 L^2} \mathbf{v}_1 \times \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{n}_{21}^- \times \mathbf{G}_{21}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{n}_{21}^+ \times \mathbf{G}_{21}^+ \right], \\ \frac{c}{T} m_2 \frac{d(\gamma_2 \mathbf{v}_2)}{dt} &= \frac{q_2 q_1}{4\pi\epsilon_0 L^2} \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{G}_{12}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{G}_{12}^+ \right] \\ &\quad + \frac{q_2 q_1}{4\pi\epsilon_0 L^2} \mathbf{v}_2 \times \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{n}_{12}^- \times \mathbf{G}_{12}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{n}_{12}^+ \times \mathbf{G}_{12}^+ \right]. \end{aligned}$$

Multiplicando ambas as equações por  $T/cm_2$  e definindo  $\eta = m_1/m_2$ , temos:

$$\begin{aligned} \eta \frac{d(\gamma_1 \mathbf{v}_1)}{dt} &= \frac{T q_1 q_2}{cm_2 4\pi\epsilon_0 L^2} \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{G}_{21}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{G}_{21}^+ \right] \\ &\quad + \frac{T q_1 q_2}{cm_2 4\pi\epsilon_0 L^2} \mathbf{v}_1 \times \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{n}_{21}^- \times \mathbf{G}_{21}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{n}_{21}^+ \times \mathbf{G}_{21}^+ \right], \\ \frac{d(\gamma_2 \mathbf{v}_2)}{dt} &= \frac{T q_2 q_1}{cm_2 4\pi\epsilon_0 L^2} \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{G}_{12}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{G}_{12}^+ \right] \\ &\quad + \frac{T q_2 q_1}{cm_2 4\pi\epsilon_0 L^2} \mathbf{v}_2 \times \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{n}_{12}^- \times \mathbf{G}_{12}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{n}_{12}^+ \times \mathbf{G}_{12}^+ \right]. \end{aligned}$$

Para simplificar as expressões acima, faremos:

$$\frac{T}{cm_2} \frac{|q_1||q_2|}{4\pi\epsilon_0} \frac{1}{L^2} = 1,$$

o que nos leva as seguintes relações:

$$L = \frac{|q_1||q_2|}{4\pi\epsilon_0 m_2 c^2}, \quad T = \frac{|q_1||q_2|}{4\pi\epsilon_0 m_2 c^3}. \quad (0.0.30)$$

Definimos também o parâmetro  $S$  como:

$$S = \text{sgn}(q_1 q_2), \quad (0.0.31)$$

que pode assumir o valor 1 para o sistema repulsivo e  $-1$  para o sistema atrativo. Assim, o sistema (0.0.29) pode ser reescrito como:

$$\begin{aligned} \frac{d\mathbf{p}_1}{dt} = \eta \frac{d(\gamma_1 \mathbf{v}_1)}{dt} &= S \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{G}_{21}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{G}_{21}^+ \right] \\ &\quad + S \mathbf{v}_1 \times \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{n}_{21}^- \times \mathbf{G}_{21}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{n}_{21}^+ \times \mathbf{G}_{21}^+ \right], \end{aligned} \quad (0.0.32)$$

$$\begin{aligned} \frac{d\mathbf{p}_2}{dt} = \frac{d(\gamma_2 \mathbf{v}_2)}{dt} &= S \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{G}_{12}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{G}_{12}^+ \right] \\ &\quad + S \mathbf{v}_2 \times \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{n}_{12}^- \times \mathbf{G}_{12}^- + \left( \frac{1}{2} - \alpha \right) \mathbf{n}_{12}^+ \times \mathbf{G}_{12}^+ \right], \end{aligned} \quad (0.0.33)$$

onde,

$$\gamma_1 = \frac{1}{\sqrt{1 - \mathbf{v}_1 \cdot \mathbf{v}_1}}, \quad \gamma_2 = \frac{1}{\sqrt{1 - \mathbf{v}_2 \cdot \mathbf{v}_2}}. \quad (0.0.34)$$

Destacamos que o conjunto de equações (0.0.32) e (0.0.33) reescalados tem como parâmetros os termos  $\eta$ ,  $S$  e  $\alpha$ . Para o sistema de interesse, que é o atrativo, temos  $S = -1$  portanto, os parâmetros que importam para as análises que serão realizadas no decorrer desta tese são: A razão entre as massas das partículas,  $\eta$ , e a proporção entre as forças calculadas no tempo de retardo e no tempo avançado,  $\alpha$ .

O procedimento de reparametrização do sistema por reescalamento, das variáveis espaço e tempo, gera para todo o sistema de duas cargas um parâmetro bastante importante que é o valor de  $\eta$ . O motivo do parâmetro em questão ser tão importante é o fato de que o problema será o mesmo se a razão entre as massas for o mesmo. Portanto, se consiserarmos dois sistemas distintos, porém se ambos mantém as mesmas razões entre as massas toda a dinâmica, nas unidades reescaladas, será igual. As diferenças entre os dois sistemas será observada quando considerarmos as escalas originais, pois iremos considerar diferentes valores para as massas e as cargas.

Considerando apenas os aspectos matemáticos, podemos tomar a forma mais geral da restrição intertemporal imposta pelos campos de Lienard-Wiechert e substituir suas expressões explícitas na equação de força de Lorentz (0.0.2) e então colocar essas forças na equação de Newton (0.0.1) para finalmente obter as equações (0.0.32, 0.0.33). As equações obtidas não formam mais um sistema de E.D.O.s, mas sim, como é conhecido na literatura, um sistema de equações diferenciais funcionais (E.D.F.s) [13, 14, 15]. Este sistema de E.D.F.s consiste em equações diferenciais, equações (0.0.32, 0.0.33), e funcionais como as equações (0.0.23), que relacionam as derivadas de segunda ordem (acelerações) com suas derivadas de primeira ordem (velocidades) e segunda ordem em diferentes tempos da trajetória (restrição intertemporal). Podemos considerar este sistema de E.D.F.s como sendo a generalização natural do problema originalmente formulado por Synge para  $\alpha = 1/2$ . Aqui, a obtenção das soluções da Equação de Newton dada pelas equações (0.0.32, 0.0.33) será chamado de problema de Synge, independentemente do valor de  $\alpha$ .

Em seu trabalho [1], Synge alerta para a consistência matemática do problema proposto, mostrando que esta consistência matemática não está correlacionada com as questões sobre conservação de energia e momento para o sistema eletromagnético completo constituído por cargas e campos. Essas questões dependem da definição de como calcular os momentos e a energia de um campo eletromagnético e, do ponto de vista matemático, são irrelevantes para a definição do Problema de Synge. A história do progresso matemático na caracterização de soluções para o problema Synge, que está intimamente ligada ao desenvolvimento de teoremas de existência (e muitas vezes unicidade) em sistemas E.D.F.s neutros, confirma seu alerta. [16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31]

Uma questão importante que deve ser apontada aqui é que Synge considera o problema das cargas pontuais sem levar em conta as forças de reação decorrentes da auto interação da cada carga sobre si mesma, chamadas de forças de radiação e obtidas originalmente por Lorentz e Abraham [2, 3]. Para obter esta auto-força eles consideraram as forças entre as partes de uma pequena esfera rígida uniformemente carregada, levando o limite do volume desta esfera a zero e mantendo a quantidade total de carga. Este procedimento é hoje conhecido como renormalização e foi obtido sem o uso da ideia de uma carga rígida distribuída em uma pequena região [4, 5].

Do ponto de vista matemático, o problema de duas cargas pontuais levando em consideração a auto-interação em cada carga pode ser formulado da mesma forma que o problema de Synge, mas devemos levar em consideração os termos que corrigem as equações de Newton e levam ao aparecimento de derivados de terceira ordem das posições das partículas. Nesse caso, o sistema de E.D.F.s resultante seria um sistema diferencial não neutro de terceira ordem. Atualmente, essas equações E.D.F.s de terceira ordem são chamadas de equações de Dirac-Lorentz. No processo de renormalização, considerando



$\alpha = 0$  (campos simétricos), as auto-forças associadas aos tempos retardados se cancelam com aquelas associadas aos tempos avançados e nenhuma correção é necessária para o problema de Synge. Nesse caso, essas equações E.D.F.s de segunda ordem são chamadas de equações de Wheeler-Feynman. Uma boa discussão crítica sobre as equações de Dirac-Lorentz versus Wheeler-Feynman foi feita por Havas [32].

Além disso, no que diz respeito ao trabalho do Synge [1], vale lembrar que ele desenvolve um método iterativo autoconsistente que nos permitiria obter soluções para o problema para uma dada condição inicial. Essencialmente, este método consiste em considerar uma dada trajetória de uma carga e através de seus conhecidos campos de Liénard-Wiechert para resolver a equação de Newton associada a outra carga. A partir dessa trajetória obtida para a outra partícula, repetimos o procedimento e, a seguir, calculamos a trajetória da primeira. Este procedimento deve ser repetido indefinidamente e deve convergir para a solução do problema. A convergência desse método exigiria que as massas das cargas pontuais fossem diferentes e a velocidade de convergência dependeria da razão  $\eta$  entre a maior e a menor massa, sendo mais rápida quanto maior essa razão. Como trajetória inicial, Synge considera o problema Kepleriano (analisado por Sommerfeld [33]) onde uma das massas das partículas é considerada infinita e portanto temos um sistema E.D.O.s bem definido. Da segunda etapa em diante, um valor finito para  $\eta$  deve ser considerado.

Infelizmente, Synge não demonstrou a convergência de seu método, o que foi sugerido apenas heurísticamente. É interessante notar que este método de aproximações de Synge só foi usado com sucesso para o caso em que  $\eta = 1$  [21, 24, 34]. Nas poucas simulações numéricas desenvolvidas, seu método não converge, mesmo para velocidades muito pequenas, para cargas opostas em órbitas quase circulares [22]. Além disso, é no caso atrativo que Synge desenvolve fórmulas analíticas aproximadas (que dependem de  $\eta$  ser muito grande) para obter a descrição de trajetórias singulares quase circulares, ou seja, trajetórias espirais de duas cargas que colidem após um tempo finito. O interessante aqui é que a colisão das duas cargas não é consequência de forças de radiação, pois, como mencionado acima, estas estão ausentes na formulação do problema de Synge.

O trabalho de Angelov critica o método iterativo do Synge mostrando que ele não poderia ser implementado após a primeira etapa [35]. Essa crítica parece apontar mais para a questão da trajetória inicial, escolhida de um problema kepleriano com  $\eta = \infty$  e sua posterior passagem para um sistema com  $\eta$  finito, do que uma crítica geral ao método independente da trajetória inicial escolhida. Este fato é bem demonstrado para os casos unidimensionais onde o método de aproximação Synge foi usado com sucesso.

Cabe também destacar que foram realizadas poucas simulações numéricas para o problema de Synge: seja por meio de implementações de seu algoritmo [22, 34] ou por meio da implementação de outro tipo de algoritmos de integração numérica para sistemas de E.D.F.s [36, 37]. Este fato é bastante contrastante com a abundância de métodos numéricos desenvolvidos para calcular o campo de radiação gerado pela trajetória de uma carga pontual. Na verdade, este problema é muito importante na análise da radiação de uma carga em um acelerador de partículas [38, 39] e ao longo deste trabalho, procuramos desenvolver um método para a obtenção numérica do problema de Synge. O método desenvolvido é baseado em aproximações sucessivas do sistema de E.D.F.s de Synge por sistemas de E.D.O.s. não tendo a intenção de apresentar provas rigorosas do método, mas sim ilustrar sua praticidade e facilidade com que ele pode ser programado para obter trajetórias aproximadas para as soluções do problema de Synge.

Por fim, segue um breve comentário acerca da divisão do trabalho. Na seção I discutimos alguns aspectos técnicos e os avanços obtidos na literatura sobre o problema Synge para campos retardados ( $\alpha = 1/2$ ). Apresentamos um método iterativo que permite

obter aproximações sucessivas do problema de Synge a partir do sistemas de E.D.O.s. Já na Seção II, discutimos o problema geral de Synge com campos mistos ( $1/2 < \alpha < -1/2$ ). Ao longo da Seção III ilustramos algumas soluções numéricas obtidas a partir dos métodos desenvolvidos na seção I. A seção IV é dedicada às considerações finais.

# Capítulo I

## Campos eletromagnéticos retardados

Nosso objetivo aqui é obter um sistema dinâmico de segunda ordem, explícito e a partir das equações (0.0.32, 0.0.33) para o caso particular  $\alpha = 1/2$  é o escopo deste capítulo. O procedimento permite considerar apenas os campos eletromagnéticos retardados agindo sobre as cargas pontuais. Primeiro, vamos considerar novamente os tempos retardados  $t_r^1$  e  $t_r^2$ , que são respectivamente soluções das seguintes equações:

$$t_r^1 = t - |\mathbf{r}_2(t) - \mathbf{r}_1(t_r^1)|, \quad t_r^2 = t - |\mathbf{r}_1(t) - \mathbf{r}_2(t_r^2)|. \quad (1.0.1)$$

Então, as equações (0.0.32, 0.0.33) podem ser reescritas em uma forma matricial como o seguinte sistema dinâmico de segunda ordem:

$$\frac{d\mathbf{p}_1}{dt} = \eta \mathcal{M}_{11} \mathbf{a}_1 = \mathbf{F}_1^- - \mathcal{M}_{12}^- \mathbf{a}_2^-, \quad (1.0.2)$$

$$\frac{d\mathbf{p}_2}{dt} = \mathcal{M}_{22} \mathbf{a}_2 = \mathbf{F}_2^- - \mathcal{M}_{21}^- \mathbf{a}_1^-, \quad (1.0.3)$$

onde  $\mathbf{a}_j = \mathbf{a}_j(t)$ ,  $\mathbf{a}_j^- = \mathbf{a}_j^-(t_r^j)$  ( $j = 1, 2$ ) e os vetores e matrizes com sobrescritos "-" dependem da posição e das velocidades que precisam ser avaliadas em alguns dos tempos retardados. De fato, o vetor  $\mathbf{F}_j^-$  e as matrizes  $\mathcal{M}_{ij}^-$  com  $i = 1, 2$  ( $j \neq i$ ) depende de  $\mathbf{r}_i(t)$ ,  $\mathbf{v}_i(t)$ ,  $\mathbf{v}_j(t_r^j)$  e  $\mathbf{v}_j(t_r^j)$ .

As forças  $\mathbf{F}_1^-$  e  $\mathbf{F}_2^-$  do lado direito das igualdades são partes das forças eletromagnéticas independentes das acelerações e podem ser escritas da seguinte forma:

$$\mathbf{F}_1^- = \frac{S\gamma_2^{-2}}{\lambda_2^- |\mathbf{r}_2^- - \mathbf{r}_1|^2} \begin{pmatrix} v_{1y} (v_{2y} e_{2x}^- - v_{2x} e_{2y}^-) + v_{1z} (v_{2z} e_{2x}^- - v_{2x} e_{2z}^-) - v_{2x}^- - e_{2x}^- \\ v_{1x} (v_{2x} e_{2y}^- - v_{2y} e_{2x}^-) + v_{1z} (v_{2z} e_{2y}^- - v_{2y} e_{2z}^-) - v_{2y}^- - e_{2y}^- \\ v_{1x} (v_{2x} e_{2z}^- - v_{2z} e_{2x}^-) + v_{1y} (v_{2y} e_{2z}^- - v_{2z} e_{2y}^-) - v_{2z}^- - e_{2z}^- \end{pmatrix}, \quad (1.0.4)$$

$$\mathbf{F}_2^- = \frac{S\gamma_1^{-2}}{\lambda_1^- |\mathbf{r}_2^- - \mathbf{r}_1|^2} \begin{pmatrix} v_{2y} (v_{1y} e_{1x}^- - v_{1x} e_{1y}^-) + v_{2z} (v_{1z} e_{1x}^- - v_{1x} e_{1z}^-) + v_{1x}^- - e_{1x}^- \\ v_{2x} (v_{1x} e_{1y}^- - v_{1y} e_{1x}^-) + v_{2z} (v_{1z} e_{1y}^- - v_{1y} e_{1z}^-) + v_{1y}^- - e_{1y}^- \\ v_{2x} (v_{1x} e_{1z}^- - v_{1z} e_{1x}^-) + v_{2y} (v_{1y} e_{1z}^- - v_{1z} e_{1y}^-) + v_{1z}^- - e_{1z}^- \end{pmatrix}, \quad (1.0.5)$$

em que  $\gamma_1^-$ ,  $\gamma_2^-$ ,  $\lambda_1^-$ ,  $\lambda_2^-$ ,  $\mathbf{e}_1^-$  e  $\mathbf{e}_2^-$  são dados por:

$$\gamma_1^- = \frac{1}{\sqrt{1 - \mathbf{v}_1^- \cdot \mathbf{v}_1^-}}, \quad \gamma_2^- = \frac{1}{\sqrt{1 - \mathbf{v}_2^- \cdot \mathbf{v}_2^-}}, \quad (1.0.6)$$

$$\lambda_1^- = (\mathbf{v}_1^- \cdot \mathbf{e}_1^- - 1)^3, \quad \lambda_2^- = (\mathbf{v}_2^- \cdot \mathbf{e}_2^- + 1)^3, \quad (1.0.7)$$

$$\mathbf{e}_1^- = \frac{\mathbf{r}_2 - \mathbf{r}_1^-}{|\mathbf{r}_2 - \mathbf{r}_1^-|} = \begin{pmatrix} e_{1x}^- \\ e_{1y}^- \\ e_{1z}^- \end{pmatrix}, \quad \mathbf{e}_2^- = \frac{\mathbf{r}_2^- - \mathbf{r}_1}{|\mathbf{r}_2^- - \mathbf{r}_1|} = \begin{pmatrix} e_{2x}^- \\ e_{2y}^- \\ e_{2z}^- \end{pmatrix}. \quad (1.0.8)$$

As matrizes  $\mathcal{M}_{11}$  e  $\mathcal{M}_{22}$ , avaliadas no tempo  $t$ , são definidas da seguinte maneira:

$$\mathcal{M}_{11} = \gamma_1 \mathcal{I} + \gamma_1^3 \mathcal{Q}_{\hat{v}_1}, \quad \mathcal{Q}_{\hat{v}_1} = \begin{pmatrix} \hat{v}_{1x}^2 & \hat{v}_{1x}\hat{v}_{1y} & \hat{v}_{1x}\hat{v}_{1z} \\ \hat{v}_{1y}\hat{v}_{1x} & \hat{v}_{1y}^2 & \hat{v}_{1y}\hat{v}_{1z} \\ \hat{v}_{1z}\hat{v}_{1x} & \hat{v}_{1z}\hat{v}_{1y} & \hat{v}_{1z}^2 \end{pmatrix}, \quad (1.0.9)$$

$$\mathcal{M}_{22} = \gamma_2 \mathcal{I} + \gamma_2^3 \mathcal{Q}_{\hat{v}_2}, \quad \mathcal{Q}_{\hat{v}_2} = \begin{pmatrix} \hat{v}_{2x}^2 & \hat{v}_{2x}\hat{v}_{2y} & \hat{v}_{2x}\hat{v}_{2z} \\ \hat{v}_{2y}\hat{v}_{2x} & \hat{v}_{2y}^2 & \hat{v}_{2y}\hat{v}_{2z} \\ \hat{v}_{2z}\hat{v}_{2x} & \hat{v}_{2z}\hat{v}_{2y} & \hat{v}_{2z}^2 \end{pmatrix}, \quad (1.0.10)$$

onde

$$\hat{v}_1 = \frac{\mathbf{v}_1}{|\mathbf{v}_1|}, \quad \hat{v}_2 = \frac{\mathbf{v}_2}{|\mathbf{v}_2|}. \quad (1.0.11)$$

Já as matrizes  $\mathcal{M}_{12}^-$  e  $\mathcal{M}_{21}^-$ , que possuem quantidades físicas calculadas no tempo de retardo, são definidas da seguinte maneira:

$$\mathcal{M}_{12}^- = \frac{S}{\lambda_2^- |\mathbf{r}_2^- - \mathbf{r}_1^-|} L_{12}^-, \quad \mathcal{M}_{21}^- = \frac{S}{\lambda_1^- |\mathbf{r}_2^- - \mathbf{r}_1^-|} L_{21}^-, \quad (1.0.12)$$

com os elementos da matriz  $L_{12}^-$  dados por:

$$\begin{aligned} (L_{12}^-)_{11} &= (v_{2z}^- e_{2y}^- - v_{2y}^- e_{2z}^-) (v_{1y}^- e_{2z}^- - v_{1z}^- e_{2y}^-) + (v_{2y}^- + e_{2y}^-) (v_{1y}^- + e_{2y}^-) + (v_{2z}^- + e_{2z}^-) (v_{1z}^- + e_{2z}^-), \\ (L_{12}^-)_{12} &= (v_{2x}^- e_{2z}^- - v_{2z}^- e_{2x}^-) (v_{1y}^- e_{2z}^- - v_{1z}^- e_{2y}^-) - (v_{2x}^- + e_{2x}^-) (v_{1y}^- + e_{2y}^-), \\ (L_{12}^-)_{13} &= (v_{2y}^- e_{2x}^- - v_{2x}^- e_{2y}^-) (v_{1y}^- e_{2z}^- - v_{1z}^- e_{2y}^-) - (v_{2x}^- + e_{2x}^-) (v_{1z}^- + e_{2z}^-), \\ (L_{12}^-)_{21} &= (v_{2y}^- e_{2z}^- - v_{2z}^- e_{2y}^-) (v_{1x}^- e_{2z}^- - v_{1z}^- e_{2x}^-) - (v_{2y}^- + e_{2y}^-) (v_{1x}^- + e_{2x}^-), \\ (L_{12}^-)_{22} &= (v_{2z}^- e_{2x}^- - v_{2x}^- e_{2z}^-) (v_{1x}^- e_{2z}^- - v_{1z}^- e_{2x}^-) + (v_{2x}^- + e_{2x}^-) (v_{1x}^- + e_{2x}^-) + (v_{2z}^- + e_{2z}^-) (v_{1z}^- + e_{2z}^-), \\ (L_{12}^-)_{23} &= (v_{2x}^- e_{2y}^- - v_{2y}^- e_{2x}^-) (v_{1x}^- e_{2z}^- - v_{1z}^- e_{2x}^-) - (v_{2y}^- + e_{2y}^-) (v_{1z}^- + e_{2z}^-), \\ (L_{12}^-)_{31} &= (v_{2z}^- e_{2y}^- - v_{2y}^- e_{2z}^-) (v_{1x}^- e_{2y}^- - v_{1y}^- e_{2x}^-) - (v_{2z}^- + e_{2z}^-) (v_{1x}^- + e_{2x}^-), \\ (L_{12}^-)_{32} &= (v_{2x}^- e_{2z}^- - v_{2z}^- e_{2x}^-) (v_{1x}^- e_{2y}^- - v_{1y}^- e_{2x}^-) - (v_{2z}^- + e_{2z}^-) (v_{1y}^- + e_{2y}^-), \\ (L_{12}^-)_{33} &= (v_{2y}^- e_{2x}^- - v_{2x}^- e_{2y}^-) (v_{1x}^- e_{2y}^- - v_{1y}^- e_{2x}^-) + (v_{2y}^- + e_{2y}^-) (v_{1y}^- + e_{2y}^-) + (v_{2x}^- + e_{2x}^-) (v_{1x}^- + e_{2x}^-), \end{aligned}$$

enquanto que os elementos da matriz  $L_{21}^-$  são dados da seguinte forma:

$$\begin{aligned} (L_{21}^-)_{11} &= (v_{2y}^- e_{1z}^- - v_{2z}^- e_{1y}^-) (v_{1y}^- e_{1z}^- - v_{1z}^- e_{1y}^-) - (v_{2y}^- - e_{1y}^-) (v_{1y}^- - e_{1y}^-) - (v_{2z}^- - e_{1z}^-) (v_{1z}^- - e_{1z}^-), \\ (L_{21}^-)_{12} &= (v_{2z}^- e_{1y}^- - v_{2y}^- e_{1z}^-) (v_{1x}^- e_{1z}^- - v_{1z}^- e_{1x}^-) + (v_{2y}^- - e_{1y}^-) (v_{1x}^- - e_{1x}^-), \\ (L_{21}^-)_{13} &= (v_{2y}^- e_{1z}^- - v_{2z}^- e_{1y}^-) (v_{1x}^- e_{1y}^- - v_{1y}^- e_{1x}^-) + (v_{2z}^- - e_{1z}^-) (v_{1x}^- - e_{1x}^-), \\ (L_{21}^-)_{21} &= (v_{2z}^- e_{1x}^- - v_{2x}^- e_{1z}^-) (v_{1y}^- e_{1z}^- - v_{1z}^- e_{1y}^-) + (v_{2x}^- - e_{1x}^-) (v_{1y}^- - e_{1y}^-), \\ (L_{21}^-)_{22} &= (v_{2x}^- e_{1z}^- - v_{2z}^- e_{1x}^-) (v_{1x}^- e_{1z}^- - v_{1z}^- e_{1x}^-) - (v_{2x}^- - e_{1x}^-) (v_{1x}^- - e_{1x}^-) - (v_{2z}^- - e_{1z}^-) (v_{1z}^- - e_{1z}^-), \\ (L_{21}^-)_{23} &= (v_{2z}^- e_{1x}^- - v_{2x}^- e_{1z}^-) (v_{1x}^- e_{1y}^- - v_{1y}^- e_{1x}^-) + (v_{2z}^- - e_{1z}^-) (v_{1y}^- - e_{1y}^-), \\ (L_{21}^-)_{31} &= (v_{2x}^- e_{1y}^- - v_{2y}^- e_{1x}^-) (v_{1y}^- e_{1z}^- - v_{1z}^- e_{1y}^-) + (v_{2x}^- - e_{1x}^-) (v_{1z}^- - e_{1z}^-), \\ (L_{21}^-)_{32} &= (v_{2y}^- e_{1x}^- - v_{2x}^- e_{1y}^-) (v_{1x}^- e_{1z}^- - v_{1z}^- e_{1x}^-) + (v_{2y}^- - e_{1y}^-) (v_{1z}^- - e_{1z}^-), \\ (L_{21}^-)_{33} &= (v_{2x}^- e_{1y}^- - v_{2y}^- e_{1x}^-) (v_{1x}^- e_{1y}^- - v_{1y}^- e_{1x}^-) - (v_{2x}^- - e_{1x}^-) (v_{1x}^- - e_{1x}^-) - (v_{2y}^- - e_{1y}^-) (v_{1y}^- - e_{1y}^-). \end{aligned}$$

A auto-força  $\mathbf{F}^{auto}$  associada ao sistema de duas cargas pontuais é definida como a derivada no tempo do momento total  $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ , que, através da soma das equações (1.0.2, 1.0.3), possui a seguinte forma:

$$\mathbf{F}^{auto} = \frac{d\mathbf{p}}{dt} = \mathbf{F}_1^- + \mathbf{F}_2^- - (\mathcal{M}_{21}^- \mathbf{a}_1^- + \mathcal{M}_{12}^- \mathbf{a}_2^-). \quad (1.0.13)$$

A auto-força, neste caso, nada mais é do que a força resultante devida única e exclusivamente aos campos eletromagnéticos associados à interação entre as duas cargas pontuais. Diferente da situação clássica tradicional da mecânica newtoniana, não temos a validade de uma terceira lei de Newton que implicaria na nulidade dessa força e, conseqüentemente, na conservação do momento mecânico total  $\mathbf{p}$ .

Além disso, se considerarmos a existência de uma força não eletromagnética (chamada força externa) agindo sobre as cargas pontuais e denotada por  $\mathbf{F}_i^{ext}$  ( $i = 1, 2$ ), então as taxas de variação dos momentos  $\mathbf{p}_i$  serão respectivamente:

$$\frac{d\mathbf{p}_1}{dt} = \eta \mathcal{M}_{11} \mathbf{a}_1 = \mathbf{F}_1^- - \mathcal{M}_{12}^- \mathbf{a}_2^- + \mathbf{F}_1^{ext}, \quad (1.0.14)$$

$$\frac{d\mathbf{p}_2}{dt} = \mathcal{M}_{22} \mathbf{a}_2 = \mathbf{F}_2^- - \mathcal{M}_{21}^- \mathbf{a}_1^- + \mathbf{F}_2^{ext}. \quad (1.0.15)$$

A taxa de variação do momento total  $\mathbf{p}$  (então chamada de força resultante) é:

$$\mathbf{F}^{res} = \frac{d\mathbf{p}}{dt} = \mathbf{F}^{auto} + \mathbf{F}^{ext}, \quad \mathbf{F}^{ext} = \mathbf{F}_1^{ext} + \mathbf{F}_2^{ext}, \quad (1.0.16)$$

onde a auto-força  $\mathbf{F}^{auto}$  é calculada a partir da equação (1.0.13).

É importante notar que a existência de forças externas implica que a auto-força  $\mathbf{F}^{auto}$  não é mais igual à auto-força calculada quando não há forças externas. Isso ocorre porque a auto-força depende das acelerações, portanto, depende implicitamente das forças externas. Esta dependência implícita ocorre mesmo se a força externa total  $\mathbf{F}^{ext}$  for nula. Esta questão ficará mais clara quando analisarmos mais adiante nesta seção a aproximação de instantaneidade para as equações de Newton (1.0.2, 1.0.3).

Invertendo as matrizes  $\mathcal{M}_{11}$  e  $\mathcal{M}_{22}$  para escrever o sistema de equações em um sistema dinâmico explícito de segunda ordem com retardo, tem-se:

$$\eta \mathbf{a}_1(t) = \mathcal{M}_{11}^{-1}(t) \mathbf{F}_1^-(t, t_r^2) - \mathcal{M}_{11}^{-1}(t) \mathcal{M}_{12}^-(t, t_r^2) \mathbf{a}_2(t_r^2), \quad (1.0.17)$$

$$\mathbf{a}_2(t) = \mathcal{M}_{22}^{-1}(t) \mathbf{F}_2^-(t, t_r^1) - \mathcal{M}_{22}^{-1}(t) \mathcal{M}_{21}^-(t, t_r^1) \mathbf{a}_1(t_r^1). \quad (1.0.18)$$

Para obter as matrizes inversas de  $\mathcal{M}_{11}$  e  $\mathcal{M}_{22}$  usamos o fato que as matrizes  $\mathcal{Q}_{\hat{v}_1}$  e  $\mathcal{Q}_{\hat{v}_2}$  são idempotentes, isto é,

$$\mathcal{Q}_{\hat{v}_1}^2 = \mathcal{Q}_{\hat{v}_1}, \quad \mathcal{Q}_{\hat{v}_2}^2 = \mathcal{Q}_{\hat{v}_2}.$$

A propriedade de idempotência destas matrizes vem do fato de que eles projetam qualquer vetor na direção  $\mathbf{v}_1$  e  $\mathbf{v}_2$ , respectivamente. Portanto, se considerarmos um vetor arbitrário  $\mathbf{w}$  teremos as seguintes projeções:

$$\mathcal{Q}_{\hat{v}_1} \mathbf{w} = (\mathbf{w} \cdot \mathbf{v}_1) \hat{v}_1, \quad \mathcal{Q}_{\hat{v}_2} \mathbf{w} = (\mathbf{w} \cdot \mathbf{v}_2) \hat{v}_2.$$

Consideremos agora uma matriz escrita como uma combinação linear das matrizes identidade e idempotente, ou seja,

$$a_1 \mathcal{I} + a_2 \mathcal{Q}, \quad \mathcal{Q}^2 = \mathcal{Q}, \quad (1.0.19)$$

a inversa certamente pode ser escrita como,

$$(a_1\mathcal{I} + a_2\mathcal{Q})^{-1} = b_1\mathcal{I} + b_2\mathcal{Q}, \quad (1.0.20)$$

consequentemente:

$$\begin{aligned} (a_1\mathcal{I} + a_2\mathcal{Q})(a_1\mathcal{I} + a_2\mathcal{Q})^{-1} &= (a_1\mathcal{I} + a_2\mathcal{Q})(b_1\mathcal{I} + b_2\mathcal{Q}), \\ \mathcal{I} &= (a_1b_1)\mathcal{I} + (a_1b_2)\mathcal{Q} + (a_2b_1)\mathcal{Q} + (a_2b_2)\mathcal{Q}^2, \\ \mathcal{I} &= (a_1b_1)\mathcal{I} + (a_1b_2 + a_2b_1 + a_2b_2)\mathcal{Q}, \end{aligned} \quad (1.0.21)$$

obtendo finalmente:

$$b_1 = \frac{1}{a_1} \text{ e } b_2 = -\frac{a_2}{a_1(a_1 + a_2)}. \quad (1.0.22)$$

Esses resultados aplicados às matrizes  $\mathcal{M}_{11}$  e  $\mathcal{M}_{22}$  conduzem as seguintes expressões:

$$\mathcal{M}_{11}^{-1} = \gamma_1^{-1}\mathcal{I} - \frac{\gamma_1^{-1}|\mathbf{v}_1|^2}{|\mathbf{v}_1|^2 + \gamma_1^{-2}}\mathcal{Q}_{\hat{v}_1}, \quad (1.0.23)$$

$$\mathcal{M}_{22}^{-1} = \gamma_2^{-1}\mathcal{I} - \frac{\gamma_2^{-1}|\mathbf{v}_2|^2}{|\mathbf{v}_2|^2 + \gamma_2^{-2}}\mathcal{Q}_{\hat{v}_2}. \quad (1.0.24)$$

Por uma questão de clareza na implementação do método de integração explicitamos a dependência das variáveis do sistema diferencial em relação ao tempo atual  $t$  e os respectivos tempos de retardo  $t_r^1$  e  $t_r^2$ . Vale lembrar que em geral a explicitação do tempo atual  $t$  não é comum em estudos de sistemas de equações diferenciais, pois é latente neste caso que todas as variáveis são calculadas no mesmo tempo  $t$ . As equações de Newton em (1.0.17, 1.0.18) junto com a equação (1.0.1) constituem um sistema de E.D.F.s neutros com duas equações funcionais para tempos retardados que dependem das trajetórias das partículas.

O problema de Synge consiste basicamente na questão sobre a existência de trajetórias  $\mathbf{r}_1(t)$  e  $\mathbf{r}_2(t)$  que satisfazem o sistema de E.D.F.s, constituído pelas equações (1.0.1), (1.0.17) e (1.0.18), para todo o tempo  $t$  onde as trajetórias são definidas e  $\mathbf{F}_i^{ext}(t) = 0$ . Na verdade, esse problema diz respeito à existência de soluções globais e, na literatura, é conhecido como *backward problem* [18]. Este problema ainda não foi resolvido em geral, mas soluções analíticas ou numéricas são conhecidas para certos problemas unidimensionais [18, 24, 23, 25, 26, 31, 34]. Um problema ainda mais restrito é saber se para uma determinada condição inicial existe uma solução global compatível com essa condição e se essa solução seria única [18, 23, 25]. Um teorema de existência, mas não de unicidade para sistemas unidimensionais, é apresentado nas referências [24, 31]. A grande dificuldade aqui é construir soluções globais a partir de soluções locais, como é o caso do sistema de E.D.O.s. Na verdade, a própria noção de solução local não existe devido à presença de equações funcionais que relacionam o tempo atual aos tempos retardados.

Uma formulação mais fraca do problema de Synge foi feita por Driver [16, 17], com base na formulação típica para a questão da existência e unicidade em sistemas de E.D.F.s com tempos retardados. Neste caso, não buscamos soluções válidas para todos os tempos  $t$ , mas soluções que satisfaçam as equações (1.0.1), (1.0.17) e (1.0.18) para todo o tempo  $t > t_0$  onde temos  $\mathbf{F}_i(t) = 0$ . Nestes artigos, Driver formula condições para a obtenção de um teorema de existência e unicidade para o caso unidimensional.

Nesta formulação mais fraca, a prova de um teorema de existência e unicidade está associada ao estabelecimento de algumas condições para as trajetórias das partículas em um determinado intervalo  $[\alpha, t_0]$  ou, em outros termos, é necessário conhecer a história anterior de trajetórias em um certo intervalo de tempo finito anterior ao tempo inicial  $t_0$ . Esta condição substitui a condição inicial típica de Cauchy para problemas de E.D.O.s nos quais a trajetória é fixada apenas no tempo inicial  $t_0$ . Então Driver prova, com hipóteses pouco restritivas sobre a diferenciabilidade das trajetórias anteriores junto com a hipótese de que as equações funcionais para os tempos retardados são satisfeitas no tempo inicial  $t_0$ , um teorema muito geral para existência e unicidade de soluções.

Em outro trabalho, bem menos conhecido na literatura, Driver obtém o mesmo teorema para o problema tridimensional [40]. Os resultados deste trabalho foram republicados muito mais tarde e com demonstrações e hipóteses mais simples [27]. Mesmo assim, vários trabalhos posteriores ignoram este trabalho e recuperam os resultados de Driver em condições mais restritivas [28, 29, 41, 42].

Podemos considerar que o teorema obtido por Driver resolve, de forma muito geral, o problema de existência e unicidade para a formulação fraca do problema de Synge. No entanto, o próprio Driver, em outro trabalho, chama a atenção para o fato de que a resolução da formulação fraca não resolve o problema originalmente formulado por Synge [18]. Devido ao teorema da existência e da unicidade, já demonstrado para a formulação fraca. Então, para resolver o problema original é necessário apenas obter trajetórias que satisfaçam as equações de Synge para todo o tempo  $t < t_0$ . Por essa razão, o problema de encontrar soluções globais ficou conhecido como *Backward Problem*.

Agora vale a pena fazer uma discussão mais aprofundada desses resultados e da questão de duas cargas pontuais isoladas como um problema de valor inicial para o sistema de E.D.O.s e E.D.P.s definido nas equações (0.0.1) - (0.0.5). Um problema de valor inicial para essas equações consiste em considerar em um dado instante inicial  $t_0$  o estado das partículas e a configuração espacial dos campos eletromagnéticos. A grande dúvida é se essa condição inicial dos campos é compatível com a ideia de que o sistema está isolado. Consideramos um sistema isolado quando não há outra fonte externa que influencie o campo eletromagnético do sistema.

Considerando o resultado de existência e unicidade obtido por Driver na formulação fraca do problema de Synge, podemos deduzir algumas consequências. Não há forças externas agindo sobre as cargas para  $t > t_0$ , mas para  $t < t_0$  essas forças externas podem existir e são responsáveis por impor trajetórias arbitrárias antes do tempo inicial, implicando que não precisam satisfazer o sistema de equações para duas cargas isoladas. No entanto, o fato de que devemos conhecer as trajetórias em um certo intervalo finito antes do tempo inicial implica que não precisamos conhecer as configurações do campo eletromagnético em todo o espaço para determinar as trajetórias das partículas para  $t > t_0$ .

De fato, a trajetória anterior em um intervalo finito implica, devido à velocidade de propagação finita das interações, em um campo eletromagnético definido no tempo  $t_0$  para uma região finita e compacta  $\mathcal{R}$  do espaço que contém as duas cargas. Todos os campos gerados pelas cargas antes do intervalo anterior não podem mais atuar sobre eles após o tempo  $t_0$ . Dessa forma, poderíamos considerar qualquer continuação das trajetórias antes do intervalo anterior (todas satisfazendo o princípio relativístico de ter velocidades menores que  $c$ ) e compatível com forças externas arbitrárias. Essas trajetórias anteriores definiriam uma configuração de campo no tempo  $t_0$  para os campos na região  $\mathbb{R}^3 - \mathcal{R}$  (região complementar de  $\mathcal{R}$  no instante  $t_0$ ). Todas essas configurações iniciais correspondem a uma única trajetória possível para  $t > t_0$ .

Assim, embora o problema do valor inicial possa ser formulado, ele implica condi-

ções matemáticas extremamente complicadas a serem impostas ao campo eletromagnético no tempo inicial  $t_0$ , sem se referir à dinâmica anterior das cargas. Isso nada mais é do que uma consequência de que os campos de Lienard-Wiechert são uma restrição intertemporal para as trajetórias das cargas. De fato, o teorema da unicidade e existência de Driver implica que essa restrição intertemporal é satisfeita para todos os tempos  $t \geq t_0$ . Para esta questão, recomendamos as excelentes reflexões feitas por Deckert e Hartenstein [6].

Outra questão (também discutida por Deckert e Hartenstein) é a possibilidade física de realizar trajetórias anteriores compatíveis com forças externas viáveis que não influenciam mais o movimento das cargas após o instante inicial  $t_0$ , especialmente se essas forças estão relacionadas a outras fontes de campo eletromagnético. Resumindo toda a discussão: o grande problema é garantir o isolamento das duas cargas da ação de forças externas. Esse problema, embora menos dramático (devido à inexistência de restrições intertemporais), também está presente no sistema de dois corpos pontuais com forças instantâneas entre eles, onde as equações de Newton definem um sistema de E.D.O.s. Aqui, o problema consiste em preparar uma condição inicial arbitrária, com a ajuda de forças externas, e garantir que essas forças externas não atuem mais após o momento inicial.

Por outro lado, a versão original do problema de Synge (*Backward Problem*) implica uma restrição muito mais forte na configuração inicial dos campos eletromagnéticos para garantir que as soluções do problema sejam compatíveis com um sistema de duas cargas completamente isolado para todos os momentos em que a solução é definida. Geralmente, isso implica em trajetórias que devem existir em intervalos semi-infinitos  $(-\infty, T)$  ou duplamente infinito  $(-\infty, \infty)$ . Além disso, podemos considerar quaisquer soluções globais de Synge do sistema de equações (1.0.1), (1.0.17) e (1.0.18) como a expressão da possível configuração espaço-temporal para os campos eletromagnéticos associados a um sistema isolado de duas cargas pontuais. Além disso, as trajetórias das duas cargas pontuais podem ser interpretadas apenas como uma forma de observar essas configurações espaço-temporais dos campos por meio do movimento de suas singularidades.

Por fim, observamos que o acoplamento das equações de Maxwell, associadas a uma distribuição de duas cargas pontuais, com a equação de Newton, obtida por meio da força de Lorentz, postula implicitamente que as possíveis configurações espaço-temporais do campo eletromagnético não podem ser estabelecidas sem as equações da mecânica. Deste acoplamento surge uma autoforça diferente de zero (ver equação (1.0.13)) que implica na não conservação do momento mecânico total para as partículas inerciais associadas às cargas elétricas.

O fato de considerarmos as equações de Dirac-Lorenz em vez das equações de Synge, com as respectivas correções de autoforça para cada partícula, não altera profundamente o problema matemático de definir sistemas isolados a partir das condições iniciais impostas aos campos eletromagnéticos, bem como a viabilidade da realização física de tais trajetórias iniciais anteriores para as partículas carregadas.

Para ilustrar o método de integração que será desenvolvido nesta tese, restringimos nossa análise a um sistema bidimensional, que consiste em considerar (sem perda de generalidade) todos os componentes de  $z$  nulos nas equações (1.0.17, 1.0.18). Neste caso nós temos as seguintes expressões para os vetores e matrizes que aparecem nessas equações.

$$\mathbf{F}_1^- = \frac{S\gamma_2^{-2}}{\lambda_2^- |\mathbf{r}_2^- - \mathbf{r}_1^-|^2} \begin{pmatrix} v_{1y} (v_{2y}^- e_{2x}^- - v_{2x}^- e_{2y}^-) - v_{2x}^- - e_{2x}^- \\ v_{1x} (v_{2x}^- e_{2y}^- - v_{2y}^- e_{2x}^-) - v_{2y}^- - e_{2y}^- \end{pmatrix}, \quad (1.0.25)$$

$$\mathbf{F}_2^- = \frac{S\gamma_1^{-2}}{\lambda_1^- |\mathbf{r}_2^- - \mathbf{r}_1^-|^2} \begin{pmatrix} v_{2y} (v_{1y}^- e_{1x}^- - v_{1x}^- e_{1y}^-) + v_{1x}^- - e_{1x}^- \\ v_{2x} (v_{1x}^- e_{1y}^- - v_{1y}^- e_{1x}^-) + v_{1y}^- - e_{1y}^- \end{pmatrix}, \quad (1.0.26)$$



As matrizes  $\mathcal{M}_{12}^-$  e  $\mathcal{M}_{21}^-$  são definidas da seguinte maneira:

$$\mathcal{M}_{12}^- = \frac{S}{\lambda_2^- |\mathbf{r}_2^- - \mathbf{r}_1^-|} \begin{pmatrix} + (v_{2y}^- + e_{2y}^-) (v_{1y} + e_{2y}^-) & - (v_{2x}^- + e_{2x}^-) (v_{1y} + e_{2y}^-) \\ - (v_{2y}^- + e_{2y}^-) (v_{1x} + e_{2x}^-) & + (v_{2x}^- + e_{2x}^-) (v_{1x} + e_{2x}^-) \end{pmatrix}, \quad (1.0.27)$$

$$\mathcal{M}_{21}^- = \frac{S}{\lambda_1^- |\mathbf{r}_2^- - \mathbf{r}_1^-|} \begin{pmatrix} - (v_{2y}^- - e_{1y}^-) (v_{1y}^- - e_{1y}^-) & + (v_{2y}^- - e_{1y}^-) (v_{1x}^- - e_{1x}^-) \\ + (v_{2x}^- - e_{1x}^-) (v_{1y}^- - e_{1y}^-) & - (v_{2x}^- - e_{1x}^-) (v_{1x}^- - e_{1x}^-) \end{pmatrix}, \quad (1.0.28)$$

as matrizes  $\mathcal{M}_{11}$  e  $\mathcal{M}_{22}$ , avaliadas no presente  $t$ , dependem dos operadores do projetor definidos como:

$$\mathcal{Q}_{\hat{v}_1} = \begin{pmatrix} \hat{v}_{1x}^2 & \hat{v}_{1x}\hat{v}_{1y} \\ \hat{v}_{1y}\hat{v}_{1x} & \hat{v}_{1y}^2 \end{pmatrix}, \quad \mathcal{Q}_{\hat{v}_2} = \begin{pmatrix} \hat{v}_{2x}^2 & \hat{v}_{2x}\hat{v}_{2y} \\ \hat{v}_{2y}\hat{v}_{2x} & \hat{v}_{2y}^2 \end{pmatrix}. \quad (1.0.29)$$

É interessante reescrever as equações diferenciais em (1.0.17, 1.0.18) de uma forma mais genérica como um sistema de primeira ordem:

$$\frac{d\mathbf{r}_1(t)}{dt} = \mathbf{v}_1(t), \quad \frac{d\mathbf{v}_1(t)}{dt} = \Theta_1 [\mathbf{r}_1(t), \mathbf{v}_1(t), \mathbf{r}_2(t_r^2), \mathbf{v}_2(t_r^2), \mathbf{a}_2(t_r^2)], \quad (1.0.30)$$

$$\frac{d\mathbf{r}_2(t)}{dt} = \mathbf{v}_2(t), \quad \frac{d\mathbf{v}_2(t)}{dt} = \Theta_2 [\mathbf{r}_2(t), \mathbf{v}_2(t), \mathbf{r}_1(t_r^1), \mathbf{v}_1(t_r^1), \mathbf{a}_1(t_r^1)], \quad (1.0.31)$$

onde as expressões dos campos vetoriais  $\Theta_1$  e  $\Theta_2$  são dadas respectivamente pelas equações (1.0.17, 1.0.18).

## 1.1 Aproximação instantânea para as acelerações

Para compreender, pelo menos aproximadamente, as soluções do sistema (1.0.17, 1.0.18) ou, equivalentemente, do sistema (1.0.30), (1.0.31), levaremos em consideração uma condição em que os tempos de retardo podem ser considerados muito pequenos. Então, consideremos a aproximação em que  $t_r^1 = t_r^2 = t$ , de fato, estamos considerando as interações eletromagnéticas como sendo aproximadamente instantâneas. A consequência mais importante desta aproximação é o acoplamento entre as acelerações do tempo atual  $t$ , devido as forças newtonianas, com as acelerações no tempo de retardo  $t_r$  proveniente dos campos de radiação levando-nos ao seguinte sistema dinâmico de segunda ordem:

$$\eta \mathbf{a}_1 + \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \mathbf{a}_2 = \mathcal{M}_{11}^{-1} \mathbf{F}_1, \quad (1.1.32)$$

$$\mathbf{a}_2 + \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \mathbf{a}_1 = \mathcal{M}_{22}^{-1} \mathbf{F}_2. \quad (1.1.33)$$

O sobrescrito "-" é eliminado de todas as equações, uma vez que todas as quantidades físicas são calculadas no tempo atual  $t$ . Além disso, nós temos que:

$$\mathbf{e}_1 = \mathbf{e}_2 = \hat{r} = \frac{\mathbf{r}_2 - \mathbf{r}_1}{|\mathbf{r}_2 - \mathbf{r}_1|}, \quad (1.1.34)$$

$$\lambda_1(t, t) = (\mathbf{v}_1 \cdot \hat{r} - 1)^3, \quad \lambda_2(t, t) = (\mathbf{v}_2 \cdot \hat{r} + 1)^3. \quad (1.1.35)$$

Através de uma combinação linear simples e direta entre as equações (1.1.32, 1.1.33), temos:

$$\left( \mathcal{I} - \frac{1}{\eta} \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \right) \eta \mathbf{a}_1 = \mathcal{M}_{11}^{-1} \mathbf{F}_1 - \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \mathcal{M}_{22}^{-1} \mathbf{F}_2, \quad (1.1.36)$$

$$\left( \mathcal{I} - \frac{1}{\eta} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \right) \mathbf{a}_2 = \mathcal{M}_{22}^{-1} \mathbf{F}_2 - \frac{1}{\eta} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \mathcal{M}_{11}^{-1} \mathbf{F}_1. \quad (1.1.37)$$

Escrevendo o sistema em um conjunto explícito de equações diferenciais de segunda ordem instantâneas obtemos:

$$\mathbf{a}_1 = \frac{1}{\eta} \left( \mathcal{I} - \frac{1}{\eta} \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \right)^{-1} \left( \mathcal{M}_{11}^{-1} \mathbf{F}_1 - \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \mathcal{M}_{22}^{-1} \mathbf{F}_2 \right), \quad (1.1.38)$$

$$\mathbf{a}_2 = \left( \mathcal{I} - \frac{1}{\eta} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \right)^{-1} \left( \mathcal{M}_{22}^{-1} \mathbf{F}_2 - \frac{1}{\eta} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \mathcal{M}_{11}^{-1} \mathbf{F}_1 \right). \quad (1.1.39)$$

Definindo os campos vetoriais  $\mathbf{H}_1^0$  e  $\mathbf{H}_2^0$  como:

$$\mathbf{H}_1^0 = \frac{1}{\eta} \left( \mathcal{I} - \frac{1}{\eta} \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \right)^{-1} \left( \mathcal{M}_{11}^{-1} \mathbf{F}_1 - \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \mathcal{M}_{22}^{-1} \mathbf{F}_2 \right), \quad (1.1.40)$$

$$\mathbf{H}_2^0 = \left( \mathcal{I} - \frac{1}{\eta} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \mathcal{M}_{11}^{-1} \mathcal{M}_{12} \right)^{-1} \left( \mathcal{M}_{22}^{-1} \mathbf{F}_2 - \frac{1}{\eta} \mathcal{M}_{22}^{-1} \mathcal{M}_{21} \mathcal{M}_{11}^{-1} \mathbf{F}_1 \right), \quad (1.1.41)$$

portanto,

$$\mathbf{a}'_1 = \mathbf{H}_1^0 [\mathbf{r}'_1, \mathbf{v}'_1, \mathbf{r}'_2, \mathbf{v}'_2], \quad \mathbf{a}'_2 = \mathbf{H}_2^0 [\mathbf{r}'_1, \mathbf{v}'_1, \mathbf{r}'_2, \mathbf{v}'_2]. \quad (1.1.42)$$

Renomeamos as variáveis com o acento para não confundi-las com as variáveis que aparecem no sistema de equações diferenciais dado em (1.0.30,1.0.31). Chamamos o sistema de equações diferenciais de segunda ordem em (1.1.42) de sistema instantâneo.

## 1.2 Aproximação para os tempos retardados

O desenvolvimento de um método iterativo para obter aproximações sucessivas para resolver o problema de Synge com um sistema de Equações Diferenciais Ordinárias Instantâneas será discutido ao longo desta seção.

No entanto, antes de iniciarmos esse processo, vamos reescrever o sistema (1.0.30, 1.0.31) de forma mais compacta. Para maior clareza na exposição, continuamos a explicitar as variáveis que são calculadas no tempo atual, para diferenciá-las daquelas que são calculadas nos tempos de retardo.

Definindo o vetor de estado  $\mathbf{X} = (\mathbf{r}_1, \mathbf{v}_1, \mathbf{r}_2, \mathbf{v}_2)$  e o vetor de estado estendido  $\bar{\mathbf{X}} = (\mathbf{r}_1, \mathbf{v}_1, \mathbf{a}_1, \mathbf{r}_2, \mathbf{v}_2, \mathbf{a}_2)$ , então o sistema (1.0.30, 1.0.31) pode ser escrito em uma forma compacta como:

$$\begin{aligned} \frac{d\mathbf{X}(t)}{dt} &= \Theta [\mathbf{X}(t), \bar{\mathbf{X}}(t_r^1), \bar{\mathbf{X}}(t_r^2)], \\ t_r^1 &= t - |\mathbf{r}_2(t) - \mathbf{r}_1(t_r^1)|, \quad t_r^2 = t - |\mathbf{r}_1(t) - \mathbf{r}_2(t_r^2)|, \end{aligned} \quad (1.2.43)$$

onde o campo vetorial  $\Theta$  é unicamente definido a partir do sistema de equações diferenciais (1.0.30, 1.0.31). Observe que os tempos retardados são funções do tempo atual dadas por uma relação implícita dependente da trajetória anterior ao tempo atual.

Além disso, podemos reescrever o sistema de segunda ordem (1.1.42) como o seguinte sistema de primeira ordem:

$$\frac{d\mathbf{X}}{dt} = \mathbf{H}^{(0)}(\mathbf{X}) \iff \begin{cases} \frac{d\mathbf{r}_1}{dt} = \mathbf{v}_1, & \frac{d\mathbf{v}_1}{dt} = \mathbf{H}_1^{(0)}[\mathbf{r}_1, \mathbf{v}_1, \mathbf{r}_2, \mathbf{v}_2], \\ \frac{d\mathbf{r}'_2}{dt} = \mathbf{v}_2, & \frac{d\mathbf{v}_2}{dt} = \mathbf{H}_2^{(0)}[\mathbf{r}_1, \mathbf{v}_1, \mathbf{r}_2, \mathbf{v}_2], \end{cases} \quad (1.2.44)$$

assim, podemos definir o seguinte fluxo estendido:

$$\phi_{\mathbf{X}}^{(0)}(\tau) = \left( \mathbf{r}_1^{(0)}(\tau), \mathbf{v}_1^{(0)}(\tau), \mathbf{a}_1^{(0)}(\tau), \mathbf{r}_2^{(0)}(\tau), \mathbf{v}_2^{(0)}(\tau), \mathbf{a}_2^{(0)}(\tau) \right),$$

como sendo a solução única do sistema (1.2.44) tal que  $\phi_{\mathbf{X}}^{(0)}(0) = \bar{\mathbf{X}}$ .

Obtemos uma aproximação para os tempos de retardo  $t_r^1$  e  $t_r^2$  resolvendo as equações:

$$\tau_1^{(0)} = t - \left| \mathbf{r}_2(t) - \mathbf{r}_1^{(0)}(\tau_1^{(0)}) \right|, \quad \tau_2^{(0)} = t - \left| \mathbf{r}_1(t) - \mathbf{r}_2^{(0)}(\tau_2^{(0)}) \right|,$$

do fluxo  $\phi_{\mathbf{X}(t)}^{(0)}(\tau)$  que, substituindo a equação

$$\bar{\mathbf{X}}(t_r^1) = \phi_{\mathbf{X}(t)}^{(0)}(\tau_1^{(0)}), \quad \bar{\mathbf{X}}(t_r^2) = \phi_{\mathbf{X}(t)}^{(0)}(\tau_2^{(0)})$$

fornece a seguinte equação:

$$\frac{d\mathbf{X}(t)}{dt} = \Theta \left[ \mathbf{X}(t), \phi_{\mathbf{X}(t)}^{(0)}(\tau_1^{(0)}), \phi_{\mathbf{X}(t)}^{(0)}(\tau_2^{(0)}) \right]. \quad (1.2.45)$$

Vale ressaltar que, da forma como foram definidas, temos que  $\tau_1^{(0)}$  e  $\tau_2^{(0)}$  são funções determinadas pelo vetor de estado  $\mathbf{X}(t)$  avaliado no tempo  $t$ . Desta forma, podemos remover a forma evidente das variáveis que dependem do tempo atual e reescrever o sistema como:

$$\begin{aligned} \frac{d\mathbf{X}}{dt} &= \mathbf{H}^{(1)}(\mathbf{X}) = \Theta \left[ \mathbf{X}, \phi_{\mathbf{X}}^{(0)}(\tau_1^{(0)}), \phi_{\mathbf{X}}^{(0)}(\tau_2^{(0)}) \right], \\ \tau_1^{(0)} &= -|\mathbf{r}_2 - \mathbf{r}_1^{(0)}(\tau_1^{(0)})|, \quad \tau_2^{(0)} = -|\mathbf{r}_1 - \mathbf{r}_2^{(0)}(\tau_2^{(0)})|. \end{aligned} \quad (1.2.46)$$

Observe que  $\tau_1^{(0)}$  e  $\tau_2^{(0)}$  são funções bem determinadas pelo vetor de estado  $\mathbf{X}$ .

O sistema de equações (1.2.46) pode ser visto como uma aproximação do sistema de equações em (1.0.30, 1.0.31), a vantagem aqui é que o primeiro é um sistema de equações diferenciais tradicional e pode ser integrado usando os algoritmos numéricos típicos para esses tipos de sistemas.

Se observarmos atentamente o procedimento desenvolvido até agora, é fácil notar que podemos usar um sistema de equações diferenciais, definido pelo campo vetorial  $\mathbf{H}^{(0)}(\mathbf{X})$  em (1.2.44), para obter estimativas das quantidades calculadas nos tempos de retardo no sistema de equações (1.0.30, 1.0.31). Este procedimento nos levou a um novo sistema de equações diferenciais em (1.2.46), que é definido pelo campo vetorial  $\mathbf{H}^{(1)}(\mathbf{X})$ . Se pensarmos neste procedimento como uma transformação de um sistema de equações diferenciais em outro sistema de equações diferenciais, poderíamos aplicá-lo de forma análoga ao sistema (1.2.46), que passaria a desempenhar o papel do sistema (1.2.44), para obter um novo sistema de equações diferenciais a partir das equações (1.0.30, 1.0.31).

Formalmente, podemos pensar neste procedimento como um mapa  $\mathcal{T}$  que gera uma nova sequência de sistemas de campos vetoriais (respectivamente sistemas de equações diferenciais):

$$\frac{d\mathbf{X}}{dt} = \mathbf{H}^n(\mathbf{X}) \quad (n = 0, 1, 2, \dots) \quad \Rightarrow \quad \mathbf{H}^{(n)}(\mathbf{X}) \xrightarrow{\mathcal{T}} \mathbf{H}^{(n+1)}(\mathbf{X}) \quad (1.2.47)$$

de forma que, sendo  $\phi_{\mathbf{X}}^{(n)}(\tau)$  o fluxo estendido do sistema de equações diferenciais determinado pelo campo vetorial  $\mathbf{H}^{(n)}(\mathbf{X})$ , teremos:

$$\begin{aligned} \frac{d\mathbf{X}}{dt} &= \mathbf{H}^{(n+1)}(\mathbf{X}) = \Theta \left[ \mathbf{X}, \phi_{\mathbf{X}}^{(n)}(\tau_1^{(n)}), \phi_{\mathbf{X}}^{(n)}(\tau_2^{(n)}) \right], \\ \tau_1^{(n)} &= -|\mathbf{r}_2 - \mathbf{r}_1^{(n)}(\tau_1^{(n)})|, \quad \tau_2^{(n)} = -|\mathbf{r}_1 - \mathbf{r}_2^{(n)}(\tau_2^{(n)})|, \end{aligned} \quad (1.2.48)$$

com  $\mathbf{H}^{(0)}(\mathbf{X})$  definido pelo sistema instantâneo dado em (1.2.44).

Resumidamente, para determinar as posições e velocidades das partículas, construímos o sistema de ordem zero,  $\mathbf{H}^{(0)}$ , que descreve as trajetórias das partículas desconsiderando os termos retardados nas equações. O próximo passo é considerar as equações funcionais para determinar os tempos retardados das posições e velocidades das partículas em um certo momento anterior ao tempo  $t$ . Assim, podemos determinar, utilizando o sistema de ordem zero, quais são as posições e velocidades no tempo  $t_r$ . Desta maneira, determinamos as acelerações no retardo e assim podemos resolver numericamente o sistema  $\mathbf{H}^{(1)}$  em função do tempo  $t$ .

Se a sequência de campos vetoriais converge para um ponto fixo do mapa  $\mathcal{T}$ :

$$\lim_{n \rightarrow \infty} \mathbf{H}^n(\mathbf{X}) = \mathbf{H}^\infty(\mathbf{X}), \quad \mathbf{H}^\infty(\mathbf{X}) \xrightarrow{\mathcal{T}} \mathbf{H}^\infty(\mathbf{X}), \quad (1.2.49)$$

então teremos

$$\lim_{n \rightarrow \infty} \phi_{\mathbf{X}}^{(n)}(\tau) = \phi_{\mathbf{X}}^{(\infty)}(\tau), \quad (1.2.50)$$

onde  $\phi_{\mathbf{X}}^{(\infty)}(\tau)$  é o fluxo das soluções do sistema de equações diferenciais definido pelo campo vetorial  $\mathbf{H}^\infty(\mathbf{X})$ . Da forma que a sequência foi construída, podemos esperar que este fluxo, correspondente ao ponto fixo da transformação  $\mathcal{T}$ , defina uma solução de Synge global do sistema E.D.F.s determinado pelas equações (1.0.30, 1.0.31). Nesse caso, cada campo vetorial  $\mathbf{H}^n(\mathbf{X})$  pode ser visto como definindo um sistema de equações diferenciais cujas soluções devem representar aproximações sucessivas para uma solução de Synge global.

## Capítulo II

# Campos eletromagnéticos retardados e avançados

As mesmas discussões que foram feitas para o problema Synge com  $\alpha = 1/2$ , onde apenas campos retardados são considerados, podem ser feitas para o caso em que consideramos as restrições intertemporais impostas pelos campos de Liénard-Wiechert para valores de  $\alpha > 1/2$ , onde levamos os campos avançados em consideração. No que diz respeito à existência de soluções globais, mostrou-se a existência de soluções circulares, quase circulares e particulares para o caso simétrico ( $\alpha = 0$ ) nas referências [19, 21, 22]. Além disso, alguns teoremas de existência e unicidade de soluções globais associadas a uma determinada condição inicial das partículas foram demonstrados para casos unidimensionais [26].

A versão fraca do problema tridimensional de Synge para campos simétricos é demonstrada nas referências [30, 31], onde a ideia de conhecer uma trajetória anterior é substituída pelo conhecimento de trajetórias assintóticas nos intervalos  $(-\infty, -T)$  e  $(T, \infty)$  com  $T > 0$ . Nesse caso, um teorema de existência e unicidade garante a existência de uma trajetória única no intervalo finito  $[-T, T]$ . A partir desse resultado, os autores demonstram um teorema análogo ao obtido por Driver para o problema original de Synge ( $\alpha = 1/2$ ) [27]. Aparentemente, os autores deste artigo não conheciam o teorema da existência e da unicidade de Driver para o caso tridimensional. Além disso, como os autores na referência [30] estão interessados apenas no caso simétrico, eles apenas indicam que as provas para o caso geral com qualquer  $\alpha$  seriam análogas. Até onde sabemos, este parece ser a única prova de um teorema de existência e unicidade para o caso geral de campos eletromagnéticos mistos.

Podemos reescrever as equações (0.0.32) e (0.0.33) como um sistema de equações diferenciais de segunda ordem na forma matricial com os tempos de retardo e avançado:

$$\eta \mathcal{M}_{11} \mathbf{a}_1 = (1/2 + \alpha) (\mathbf{F}_1^- - \mathcal{M}_{12}^- \mathbf{a}_2^-) + (1/2 - \alpha) (\bar{\mathbf{F}}_1^+ - \bar{\mathcal{M}}_{12}^+ \mathbf{a}_2^+), \quad (2.0.1)$$

$$\mathcal{M}_{22} \mathbf{a}_2 = (1/2 + \alpha) (\mathbf{F}_2^- - \mathcal{M}_{21}^- \mathbf{a}_1^-) + (1/2 - \alpha) (\bar{\mathbf{F}}_2^+ - \bar{\mathcal{M}}_{21}^+ \mathbf{a}_1^+), \quad (2.0.2)$$

onde devemos levar em consideração dois tempos de retardo e dois tempos avançados, respectivamente, dados por:

$$t_r^1 = t - |\mathbf{r}_2(t) - \mathbf{r}_1(t_r^1)|, \quad t_r^2 = t - |\mathbf{r}_1(t) - \mathbf{r}_2(t_r^2)|, \quad (2.0.3)$$

$$t_a^1 = t + |\mathbf{r}_2(t) - \mathbf{r}_1(t_a^1)|, \quad t_a^2 = t + |\mathbf{r}_1(t) - \mathbf{r}_2(t_a^2)|. \quad (2.0.4)$$

As quantidades matemáticas com sobrescritos "-" e "+" devem ser calculadas respectivamente nos tempos de retardo e avançado, onde as quantidades que aparecem na equação (2.0.1) dependem de  $t_r^2$  e  $t_a^2$  e das quantidades que aparecem na equação (2.0.2) dependem de  $t_r^1$  e  $t_a^1$ .

De forma análoga desenvolvida em (1.2), podemos associar uma sequência de sistemas de equações diferenciais, como feito em (1.2.47) e (1.2.48), para encontrar soluções naturais para o sistema de equações com tempos retardados e avançados dados em (2.0.1) e (2.0.2). Para fazer isso, devemos reescrever este sistema de forma semelhante a (1.2.43):

$$\frac{d\mathbf{X}(t)}{dt} = \Theta [\mathbf{X}(t), \bar{\mathbf{X}}(t_r^1), \bar{\mathbf{X}}(t_r^2), \bar{\mathbf{X}}(t_a^1), \bar{\mathbf{X}}(t_a^2)], \quad (2.0.5)$$

onde temos que levar em consideração tanto a dependência dos tempos retardados em (2.0.3) como também dos tempos avançados em (2.0.4).

A transformação do sistema de equações diferenciais de ordem  $n$  para o sistema de ordem  $n + 1$  é definida da mesma forma que em (1.2.48), mas levando em consideração o campo vetorial  $\Theta$  definido em (2.0.5). Assim, sendo  $\phi_{\mathbf{X}}^{(n)}(\tau)$  o fluxo estendido do sistema de equações diferenciais determinado pelo campo vetorial  $\mathbf{H}^{(n)}(\mathbf{X})$ , teremos:

$$\frac{d\mathbf{X}}{dt} = \mathbf{H}^{(n+1)}(\mathbf{X}) = \Theta [\mathbf{X}, \phi_{\mathbf{X}}^{(n)}(\tau_1^{(n)}), \phi_{\mathbf{X}}^{(n)}(\tau_2^{(n)}), \phi_{\mathbf{X}}^{(n)}(\bar{\tau}_1^{(n)}), \phi_{\mathbf{X}}^{(n)}(\bar{\tau}_2^{(n)})], \quad (2.0.6)$$

$$\begin{aligned} \tau_1^{(n)} &= -|\mathbf{r}_2 - \mathbf{r}_1^{(0)}(\tau_1^{(n)})|, & \tau_2^{(n)} &= -|\mathbf{r}_1 - \mathbf{r}_2^{(n)}(\tau_2^{(n)})|, \\ \bar{\tau}_1^{(n)} &= |\mathbf{r}_2 - \mathbf{r}_1^{(0)}(\bar{\tau}_1^{(n)})|, & \bar{\tau}_2^{(n)} &= |\mathbf{r}_1 - \mathbf{r}_2^{(n)}(\bar{\tau}_2^{(n)})|. \end{aligned} \quad (2.0.7)$$

Agora o fluxo estendido

$$\phi_{\mathbf{X}}^{(n)}(\tau) = \left( \mathbf{r}_1^{(n)}(\tau), \mathbf{v}_1^{(n)}(\tau), \mathbf{a}_1^{(n)}(\tau), \mathbf{r}_2^{(n)}(\tau), \mathbf{v}_2^{(n)}(\tau), \mathbf{a}_2^{(n)}(\tau) \right)$$

também deve ser calculado nos tempos  $\bar{\tau}_1^{(n)}$  e  $\bar{\tau}_2^{(n)}$ , que constituem uma sequência de aproximação para os tempos avançados.

## 2.1 Aproximação instantânea para o campo simétrico

Supondo que os intervalos do retardo e os intervalos avançados sejam muito pequenos, podemos fazer a aproximação de instantaneidade  $t_r^1 \approx t_r^2 \approx t_a^1 \approx t_a^2 \approx t$  nas equações (2.0.1, 2.0.2), que então se transformam em um sistema de equações diferenciais implícito de segunda ordem com todas as funções e derivadas definidas no tempo  $t$ . Procedendo de forma análoga ao que foi feito para campos retardados na subseção (1.1) para obter as equações (1.1.38, 1.1.39), podemos resolver explicitamente o sistema como uma função das acelerações para obter um sistema de equações diferenciais explicitamente de segunda ordem dado por:

$$\mathbf{a}_1 = \frac{1}{\eta} \left( \mathcal{I} - \frac{1}{\eta} \mathcal{M}_{11}^{-1} \mathcal{M}_{12}^{\alpha} \mathcal{M}_{22}^{-1} \mathcal{M}_{21}^{\alpha} \right)^{-1} \left( \mathcal{M}_{11}^{-1} \mathbf{F}_1^{\alpha} - \mathcal{M}_{11}^{-1} \mathcal{M}_{12}^{\alpha} \mathcal{M}_{22}^{-1} \mathbf{F}_2^{\alpha} \right), \quad (2.1.8)$$

$$\mathbf{a}_2 = \left( \mathcal{I} - \frac{1}{\eta} \mathcal{M}_{22}^{-1} \mathcal{M}_{21}^{\alpha} \mathcal{M}_{11}^{-1} \mathcal{M}_{12}^{\alpha} \right)^{-1} \left( \mathcal{M}_{22}^{-1} \mathbf{F}_2^{\alpha} - \frac{1}{\eta} \mathcal{M}_{22}^{-1} \mathcal{M}_{21}^{\alpha} \mathcal{M}_{11}^{-1} \mathbf{F}_1^{\alpha} \right), \quad (2.1.9)$$

onde

$$\mathbf{F}_i^{\alpha} = \left[ \left( \frac{1}{2} + \alpha \right) \mathbf{F}_i^{-} + \left( \frac{1}{2} - \alpha \right) \mathbf{F}_i^{+} \right]_{t_r=t, t_a=t}, \quad (2.1.10)$$

$$\mathcal{M}_{ij}^{\alpha} = \left[ \left( \frac{1}{2} + \alpha \right) \mathcal{M}_{ij}^{-} + \left( \frac{1}{2} - \alpha \right) \mathcal{M}_{ij}^{+} \right]_{t_r=t, t_a=t}, \quad (2.1.11)$$

com  $\mathbf{F}_i^{-}$ ,  $\mathbf{F}_i^{+}$ ,  $\mathcal{M}_{ij}^{-}$  e  $\mathcal{M}_{ij}^{+}$  obtidos das respectivas quantidades sobrescritas avaliadas no tempo  $t$ .

Analisaremos apenas soluções planares e, portanto, serão considerados, sem perda de generalidade, soluções contidas no plano  $z = 0$  do sistema (2.1.8, 2.1.9) quando expressas em coordenadas cartesianas  $(x, y, z)$ . Isso corresponde a considerar as condições iniciais para velocidades e posições das partículas que estão todas contidas no plano  $z = 0$ , sendo fácil de verificar, diretamente das equações (2.1.8, 2.1.9), que as duas partículas terão acelerações com componentes  $z$  nulas.

## 2.2 A existência de soluções periódicas circulares

Para estudar a existência de trajetórias circulares no caso de campos eletromagnéticos simétricos ( $\alpha = 0$ ) e forças atrativas coulombianas ( $S = -1$ ), consideremos a aproximação não relativística das equações (2.1.8, 2.1.9) para condições iniciais planas. Isso pode ser feito considerando que os componentes  $z$  de todas as quantidades que aparecem no lado esquerdo das equações são nulos, implicando em um sistema de equações diferenciais de segunda ordem para os componentes  $(x, y)$  das coordenadas cartesianas escolhidas para expressar as equações, uma vez que a componente  $z$  das acelerações de partículas torna-se nulo.

A aproximação não relativística implica que fazemos uma expansão de segunda ordem nas velocidades do lado esquerdo das equações de Newton, uma vez que para  $\alpha = 0$  os termos de primeira ordem nesta expansão são nulos. O sistema de equações de segunda ordem obtido em coordenadas cartesianas é mostrado no apêndice (A). Essas equações, projetadas em coordenadas polares usadas para descrever o vetor de posição relativo  $\mathbf{r}$ , podem ser escritas como<sup>1</sup>:

$$\frac{d\theta}{dt} = \frac{\eta + 1}{\eta} \frac{l_z}{r^2}, \quad (2.2.12)$$

$$\begin{aligned} \frac{dl_z}{dt} = & \frac{(4\eta^2 r^2 - 3\eta^2 + 2\eta - 3)l_z v_r}{\eta r^2(\eta r^2 - 1)(\eta + 1)} - \frac{2(\eta - 1)l_z V_r}{\eta r^2(\eta r^2 - 1)} \\ & - \frac{(\eta r^2 + 3)(\eta - 1)v_r V_\theta}{r(\eta + 1)(\eta r^2 - 1)} + \frac{2V_r V_\theta}{r}, \end{aligned} \quad (2.2.13)$$

$$\begin{aligned} \frac{dv_r}{dt} = & -\frac{\eta + 1}{\eta r^2} - \frac{2v_r^2}{r^2(\eta + 1)} + \frac{(\eta + 1)^2 l_z^2}{r^3 \eta^2} - \frac{2l_z^2}{r^5 \eta^2} - \frac{2(\eta + 1)V_r^2}{\eta r^2} \\ & + \frac{2(\eta r + r + 1)V_\theta^2}{\eta r^3} + \frac{(\eta - 1)(\eta r + r + 2)l_z V_\theta}{r^4 \eta^2}, \end{aligned} \quad (2.2.14)$$

$$\begin{aligned} \frac{dV_r}{dt} = & \frac{(\eta + 1)l_z V_\theta}{\eta r^2} - \frac{(\eta - 1)(\eta r + r - 1)l_z^2}{\eta^2 r^5 (\eta + 1)} + \frac{2(\eta - 1)v_r^2}{r^2(\eta + 1)^2} + \frac{(1 - \eta)V_\theta^2}{r^3 \eta (\eta + 1)} \\ & + \frac{4v_r V_r}{r^2(\eta + 1)} - \frac{(2\eta^2 r + \eta^2 + 2\eta r - 2\eta + 1)l_z V_\theta}{(\eta + 1)\eta^2 r^4}, \end{aligned} \quad (2.2.15)$$

$$\begin{aligned} \frac{dV_\theta}{dt} = & \frac{4(r + 1)(\eta r + 1)v_r V_\theta}{r^2(\eta + 1)(\eta r^2 - 1)} - \frac{3(r + 1)(\eta - 1)(\eta r + 1)l_z v_r}{(\eta + 1)\eta r^3(\eta r^2 - 1)} \\ & - \frac{2(r + 1)(\eta r + 1)l_z V_r}{\eta r^3(\eta r^2 - 1)} - \frac{(\eta + 1)l_z V_r}{\eta r^2}, \end{aligned} \quad (2.2.16)$$

com  $l_z$  sendo a componente  $z$  do momento angular e as velocidades, projetadas na direção relativa, são definidas como  $v_r = \mathbf{v} \cdot \hat{r}$ ,  $V_r = \mathbf{V} \cdot \hat{r}$ ,  $V_\theta = \mathbf{V} \cdot \hat{\theta}$ , onde  $\mathbf{v}$  é a velocidade relativa e  $\mathbf{V}$  a velocidade do centro de massa. Lembrando que o vetor relativo em coordenadas

<sup>1</sup>O passos necessários para realizar a transformação dos vetores e a projeção para as coordenadas polares são indicados no apêndice (B).

cartesianas é escrito como  $\mathbf{r} = r \cos(\theta)\hat{x} + r \sin(\theta)\hat{y}$  e temos as seguintes definições:  $\hat{r} = \cos(\theta)\hat{x} + \sin(\theta)\hat{y}$  and  $\hat{\theta} = -\sin(\theta)\hat{x} + \cos(\theta)\hat{y}$ .

Para obter uma solução estacionária, devemos impor que todas as derivadas de tempo sejam nulas, o que implica em:

$$v_r = 0, \quad (2.2.17)$$

$$V_r = 0, \quad (2.2.18)$$

$$V_\theta = \frac{1}{2} \frac{l_z}{\eta(\eta-1)r} \left( -1 + \eta^3 r^2 + (2r^2 - 2r - 1)\eta^2 + (r^2 - 2r + 2)\eta \right. \\ \left. - (\eta+1) \sqrt{\eta^2 r^4 (\eta+1)^2 - 4\eta^2 r^3 (\eta+1) - 2\eta r^2 (\eta^2 - 4\eta + 1) + (\eta-1)^2} \right), \quad (2.2.19)$$

$$0 = -\frac{\eta+1}{\eta r^2} + \frac{(\eta+1)^2 l_z^2}{r^3 \eta^2} - \frac{2l_z^2}{r^5 \eta^2} + \frac{2(\eta r + r + 1)V_\theta^2}{\eta r^3} + \frac{(\eta-1)(\eta r + r + 2)l_z V_\theta}{r^4 \eta^2}. \quad (2.2.20)$$

A equação (2.2.19) pode ser reescrito como  $V_\theta = f(\eta, r) l_z$ , que substituída na equação (2.2.20) leva a:

$$0 = g(\eta, r) l_z^2 - \frac{\eta+1}{\eta r^2} \Rightarrow l_z^2 = \frac{\eta+1}{\eta r^2 g(\eta, r)}, \quad (2.2.21)$$

onde  $g(\eta, r)$  é definida como:

$$g(\eta, r) = \frac{(\eta+1)^2}{\eta^2 r^3} - \frac{2}{\eta^2 r^5} + \frac{2(\eta r + r + 1)f^2}{\eta r^3} + \frac{(\eta-1)(\eta r + r + 2)f}{\eta^2 r^4}. \quad (2.2.22)$$

Portanto, mostramos a existência de uma solução estacionária em que o momento angular ( $l_z$ ) e a coordenada radial ( $r$ ) são constantes, ou seja, temos um movimento circular na coordenada relativa. Esta solução estacionária implica uma velocidade não nula e constante ( $V_\theta$ ) para o centro de massa, mostrando que ele tem uma aceleração não nula. Como veremos em capítulos futuros as soluções obtidas a partir da integração numérica das equações de Newton (2.1.8, 2.1.9) para valores de  $\eta$  maior que 1 veremos claramente que o centro de massa não permanece em repouso.

A expansão das funções  $g(\eta, r)$  e  $V_\theta$  em potências de  $1/r$  implica que:

$$g(\eta, r) = \frac{(\eta+1)^2}{\eta^2 r^3} + O\left(\frac{1}{r^5}\right), \quad V_\theta = \frac{l_z(\eta-1)}{\eta r^2(\eta+1)} + O\left(\frac{1}{r^3}\right), \quad (2.2.23)$$

assim, para  $r \gg 1$  temos  $V_\theta \ll 1$  e podemos considerar a aproximação onde a velocidade do centro de massa é considerada nula para obter, a partir das equações (2.2.12, 2.2.13, 2.2.14), o seguinte sistema diferencial para o vetor relativo  $\mathbf{r}$ :

$$\dot{\theta} = \frac{\eta+1}{\eta} \frac{l_z}{r^2}, \quad (2.2.24)$$

$$\dot{l}_z = \frac{(4\eta^2 r^2 - 3\eta^2 + 2\eta - 3) l_z \dot{r}}{\eta r^2 (\eta r^2 - 1) (\eta + 1)}, \quad (2.2.25)$$

$$\ddot{r} = -\frac{\eta+1}{\eta r^2} - \frac{2\dot{r}^2}{r^2(\eta+1)} + \frac{(\eta+1)^2 l_z^2}{r^3 \eta^2} - \frac{2l_z^2}{r^5 \eta^2}. \quad (2.2.26)$$

O sistema acima tem uma solução estacionária bem definida que corresponde a uma órbita circular para a coordenada relativa que está bem próxima da órbita do sistema



sem considerar o centro de massa em repouso, desde que o raio  $r$  da órbita circular seja supostamente muito grande. Podemos obter outra aproximação direta para  $r \gg 1$  que leva ao seguinte sistema:

$$\dot{\theta} = \frac{\eta + 1}{\eta} \frac{l_z}{r^2}, \quad (2.2.27)$$

$$\dot{l}_z = \frac{4l_z \dot{r}}{(\eta + 1)r^2}, \quad (2.2.28)$$

$$\ddot{r} = -\frac{(\eta + 1)}{\eta r^2} + \frac{(\eta + 1)^2 l_z^2}{r^3 \eta^2} - \frac{2\dot{r}^2}{(\eta + 1)r^2}. \quad (2.2.29)$$

Este sistema pode ser visto como uma correção do modelo de Coulomb de interação entre duas cargas pontuais, considerando as interações instantâneas entre as cargas através de seus campos eletromagnéticos simétricos retardados e avançados.

## Capítulo III

# Implementação do método iterativo para obter aproximações sucessivas das equações diferenciais instantâneas

A implementação numérica do algoritmo para obtenção de soluções do sistema de E.D.O.s definida pelo campo vetorial  $\mathbf{H}^{(n+1)}$  na equação (2.0.6) e a equação funcional (2.0.7) depende do fluxo  $\phi_{\mathbf{X}}^{(n)}(\tau)$ . Este fluxo é obtido através da implementação de um algoritmo de Runge-Kutta de quarta ordem para o sistema definido pelo campo vetorial  $\mathbf{H}^{(n)}$ . Para o caso do sistema de E.D.O.s instantâneas, definido pelo campo vetorial  $\mathbf{H}^{(0)}$ , não precisamos resolver a equação funcional (2.0.7). Depois de construir o algoritmo para resolver as equações funcionais, construímos um novo Runge-Kutta de quarta ordem para o sistema E.D.O.s definido por  $\mathbf{H}^{(n+1)}$ . Observamos que a inicialização desse processo iterativo a partir do sistema instantâneo, que é um sistema de E.D.O.s sem equações funcionais, é fundamental para a construção dos fluxos em cada etapa  $n$  do método.

Para simplificar o método, podemos fazer a seguinte aproximação para os tempos retardados e avançados calculados em cada  $n$  passo:  $\tau_1^{(n)} = \tau_2^{(n)} = -|\mathbf{r}_2 - \mathbf{r}_1|$ ,  $\bar{\tau}_1^{(n)} = \bar{\tau}_2^{(n)} = |\mathbf{r}_2 - \mathbf{r}_1|$ . Desta forma, a cada etapa do método, não precisamos mais resolver as equações funcionais. Esta abordagem é razoável para obter soluções quase circulares e simplifica a implementação do algoritmo, além de torná-la mais rápida computacionalmente. O preço a pagar é que a solução obtida, mesmo com a convergência da sequência dos sistemas de E.D.O.s na equação (1.2.49), será apenas uma aproximação para a solução do problema de Synge.

### 3.1 Trajetórias circulares coulombianas

A condição inicial para essas trajetórias deve corresponder a uma trajetória Kepleriana quase circular, conforme analisado no artigo de Synge [1]. Iremos escolher as condições iniciais que corresponderiam a trajetórias circulares coulombianas, ou seja, condições iniciais que implicariam em um movimento circular das partículas se apenas a força atrativa coulombiana atuasse entre elas, o que implica escolher  $S = -1$  no sistema de equações em (0.0.32, 0.0.33). Com as condições iniciais escolhidas desta forma, garantimos que as respectivas trajetórias serão planas.

Ilustraremos nas seguintes subseções trajetórias para sistemas com campos eletromagnéticos retardados ( $\alpha = 1/2$ ) e para sistemas com campos eletromagnéticos simétricos ( $\alpha = 0$ ). Para todas as trajetórias mostradas nas figuras desta seção, escolhemos a distância inicial entre as partículas como sendo  $|\mathbf{r}_0|$  igual a 50 e 18789, de modo que o vetor do centro de massa  $\mathbf{R}$  de ambos é nulo. Lembrando que  $\mathbf{r}_1$  e  $\mathbf{r}_2$  são respectivamente os vetores de posição das partículas 1 e 2, temos que o vetor relativo e o vetor do centro de massa são definidos como:  $\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1$  e  $\mathbf{R} = (\eta\mathbf{r}_1 + \mathbf{r}_2)/(\eta + 1)$ .

Para o tempo  $t_0 = 0$  e com a distância inicial entre as partículas dada por  $r_0$  as condições para as posições e velocidades nas coordenadas cartesianas são dados por (Ver apêndice (D)):

- Velocidade angular relativa entre as partículas no tempo inicial  $t_0$ :

$$r_0(d\theta/dt)_{t=0} = ((\eta + 1)/\eta)^{1/2}r_0^{-1/2}$$

- Condições iniciais:

$$\begin{aligned} x_1(t_0) &= -\frac{r_0}{\eta + 1}, \\ y_1(t_0) &= 0, \\ V_{1x}(t_0) &= 0, \\ V_{1y}(t_0) &= x_1 \left. \frac{d\theta}{dt} \right|_{t=0}, \\ x_2(t_0) &= \frac{\eta r_0}{\eta + 1}, \\ y_2(t_0) &= 0, \\ V_{2x}(t_0) &= 0, \\ V_{2y}(t_0) &= x_2 \left. \frac{d\theta}{dt} \right|_{t=0}. \end{aligned}$$

#### 3.1.1 Campos Eletromagnéticos Retardados

Nas figuras (3.1.1, 3.1.2) mostramos a integração numérica das trajetórias para o sistema de duas cargas com razão entre as massas iguais a  $\eta = 1$  e  $\eta = 2$ , correspondendo aos quatro primeiros sistemas de equações diferenciais na sequência (1.2.47), onde temos  $n = 0, n = 1, n = 2$  e  $n = 3$ . Para facilitar a exposição do conteúdo das figuras, denotamos esses sistemas respectivamente por  $S_0, S_1, S_2$  e  $S_3$ .

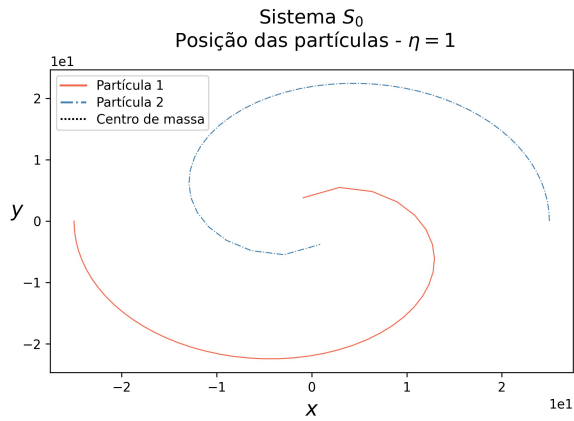
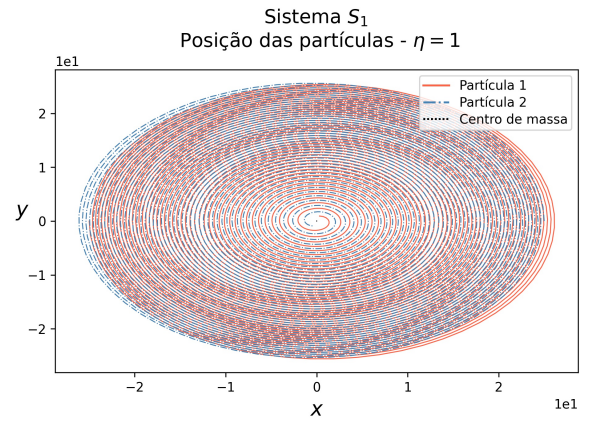
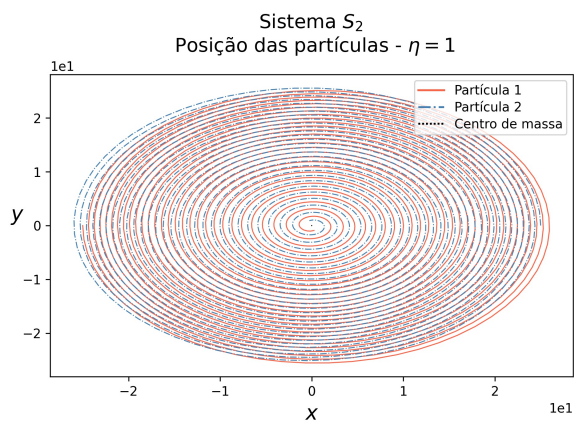
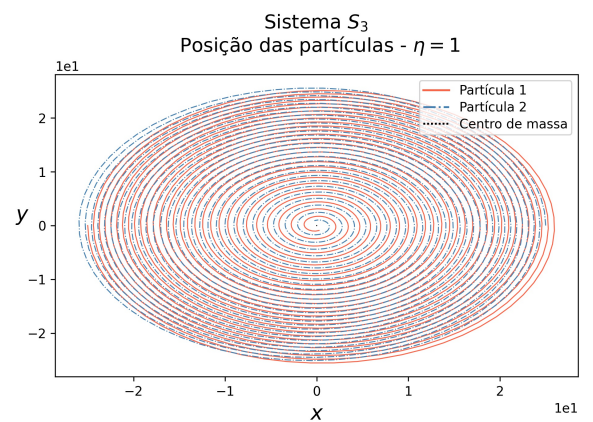
(a)  $S_0$ .(b)  $S_1$ .(c)  $S_2$ .(d)  $S_3$ .

Figura 3.1.1: Trajetórias planares em coordenadas cartesianas  $(x, y)$  para os vetores  $\mathbf{r}_1$ ,  $\mathbf{r}_2$  e  $\mathbf{R}$ , onde integramos numericamente (para a mesma condição inicial) os quatro primeiros sistemas de equações diferenciais da sequência (1.2.47) considerando  $\eta = 1$ .

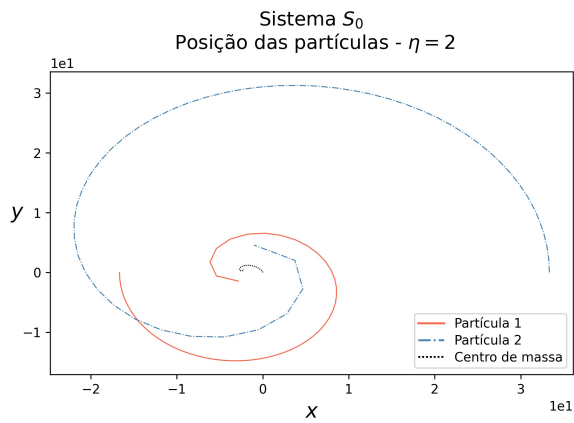
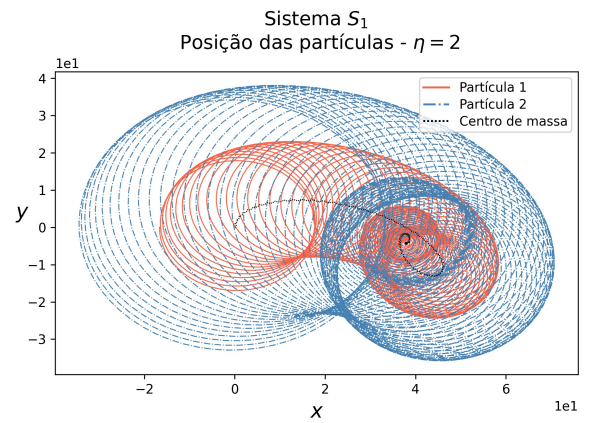
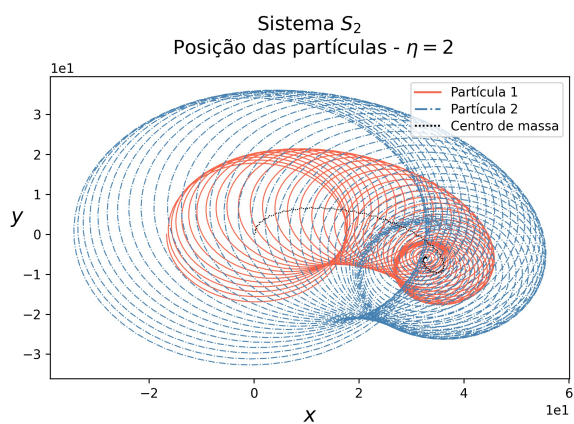
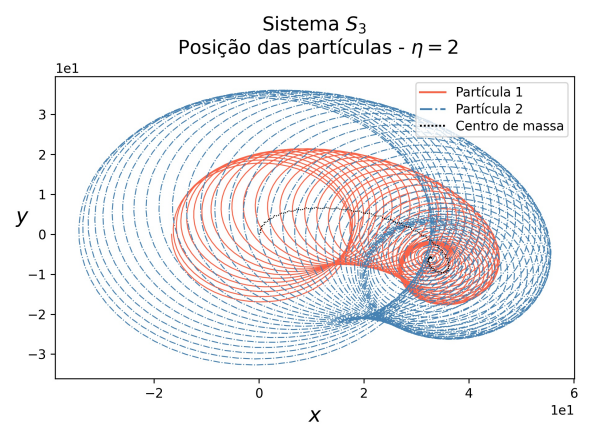
(a)  $S_0$ .(b)  $S_1$ .(c)  $S_2$ .(d)  $S_3$ .

Figura 3.1.2: Trajetórias planares em coordenadas cartesianas  $(x, y)$  para os vetores  $\mathbf{r}_1$ ,  $\mathbf{r}_2$  e  $\mathbf{R}$ , onde integramos numericamente (para a mesma condição inicial) os quatro primeiros sistemas de equações diferenciais da sequência (1.2.47) considerando  $\eta = 2$ .

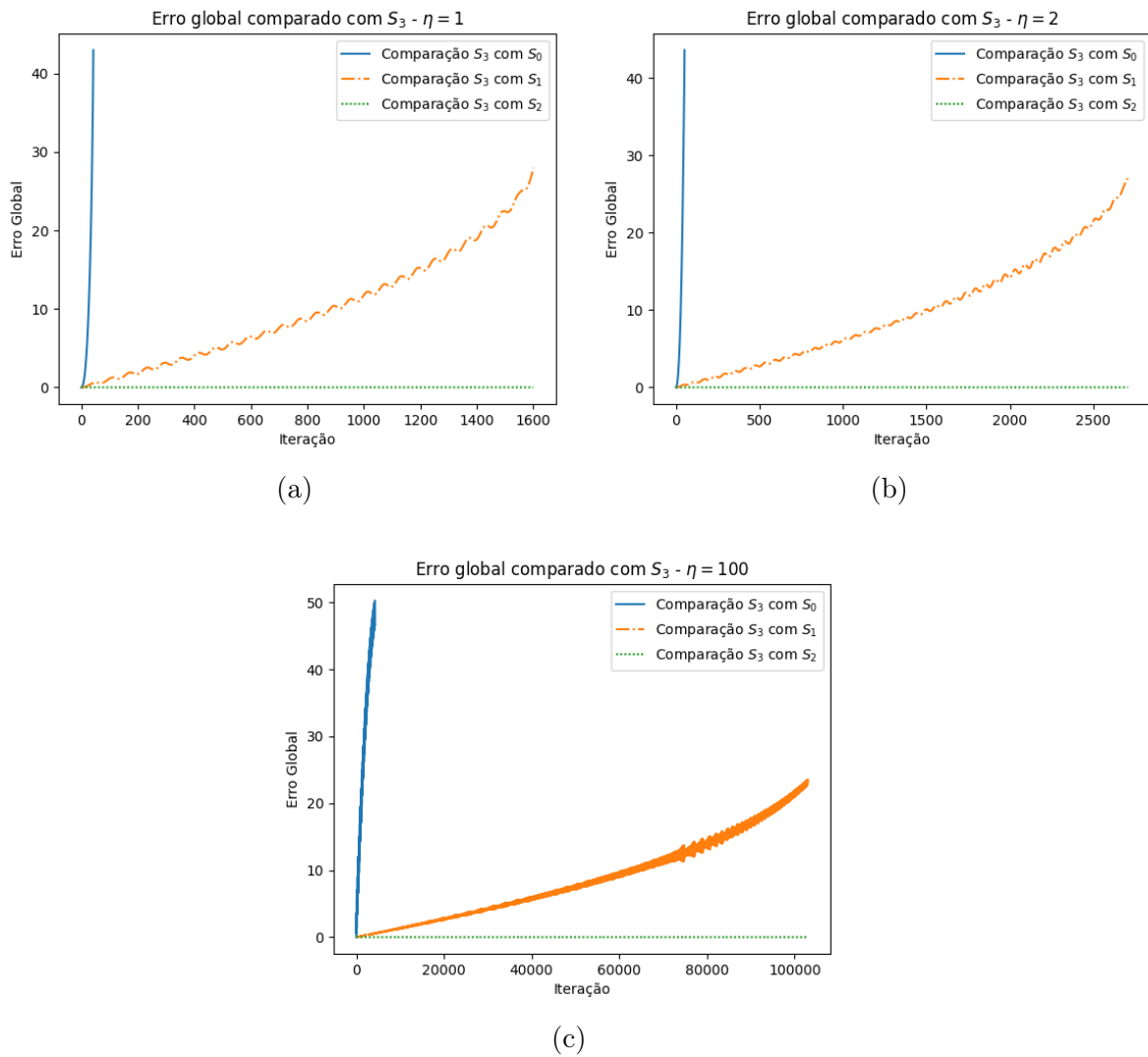


Figura 3.1.3: Comparação dos erros da aproximação  $S_3$  com relação a  $S_0$ ,  $S_1$  e  $S_2$  para  $\eta$  igual a 1, 2 e 100 considerando o raio inicial igual a  $r_0 = 50$ .

Podemos observar que o erro entre o sistema  $S_2$  e  $S_3$  é muito pequeno, mostrando que a convergência da sequência (1.2.47) é bastante rápida no caso da trajetória quase circular apresentada. Este fato pode ser claramente observado nas figuras (3.1.4, 3.1.5), onde plotamos as trajetórias relativas das partículas. Nas figuras (3.1.6, 3.1.7) temos os exemplos da convergência do método para o caso  $\eta = 2$  e  $\eta = 100$ , respectivamente, para a trajetória do centro de massa.

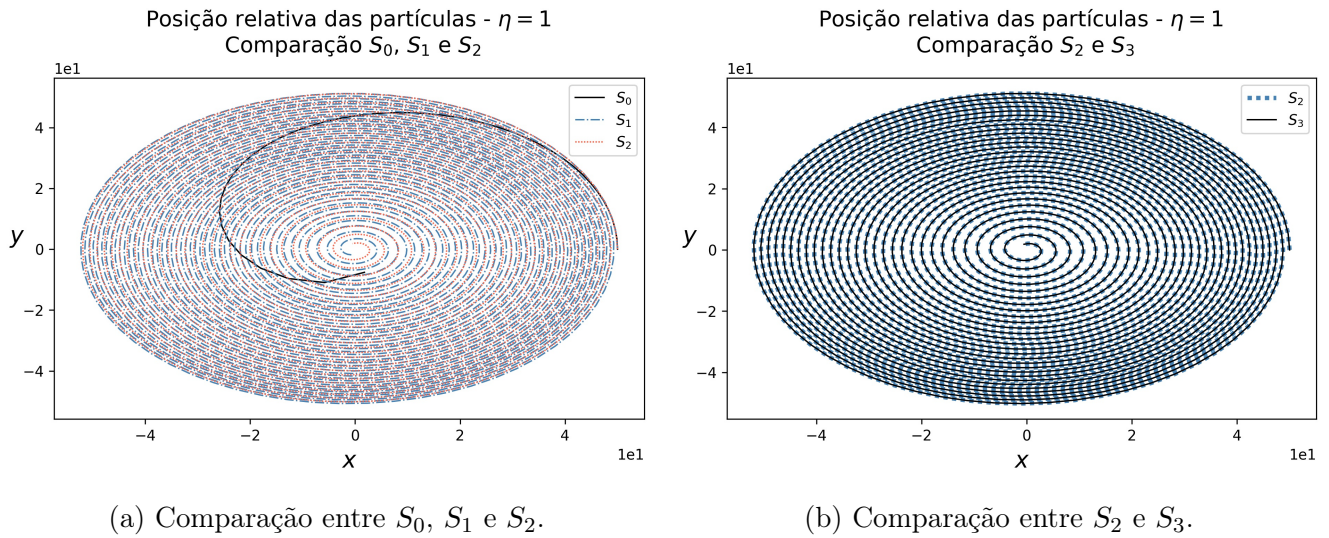
(a) Comparação entre  $S_0, S_1$  e  $S_2$ .(b) Comparação entre  $S_2$  e  $S_3$ .

Figura 3.1.4: Comparação da trajetória relativa das partículas, obtido por integração numérica dos sistemas de equações diferenciais  $S_0, S_1, S_2$  e  $S_3$  para  $\eta = 1$ . Na figura (3.1.4a) podemos observar a mudança da trajetória relativa nas sucessivas aproximações dadas por (1.2.47). Na figura (3.1.4b) o sistema  $S_2$  e o sistema  $S_3$  possuem trajetórias idênticas. Os vetores são mostrados em coordenadas cartesianas planas  $(x, y)$ .

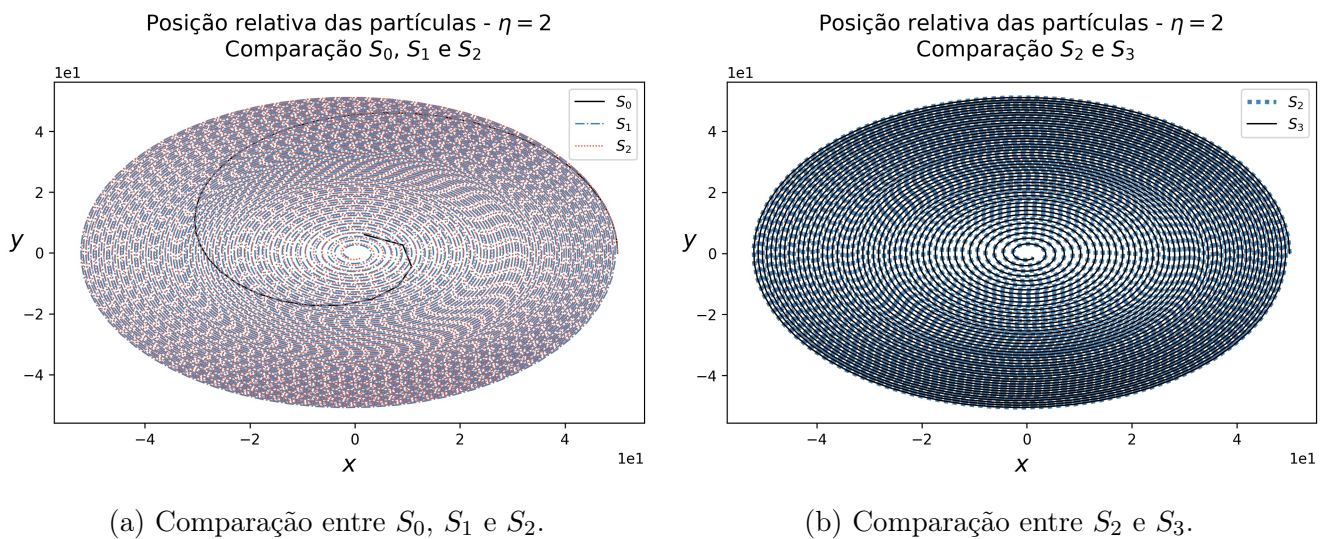
(a) Comparação entre  $S_0, S_1$  e  $S_2$ .(b) Comparação entre  $S_2$  e  $S_3$ .

Figura 3.1.5: Comparação da trajetória relativa das partículas, obtido por integração numérica dos sistemas de equações diferenciais  $S_0, S_1, S_2$  e  $S_3$  para  $\eta = 2$ . Como observado em (3.1.4) temos que na figura (3.1.5a) a mudança da trajetória relativa nas sucessivas aproximações dadas por (1.2.47). Na figura (3.1.5b) o sistema  $S_2$  e o sistema  $S_3$  possuem trajetórias idênticas. Os vetores são mostrados em coordenadas cartesianas planas  $(x, y)$ .

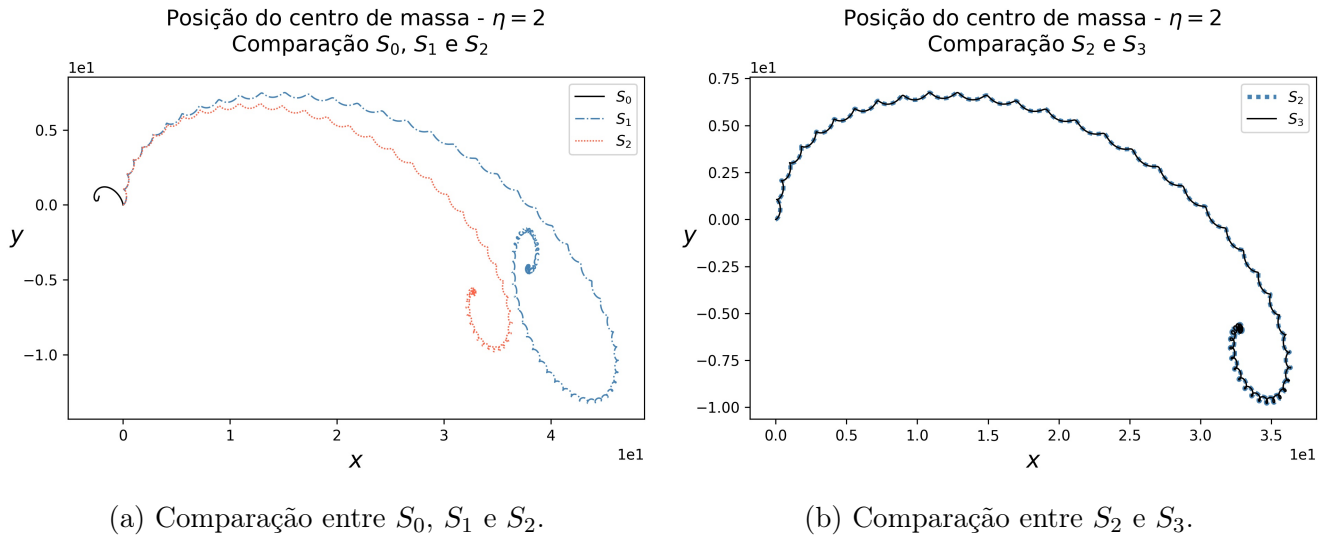


Figura 3.1.6: Comparação da trajetória do centro de massa das partículas, obtido por integração numérica dos sistemas de equações diferenciais  $S_0, S_1, S_2$  e  $S_3$  para  $\eta = 2$ . Assim como nos casos apresentados nas figuras (3.1.4) e (3.1.5), temos na figura (3.1.6a) a mudança da trajetória do centro de massa nas sucessivas aproximações dadas por (1.2.47). Na figura (3.1.6b) o sistema  $S_2$  e o sistema  $S_3$  possuem trajetórias idênticas. Os vetores são mostrados em coordenadas cartesianas planas  $(x, y)$ . A trajetória do centro de massa para o caso  $\eta = 1$  não é apresentado como exemplo, pois o centro de massa mantém-se parado.

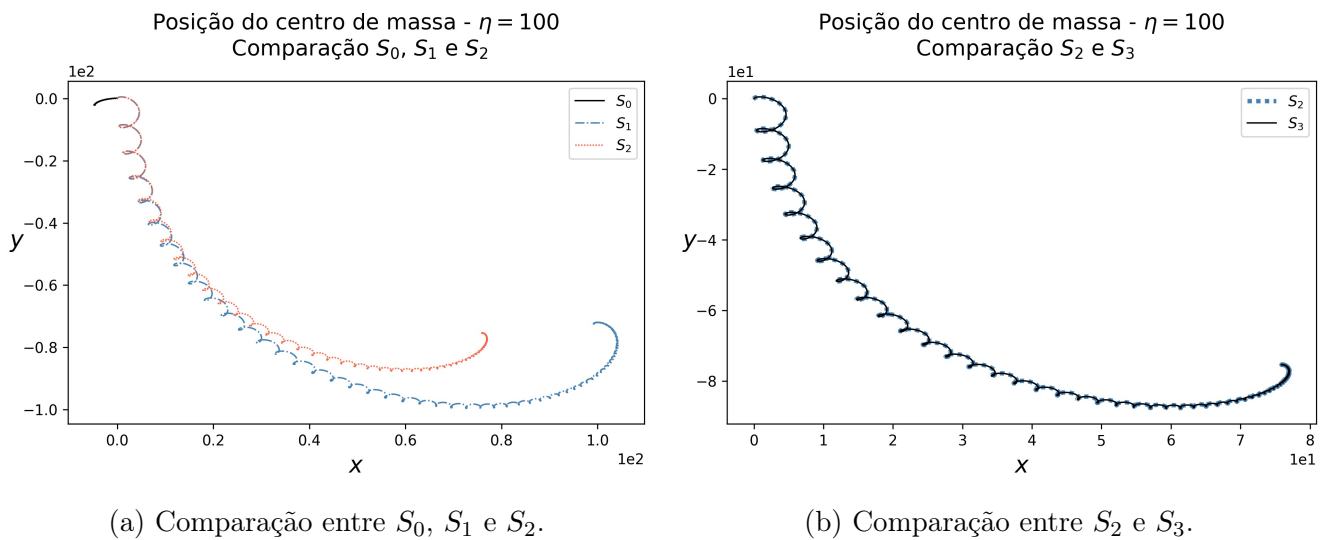


Figura 3.1.7: Comparação da trajetória do centro de massa das partículas, obtido por integração numérica dos sistemas de equações diferenciais  $S_0, S_1, S_2$  e  $S_3$  para  $\eta = 100$ . Os vetores são mostrados em coordenadas cartesianas planas  $(x, y)$ .

Podemos ver claramente nas figuras (3.1.6, 3.1.7) o movimento do centro de massa quando as cargas têm massas diferentes ( $\eta \neq 1$ ). Vemos para esse dois casos apresentados que o centro de massa tende a se mover globalmente para a direita. Para o caso  $\eta = 1$  o centro de massa está em repouso e os movimentos de carga são simétricos. Estas figuras ilustram claramente o efeito da auto-força (1.0.13) entre as cargas, conforme discutido anteriormente. Do ponto de vista da mecânica clássica tradicional, é como se a



força eletromagnética entre as cargas se comportasse como uma força externa, implicando claramente uma variação do momento linear mecânico do sistema de duas partículas.

A distância entre as partículas diminui até atingir a singularidade do sistema quando  $\mathbf{r} = 0$ . Isso é mostrado na figura (3.1.8). Com o auxílio das figuras (3.1.4) e (3.1.5), vemos que a coordenada relativa entre as partículas descreve um movimento em espiral até o colapso na singularidade. Também podemos ver que quanto maior o valor de  $\eta$ , maior o número de voltas da trajetória espiral em torno da singularidade. Esse fato expressa a propriedade de que o tempo de decaimento entre as partículas aumenta com a razão de massa entre as cargas. Observamos também com maior clareza nas figuras (3.1.8a) e (3.1.8b) a justificativa do nome de trajetória espiral quase circular, uma vez que a distância relativa diminui para a singularidade oscilando entre um valor maior e menor que também diminuem em função do tempo.

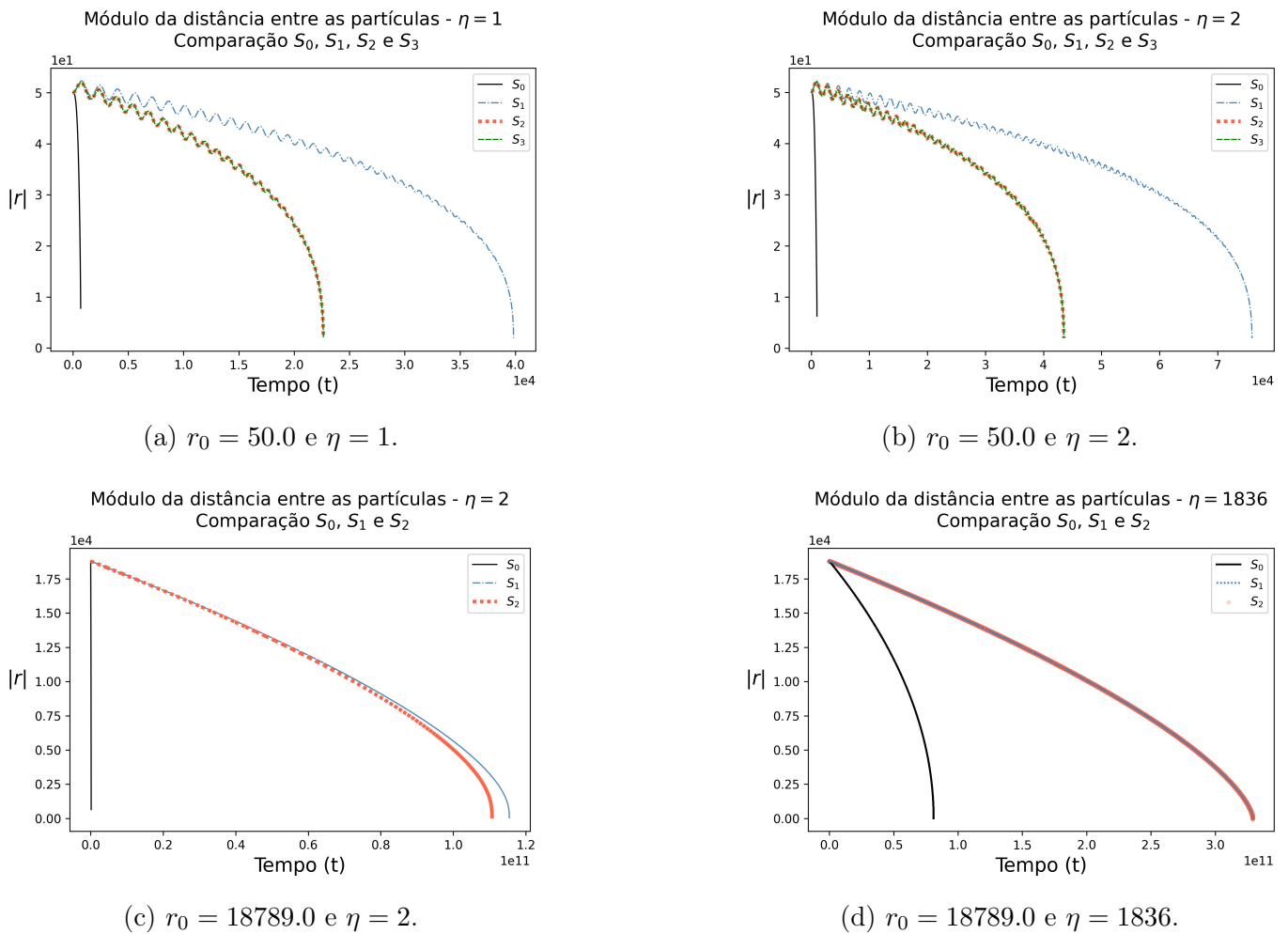
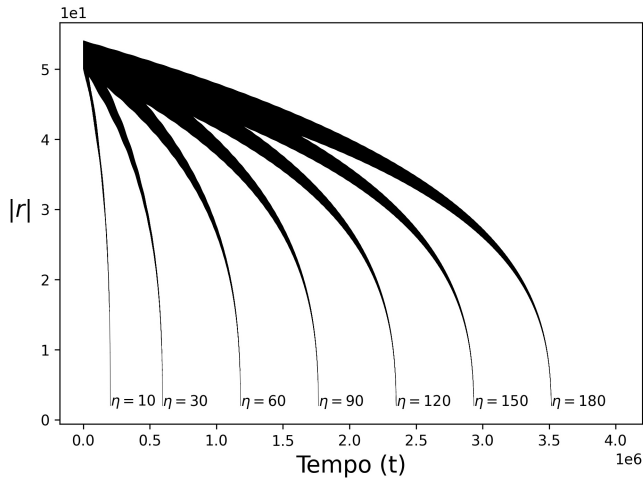


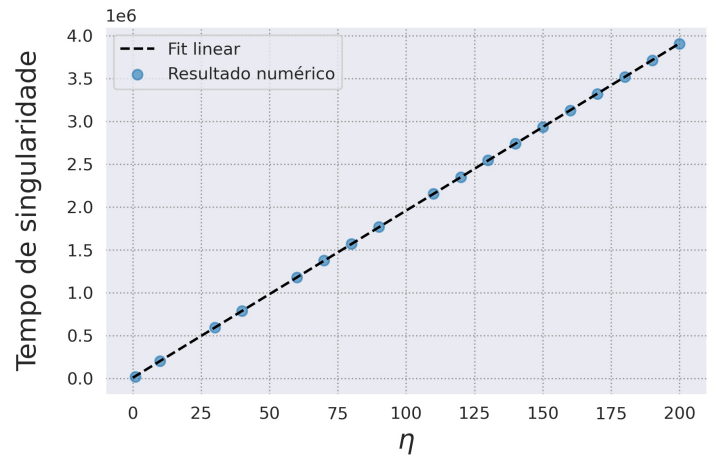
Figura 3.1.8: Comparação das distâncias relativas ( $|r|$ ), obtidas por integração numérica dos sistemas de equações diferenciais  $S_0$ ,  $S_1$ ,  $S_2$  e  $S_3$  para  $\eta$  igual a 1, 2, e 1836.

Das figuras acima e ao realizar mais integrações numéricas para diferentes valores de  $\eta$  com a mesma condição quase-circular inicial usada para obter as trajetórias das figuras (3.1.4), (3.1.5), (3.1.6) e (3.1.7), observa-se a mesma convergência para os sistemas  $S_2$  e  $S_3$ . Portanto, parece razoável utilizar o sistema  $S_2$ , uma vez que para  $n = 2$  a sequência de sistemas de equações diferenciais em (1.2.47) praticamente já convergiu. Exceto para o caso apresentado na figura (3.1.8d), em que para o raio inicial de  $r_0 = 18789$  e a razão entre as massas igual a  $\eta = 1836$  obtemos a convergência já no sistema igual a  $S_1$ .

Na figura (3.1.9) mostramos a relação entre o tempo de singularidade (estimado a partir das trajetórias numericamente integradas usando o sistema  $S_2$ ) e a razão entre as massas das cargas. Vemos claramente que existe uma relação linear entre essas grandezas, de acordo com o ajuste linear feito com os dados obtidos numericamente.



(a) Distância relativa em função do tempo.



(b) Tempos de singularidade.

Figura 3.1.9: Na figura à esquerda, mostramos a evolução da distância relativa entre as partículas  $|r|$  em função do tempo  $t$  para diferentes valores de  $\eta$ . Na figura à direita, plotamos o tempo de singularidade (obtido numericamente) em função de  $\eta$  (Pontos) e seu respectivo ajuste linear (Linha preta tracejada).

### 3.1.2 Campos eletromagnéticos simétricos

No caso de campos simétricos, usando as mesmas condições iniciais dos campos retardados da seção anterior, obtivemos uma convergência mais rápida para a sequência de sistemas de equações diferenciais em (1.2.47) e podemos observar que a diferença entre os sistemas  $S_0$  e  $S_1$  são pequenas, como mostra na figura(3.1.10).

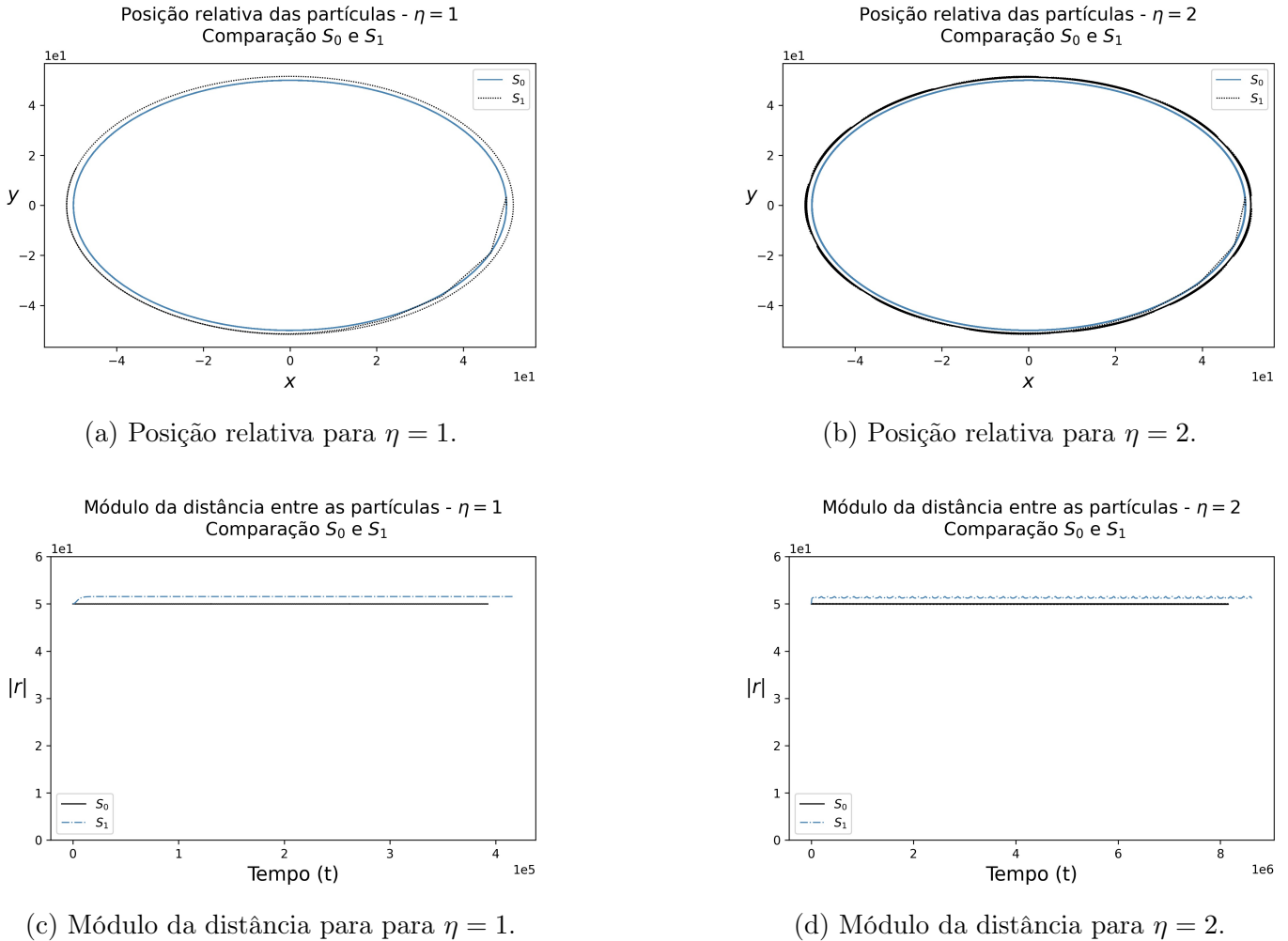


Figura 3.1.10: Trajetórias relativas planares em coordenadas cartesianas  $(x, y)$  para  $\eta = 1$  e  $\eta = 2$ , como mostrado nos gráficos (3.1.10a) e (3.1.10b), onde integramos numericamente (para a mesma condição inicial) os dois primeiros sistemas de equações diferenciais da sequência (1.2.47). Nos gráficos (3.1.10c) e (3.1.10d) temos a comparação das distâncias relativas considerando os sistemas de equações diferenciais  $S_0$  e  $S_1$ .

Assim, para analisar as trajetórias das partículas para diferentes valores de  $\eta$  iremos considerar o sistema de equações diferenciais dado por  $S_1$  da sequência (1.2.47). Vemos claramente que a trajetória das coordenadas relativas é quase circular (figuras (3.1.10a) e (3.1.10b)) e parecem oscilar entre um valor máximo e um valor mínimo, nunca decaindo para uma singularidade.

Também podemos observar o efeito da auto-força sobre o movimento do centro de massa (curva preta nas figuras (3.1.11)) para valores de  $\eta \neq 1$ . Este movimento do centro de massa parece ser dado por uma trajetória não singular que oscila entre um valor máximo e um valor mínimo. Isso demonstra que também no caso simétrico, a auto-força entre as duas cargas parece se comportar (do ponto de vista da mecânica clássica) como uma força externa que acelera o centro de massa do sistema.

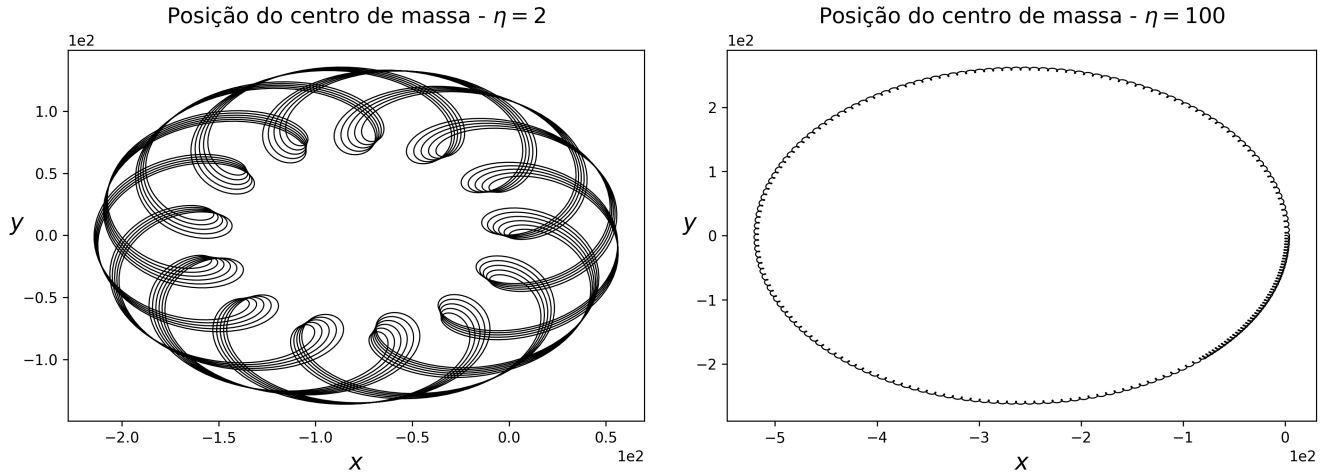


Figura 3.1.11: Trajetória do centro de massa das partículas, obtido por integração numérica dos sistemas de equações diferenciais para  $\eta = 2$  e  $\eta = 100$ . Observamos que o padrão da trajetória muda conforme o valor de  $\eta$ , porém as trajetórias parecem oscilar entre um valor máximo e um valor mínimo, nunca decaindo para uma singularidade. Os vetores são mostrados em coordenadas cartesianas planas  $(x, y)$ . A trajetória do centro de massa para o caso  $\eta = 1$  não é apresentada como exemplo, pois o centro de massa mantém-se parado.

A diferença significativa para o caso  $\alpha = 1/2$ , onde temos apenas campos de retardo, é que o sistema não evolui para uma singularidade no tempo finito e parece oscilar quase periodicamente em uma região finita do espaço (observar a figura (3.1.12)), onde vemos que as partículas descrevem loops que parecem nunca se repetir exatamente.

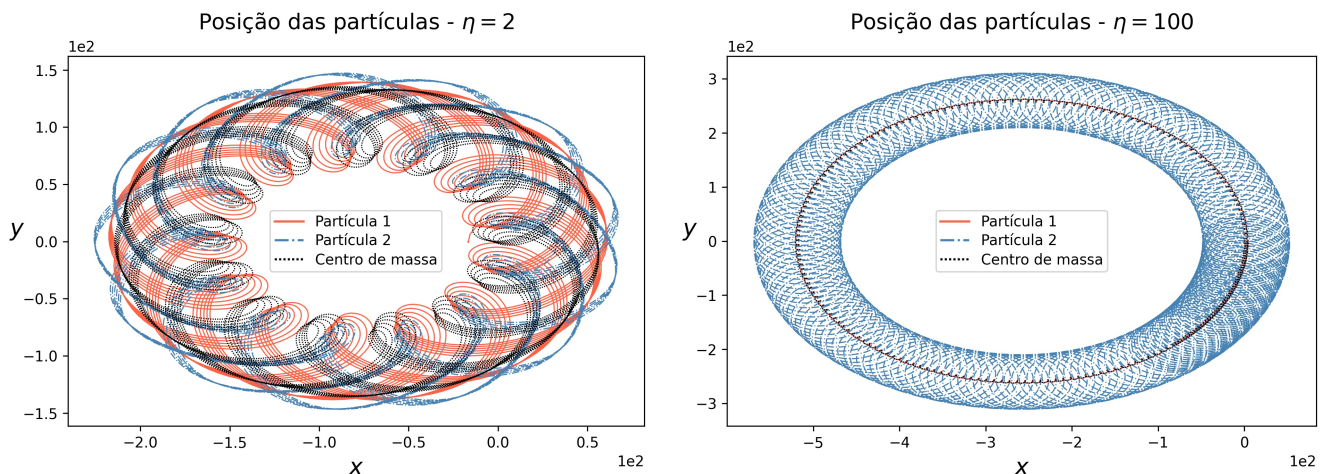


Figura 3.1.12: Trajetórias planares em coordenadas cartesianas  $(x, y)$  para os vetores  $\mathbf{r}_1$ ,  $\mathbf{r}_2$  e  $\mathbf{R}$ , onde integramos numericamente (para a mesma condição inicial) considerando  $\eta = 2$  e  $\eta = 100$ .

Nas figuras (3.1.13, 3.1.14), comparamos a evolução temporal da distância relativa entre as partículas. É bastante evidente que as trajetórias das coordenadas relativas (para os campos simétricos) são quase circulares, ou seja, a distância relativa entre as partículas oscila entre um valor máximo e um valor mínimo.

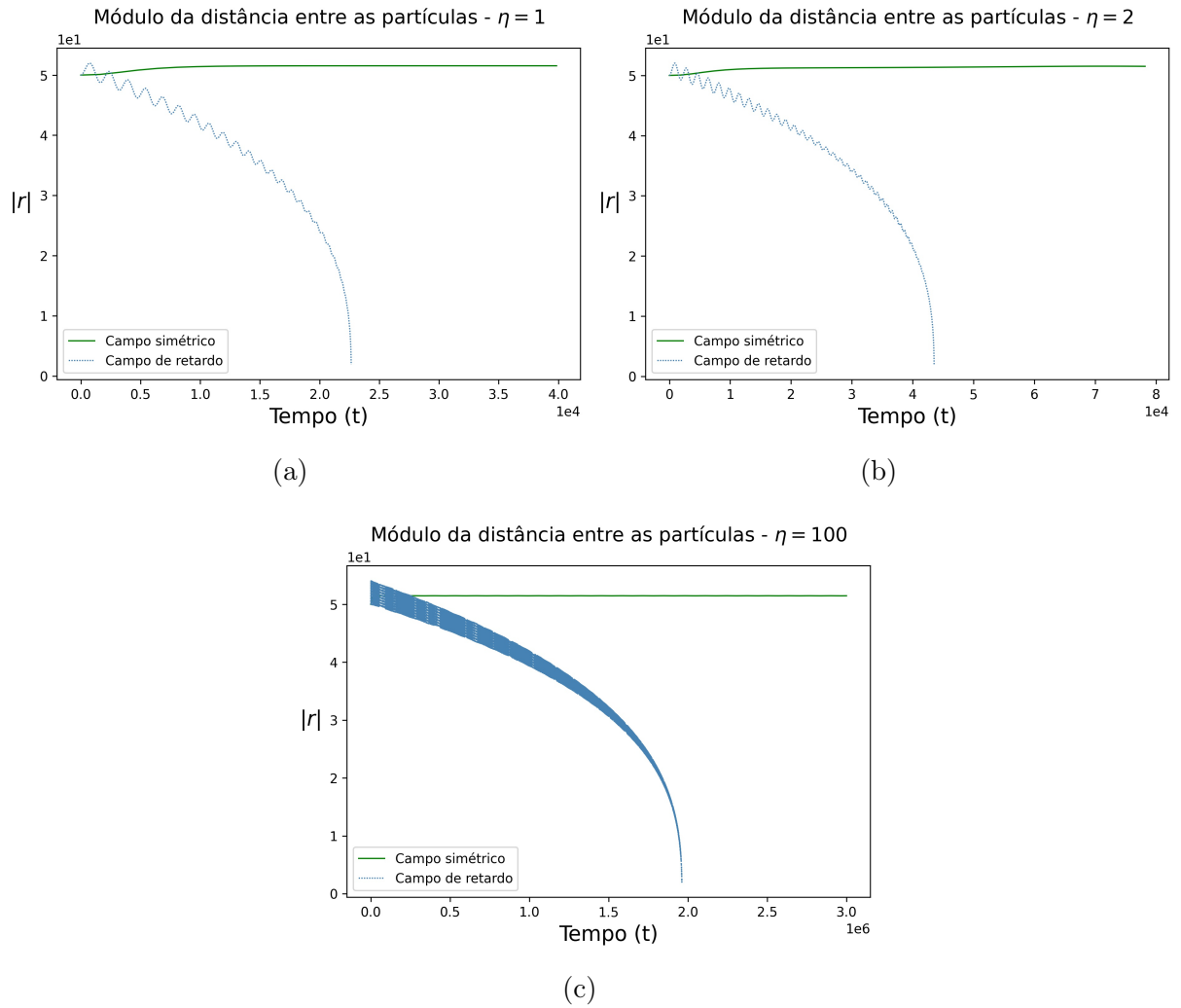


Figura 3.1.13: Comparação das distâncias relativas, obtidas por integração numérica dos sistemas de equações diferenciais considerando somente o campo retardado e o campo simétrico, para  $\eta$  igual a 1, 2 e 100 considerando o raio inicial igual a  $r_0 = 50$ .

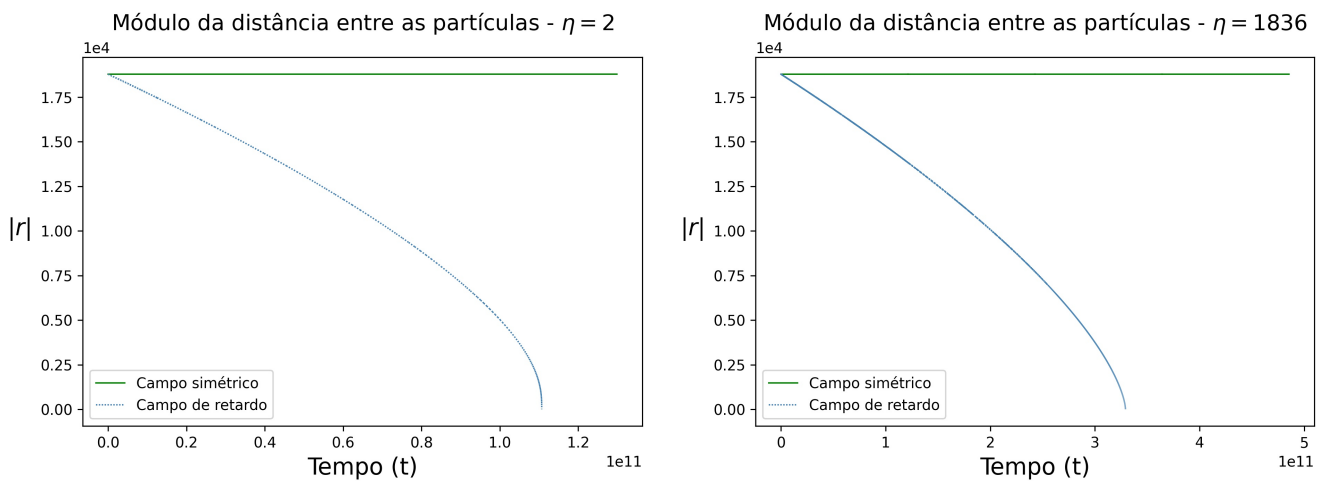


Figura 3.1.14: Comparação das distâncias relativas, obtidas por integração numérica dos sistemas de equações diferenciais considerando somente o campo retardado e o campo simétrico, para  $\eta$  igual a 2 e 1836 considerando o raio inicial igual a  $r_0 = 18789$ .

## 3.2 Trajetórias não circulares coulombianas

Iremos agora escolher as condições iniciais que não geram trajetórias circulares coulombianas. Ao aplicar uma condição inicial que não gera um movimento circular das partículas buscamos simular a trajetória das partículas considerando um condição inicial diferente daquela aplicada na seção anterior e verificar o comportamento da evolução temporal das partículas quando temos um sistema atrativo,  $S = -1$ , descrito pelas equações (0.0.32, 0.0.33).

Iremos ilustrar, nas seguintes subseções, as trajetórias para sistemas com campos eletromagnéticos retardados ( $\alpha = 1/2$ ) e para sistemas com campos eletromagnéticos simétricos ( $\alpha = 0$ ). Para todas as trajetórias mostradas nas subseções, escolhemos a distância inicial entre as partículas como sendo  $|\mathbf{r}_0| = 50$ .

Para o tempo  $t_0 = 0$  e com a distância inicial entre as partículas dada por  $r_0$  as condições para as posições e velocidades nas coordenadas cartesianas são dados por:

- Velocidade angular relativa entre as partículas no tempo inicial  $t_0$ :

$$r_0(d\theta/dt)_{t=0} = ((\eta + 1)/\eta)^{1/2}r_0^{-1/2}$$

- Condições iniciais:

$$\begin{aligned} x_1(t_0) &= -(r_0/(\eta + 1))(\sqrt{2}/2), \\ y_1(t_0) &= -(r_0/2)(1/(\eta + 1))(\sqrt{2}/2), \\ V_{1x}(t_0) &= -y_1(d\theta_0/dt), \\ V_{1y}(t_0) &= x_1(d\theta_0/dt), \\ x_2(t_0) &= (\eta r_0/(\eta + 1))(\sqrt{2}/2), \\ y_2(t_0) &= (1/2)(\eta r_0/(\eta + 1))(\sqrt{2}/2), \\ V_{2x}(t_0) &= -y_2(d\theta_0/dt), \\ V_{2y}(t_0) &= x_2(d\theta_0/dt). \end{aligned}$$

### 3.2.1 Campos Eletromagnéticos Retardados

Na figura (3.2.15) mostramos a integração numérica das trajetórias para o sistema de duas cargas com razão entre as massas iguais a  $\eta = 1$ ,  $\eta = 2$  e  $\eta = 100$ . Iremos considerar somente o caso  $n = 2$  da sequência (1.2.47), ou seja, o sistema  $S_2$ . Como visto anteriormente temos a convergência atingida rapidamente para o sistema  $S_2$  (Como podemos ver na figura (3.2.16)), portanto para os sistemas apresentados nesta subseção iremos considerar somente o valor  $n = 2$  da sequência (1.2.47).

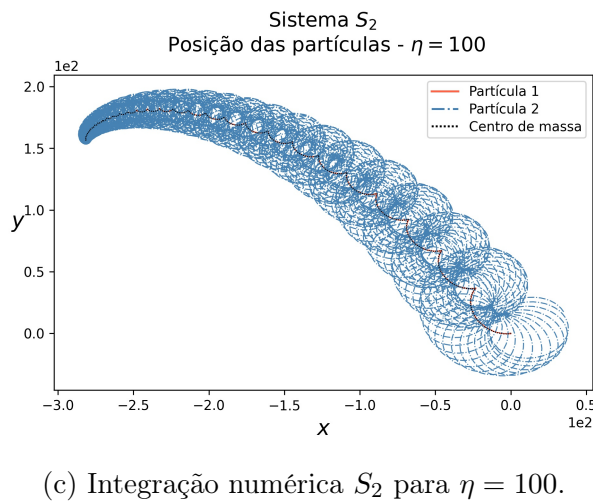
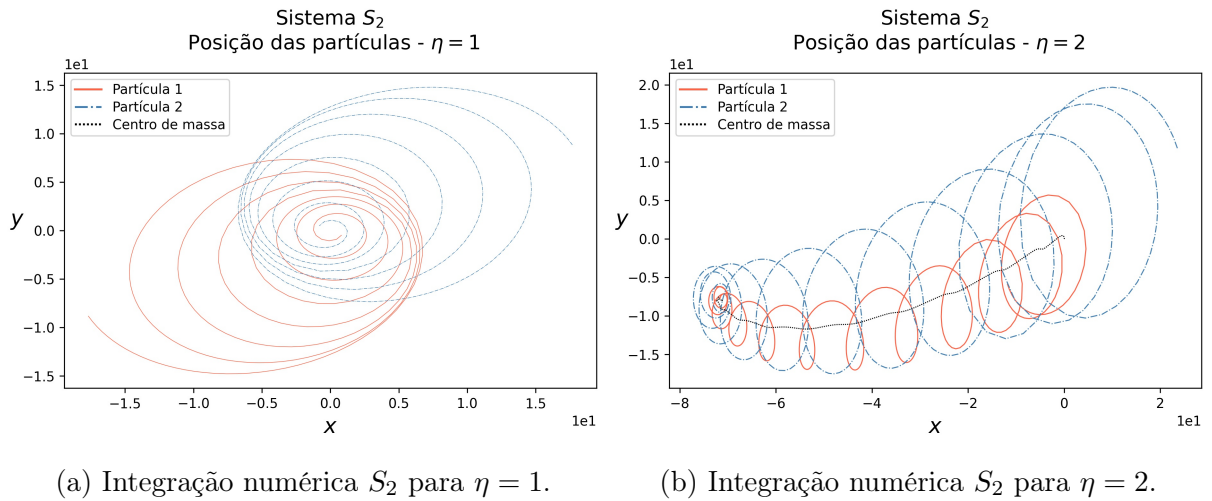


Figura 3.2.15: Trajetórias planares em coordenadas cartesianas  $(x, y)$  para os vetores  $\mathbf{r}_1$ ,  $\mathbf{r}_2$  e  $\mathbf{R}$  para o sistema  $S_2$  da sequencia (1.2.47) com a condição inicial não circular considerando  $\eta = 1$ ,  $\eta = 2$  e  $\eta = 100$ .

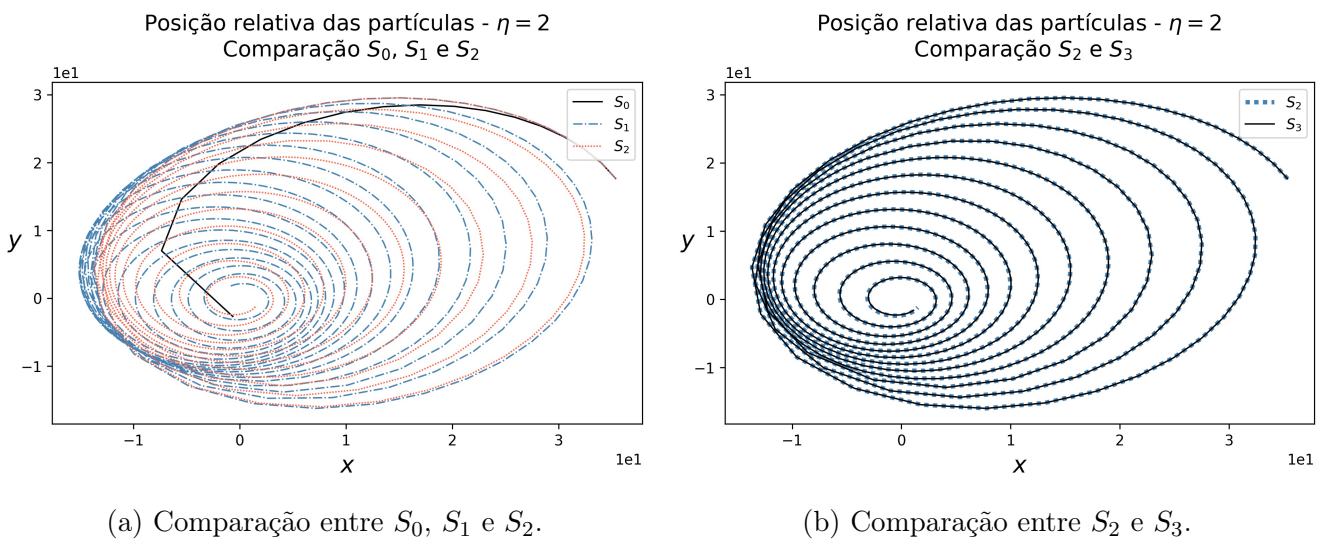


Figura 3.2.16: Comparação da trajetória relativa das partículas, obtido por integração numérica dos sistemas de equações diferenciais  $S_0$ ,  $S_1$ ,  $S_2$  e  $S_3$  para  $\eta = 2$  para uma condição inicial não circular.

Na figura (3.2.17) vemos que a coordenada relativa entre as partículas descreve um movimento em espiral, porém bem mais achatada comparada quando aplicamos uma condição inicial circular. Entretanto, a característica de colapso na singularidade permanece. Da mesma forma constatada na seção anterior, aqui também podemos ver que quanto maior o valor de  $\eta$ , maior o número de voltas da trajetória em torno da singularidade.

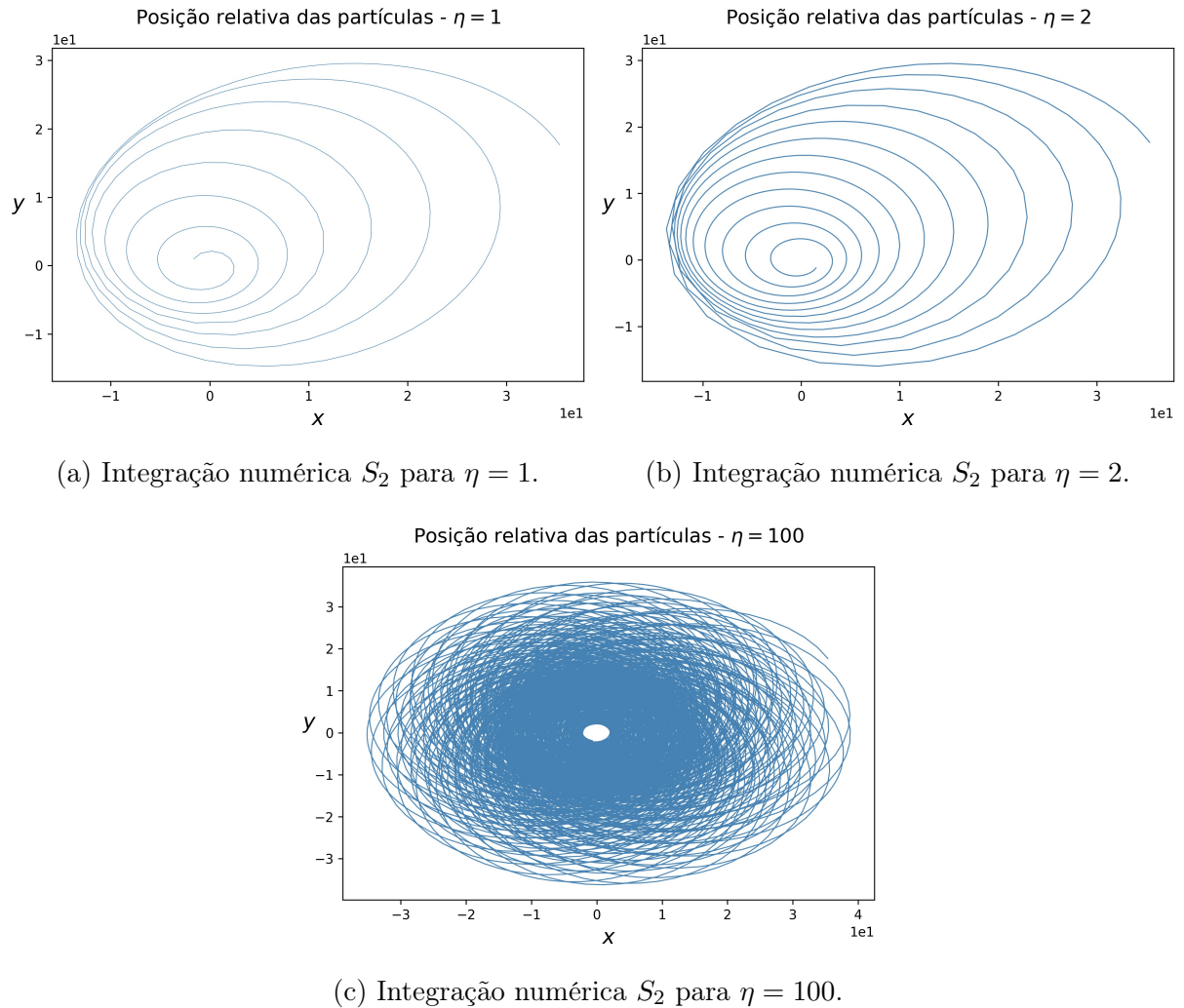


Figura 3.2.17: Trajetória relativa das partículas para  $\eta = 1$ ,  $\eta = 2$  e  $\eta = 100$  para uma condição inicial não circular.

Outra característica observada é o comportamento do movimento do centro de massa para  $\eta \neq 1$ . Na figura (3.2.18) temos para os casos  $\eta = 2$  e  $\eta = 100$  que o centro de massa tende a se mover globalmente para a esquerda. Portanto, novamente constatamos, para uma condição inicial diferente, o efeito da auto-força (1.0.13) entre as cargas.

A distância entre as partículas diminui até atingir a singularidade, como podemos ver pela figura (3.2.19). Novamente é nítido que a distância diminui para a singularidade, porém desta vez é bem mais perceptivo a oscilação entre um valor maior e menor que também diminui em função do tempo. Outra característica da mudança da condição inicial é o tempo final para a singularidade (Quando comparamos com a figura (3.1.13) para o campo retardado). Para a condição inicial não circular este tempo para a singularidade é menor comparado com os valores obtidos quando aplicamos a condição inicial circular.



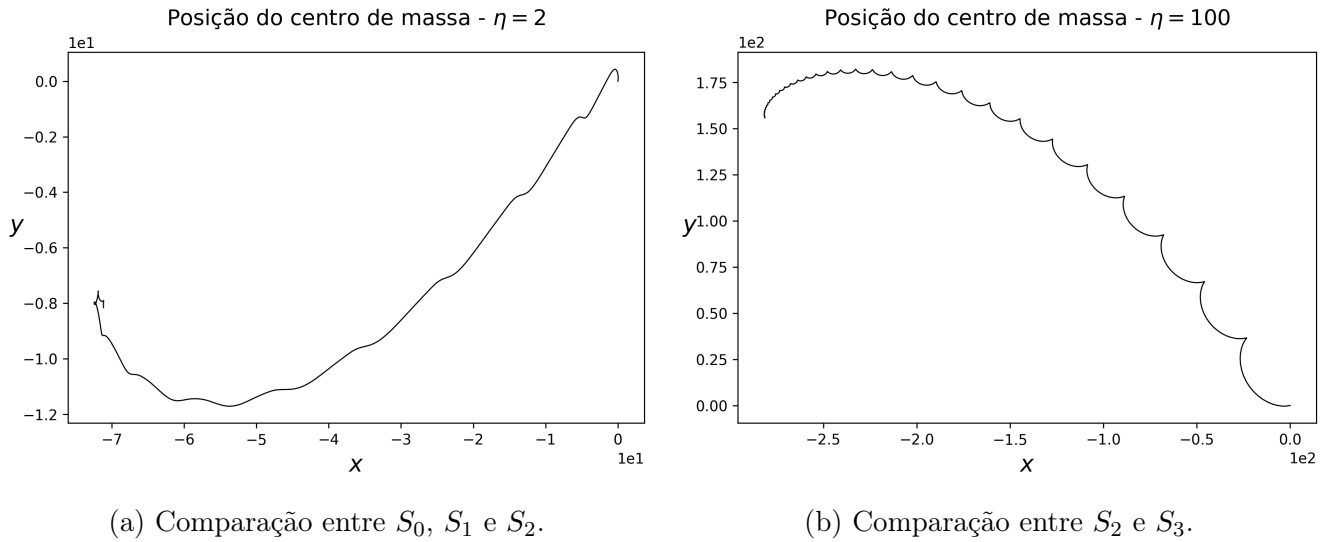


Figura 3.2.18: Trajetória do centro de massa das partículas para  $\eta = 2$  e  $\eta = 100$  para uma condição inicial circular.

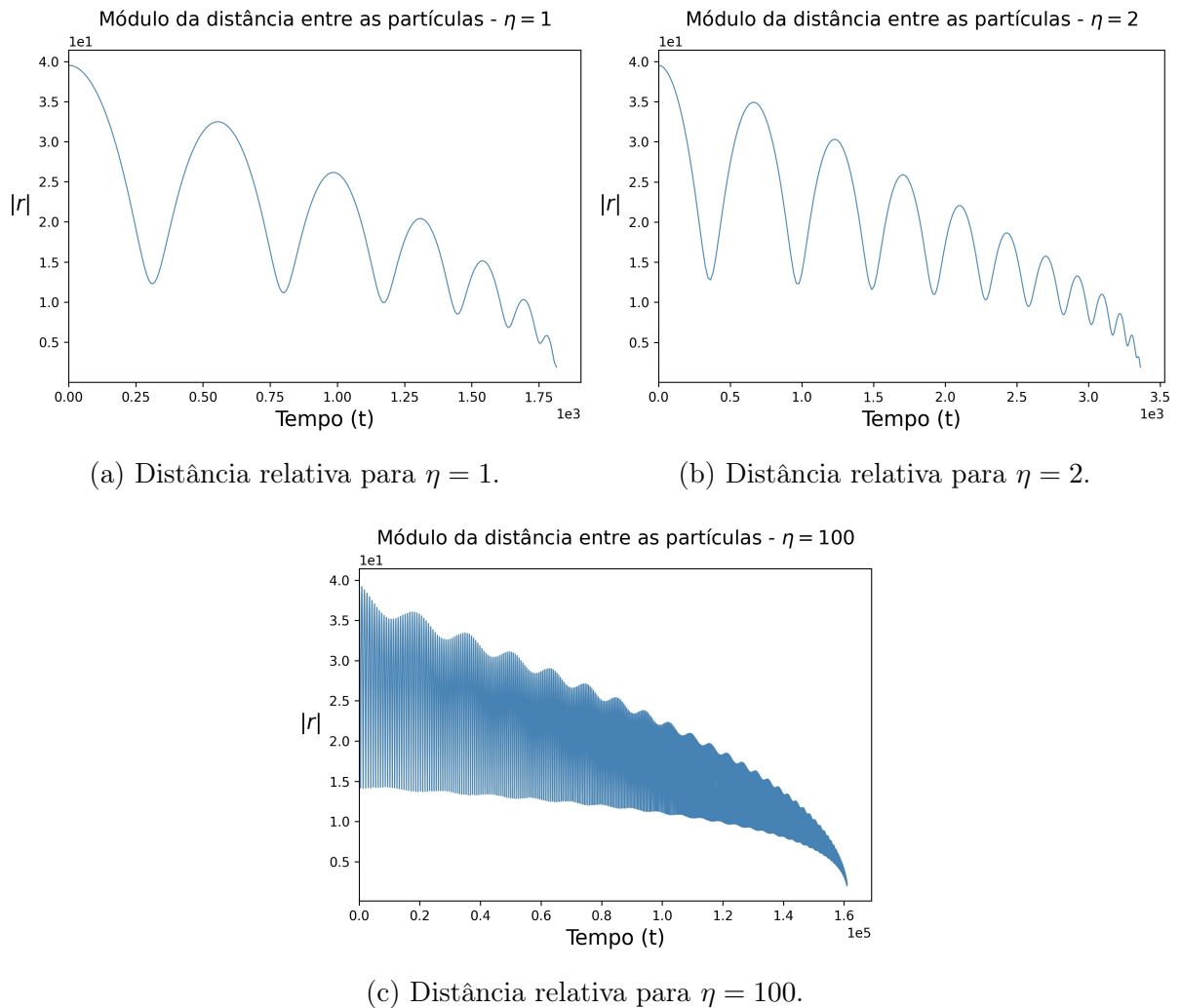
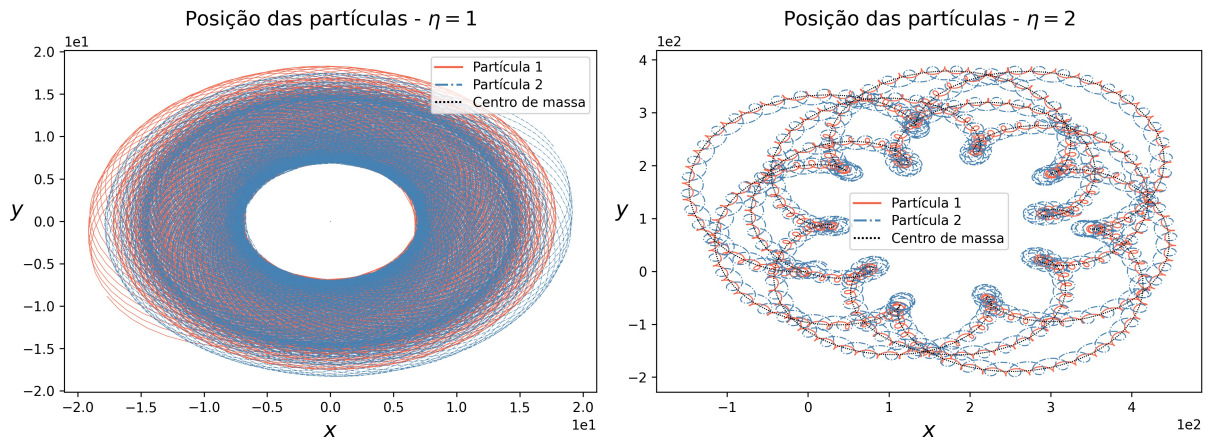


Figura 3.2.19: Distância relativa para  $\eta$  igual a 1, 2, e 100 para uma condição inicial não circular.

### 3.2.2 Campos eletromagnéticos simétricos

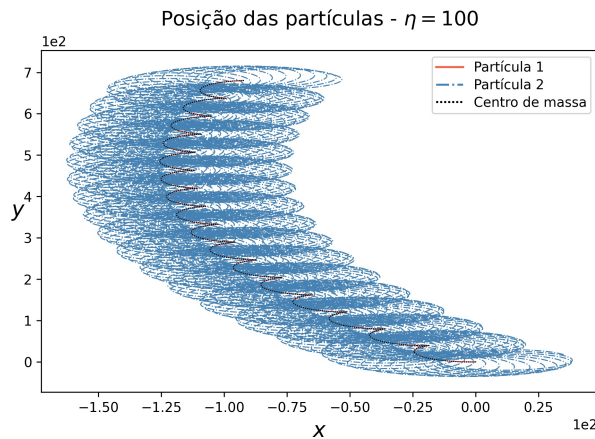
Nesta subseção iremos apresentar o caso dos campos simétricos, usando as condições iniciais não circulares de forma idêntica a subseção anterior para os campos retardados. Da mesma maneira apresentada na seção anterior para o campo simétrico com as condições iniciais circulares iremos considerar somente o caso  $n = 1$  da sequência (1.2.47). Podemos observar na figura (3.2.20) que independente da condição inicial não ser circular, ainda não é observado a singularidade para os diferentes valores de  $\eta$  apresentados. Entretanto, fica evidente o padrão das trajetórias mudam devido a condição inicial dada.

Observamos na figuras (3.2.20a, 3.2.20b) que para diferentes valores de  $\eta$  existe diferentes padrões de oscilações quase periódicos em uma região finita do espaço. No caso (3.2.20c),  $\eta = 100$ , esse padrão não é tão claro quanto os anteriores, pois seria necessário um maior tempo de simulação.



(a) Integração numérica  $S_1$  para  $\eta = 1$ .

(b) Integração numérica  $S_1$  para  $\eta = 2$ .



(c) Integração numérica  $S_1$  para  $\eta = 100$ .

Figura 3.2.20: Trajetórias planares em coordenadas cartesianas  $(x, y)$  para os vetores  $\mathbf{r}_1$ ,  $\mathbf{r}_2$  e  $\mathbf{R}$  para o sistema  $S_1$  da sequência (1.2.47) com a condição inicial não circular considerando  $\eta = 1$ ,  $\eta = 2$  e  $\eta = 100$  para o campo simétrico.

As trajetórias das coordenadas relativas, apresentadas na figura (3.2.21), apesar de não possui trajetórias quase circulares, mantém a característica de oscilação em torno de um valor máximo e um valor mínimo, nunca decaindo para a singularidade. Os *loops* que as partículas descrevem, para os casos de  $\eta$  igual a 1 e 2, parecem nunca se repetir

exatamente, entretanto para o caso  $\eta = 100$  esses *loops* parecem se repetir exatamente, porém não podemos afirmar com certeza, pois como visto anteriormente deve-se aumentar o tempo de simulação para esse caso em específico para poder observar se há um padrão de oscilação não repetitivo.

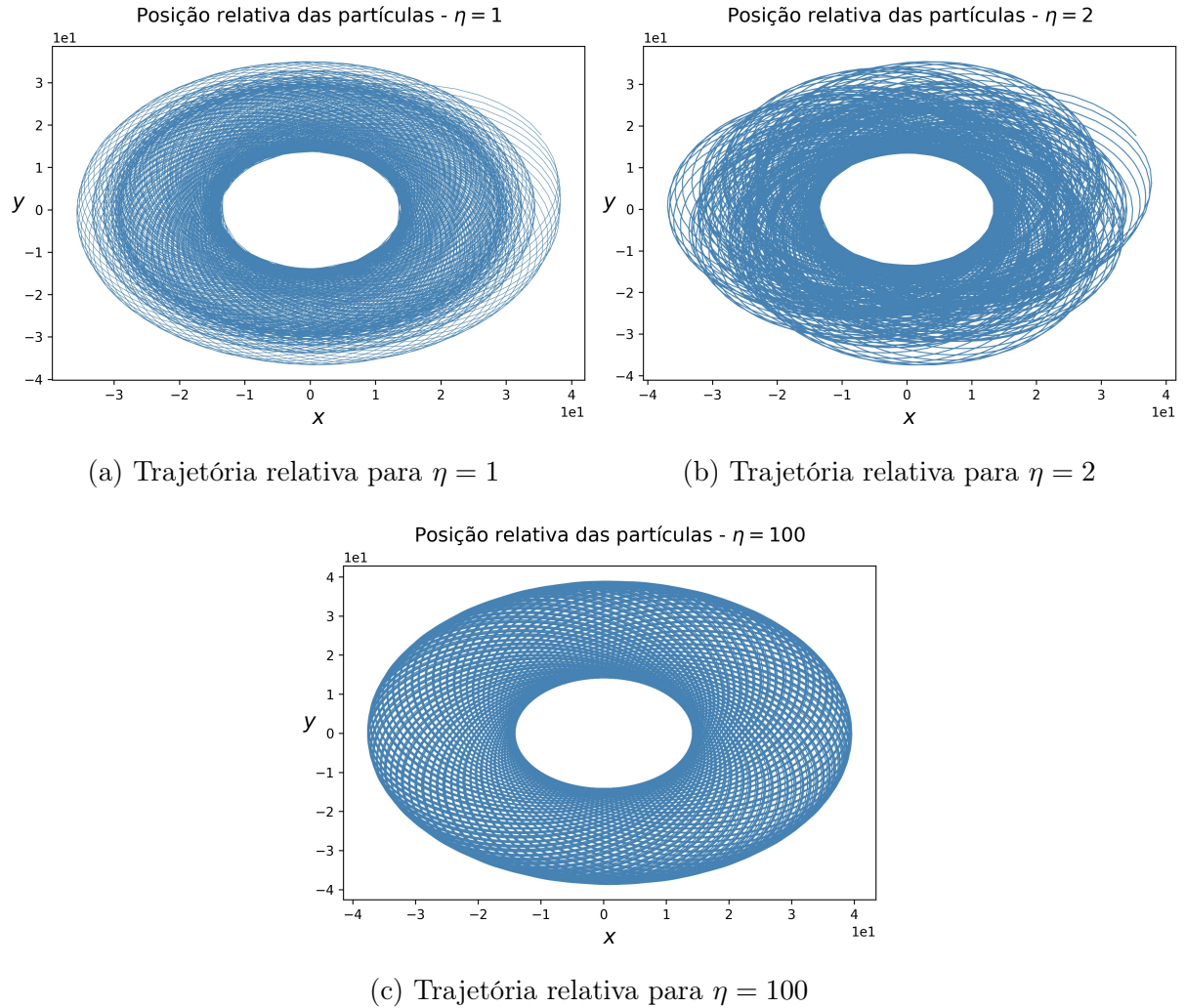
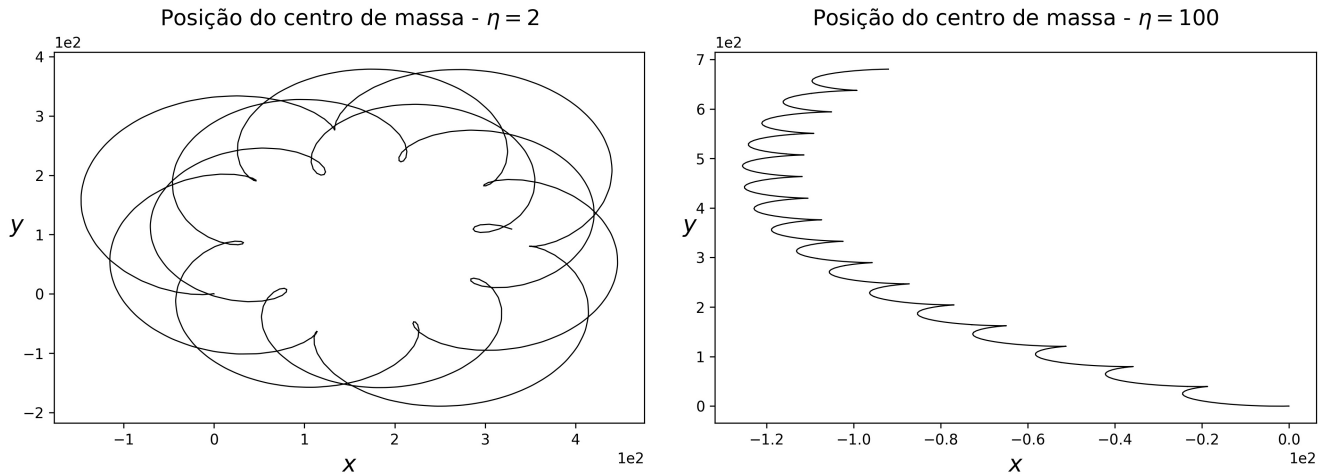


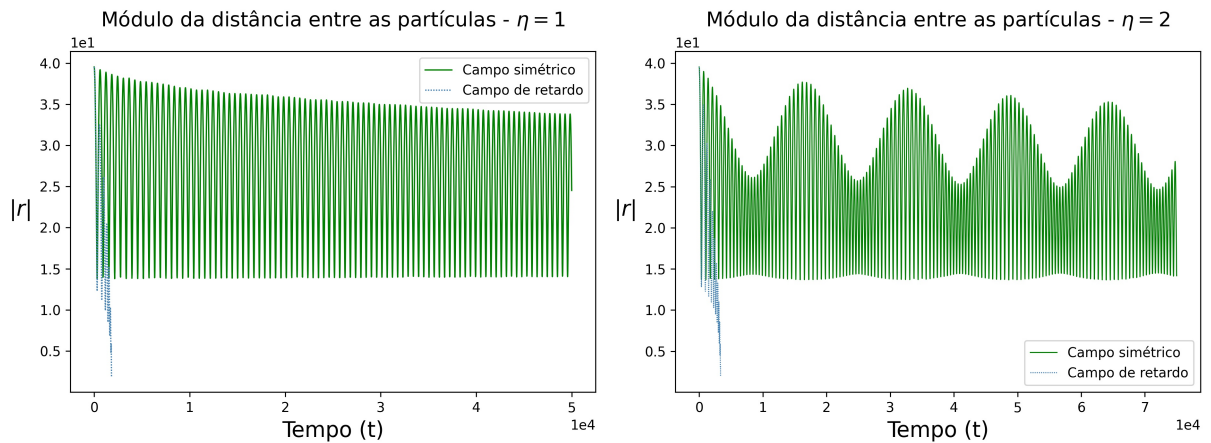
Figura 3.2.21: Trajetórias relativas para o sistema  $S_1$  da sequência (1.2.47) com a condição inicial não circular considerando  $\eta = 1$ ,  $\eta = 2$  e  $\eta = 100$  para o campo simétrico.

Considerando a trajetória do centro de massa, figura (3.2.22), observa-se novamente o efeito de auto-força para os valores de  $\eta \neq 1$ . O centro de massa para o caso  $\eta = 2$  parece ser dado por uma trajetória não singular que oscila entre um valor máximo e um valor mínimo. Para  $\eta = 100$ , como já afirmado, devemos aumentar o tempo de simulação para observar o padrão de oscilação, entretanto fica evidente que há uma oscilação entre um valor máximo e um valor mínimo.

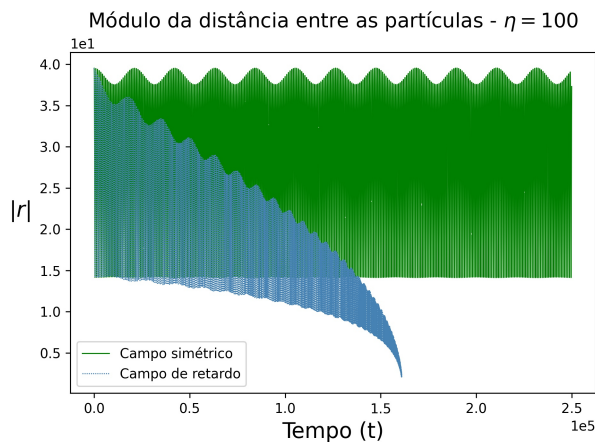


(a) Trajetória do centro de massa para  $\eta = 2$ . (b) Trajetória do centro de massa para  $\eta = 100$ .

Figura 3.2.22: Trajetórias do centro de massa para o sistema  $S_1$  da sequencia (1.2.47) com a condição inicial não circular considerando  $\eta = 2$  e  $\eta = 100$  para o campo simétrico.



(a) (b)



(c)

Figura 3.2.23: Comparação das distâncias relativas, obtidas por integração numérica dos sistemas de equações diferenciais considerando somente o campo retardado e o campo simétrico, para  $\eta$  igual a 1, 2, e 100 para uma condição inicial não circular.

Na figura (3.2.23) comparamos a evolução temporal da distância relativa entre as partículas para a condição não circular. Para o caso da condição inicial circular é evidente que as trajetórias para as coordenadas relativas (quando consideramos os campos simétricos) são quase circulares, ou seja, a distância relativa entre as partículas oscila entre um valor máximo e um valor mínimo, ver gráfico (3.2.24). Quando consideramos a condição inicial não circular a amplitude de oscilação torna-se mais proeminente. Para  $\eta = 1$  a distância relativa parece oscilar periodicamente com uma frequência bem definida, enquanto para  $\eta \neq 1$  o movimento parece quase periódico. De fato, para  $\eta \neq 1$  as oscilações de distância relativa parecem movimento de batimento, obtido a partir da superposição de ondas com frequências diferentes, mas muito próximas, porém, uma análise mais detalhada e rigorosa deve ser feita para caracterizar exatamente o tipo de movimento periódico pela coordenada relativa.

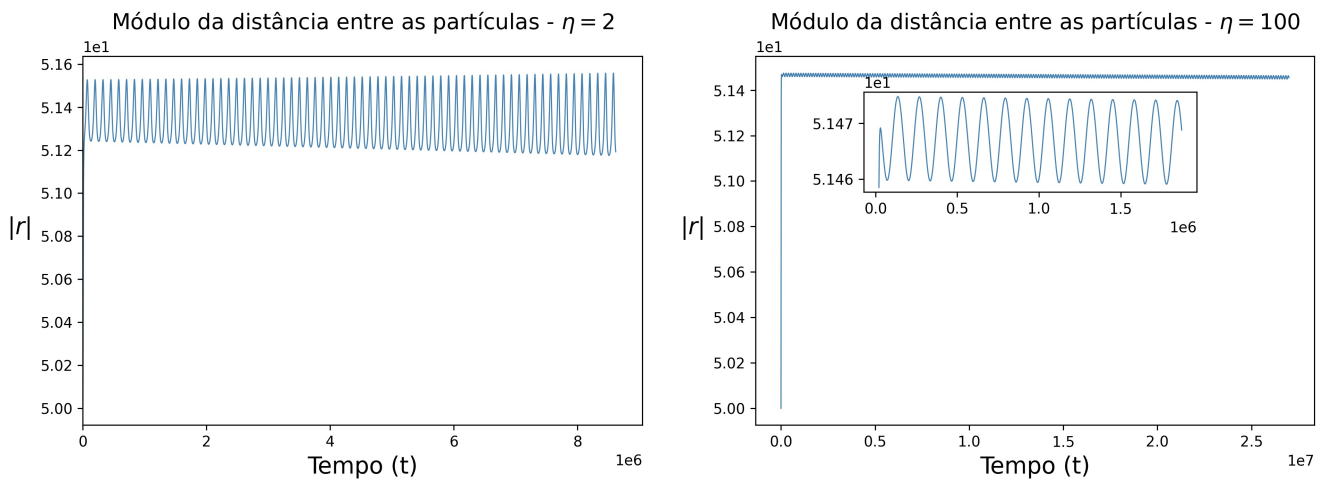
(a) Distância relativa para  $\eta = 2$ .(b) Distância relativa para  $\eta = 100$ .

Figura 3.2.24: Distâncias relativas, obtidas por integração numérica dos sistemas de equações diferenciais considerando o campo simétrico, para  $\eta$  igual a 2 e 100 para uma condição inicial circular.

# Capítulo IV

## Conclusão

### 4.1 Conclusões e considerações finais

Neste trabalho tentamos discutir os aspectos matemáticos mais relevantes associados ao problema de Synge e suas conexões com algumas questões físicas. Acima de tudo, estávamos interessados em mostrar a dificuldade de definir um sistema isolado de duas cargas, devido à presença de restrições intertemporais relacionando as posições, velocidades e acelerações de suas trajetórias.

Tentamos enfatizar, como Synge avisa, que o problema matemático de duas cargas pontuais sem auto interação é matematicamente bem definido e consistente. Isso serve para mostrar que as equações de Dirac não corrigem inconsistências matemáticas na formulação do problema de Synge, mas está ligado exclusivamente à questão da compatibilidade das equações de Synge com os princípios de conservação de energia e momento do sistema de duas cargas e seus respectivos campos eletromagnéticos, o que implica na forma como esses conceitos são definidos.

De um certo ponto de vista, podemos considerar o problema de Synge de duas cargas como mais elementar do que o problema análogo de Dirac. Uma vez que as correções da auto força para cada carga nas equações de Dirac-Maxwell podem ser derivadas através da renormalização de corpos rígidos ocupando um certo volume que deveria tender a zero, então a interação entre dois pontos no volume considerado é um problema de Synge com as forças e sua dinâmica dependem apenas da relação entre as massas (que neste caso pode ser considerada unitária). Portanto, temos um problema de Synge com forças externas que devem manter o vetor relativo das duas cargas pontuais fixo para todo o tempo [43]. Neste trabalho analisamos apenas o caso de cargas com forças Coulombianas atrativas e sem campos externos. Seria interessante aplicar o método de aproximação de Synge ao caso repulsivo com a presença de campos externos.

Além disso, este trabalho desenvolveu um método para obter soluções numéricas aproximadas para o problema Synge original que é fácil de programar (uma vez que usa apenas métodos de integração de sistemas de equações diferenciais ordinárias) e parece obter soluções consistentes com os resultados conhecidos na literatura. A metodologia proposta tornou um problema bastante complexo, não só por questões da própria natureza da qual ela é oriunda, mas da dificuldade de se resolver numericamente equações diferenciais funcionais de forma geral, na qual foi factível de ser resolvido. Um importante desenvolvimento posterior, que está além do escopo e da intenção deste trabalho, é explorar de uma forma matematicamente rigorosa a conexão clara que deve existir entre a existência de soluções Synge globais e a existência de um ponto fixo para o mapa  $\mathcal{T}$  definido na seção I.

É importante notar que fomos capazes de obter trajetórias do problema atrativo

de Synge e não apenas aproximações para as coordenadas relativas. Desta forma, para  $\alpha = 1/2$  obtivemos trajetórias espirais quase circulares que colapsam em uma singularidade e cujo tempo de decaimento para a singularidade depende linearmente da razão entre as massas. Esses resultados são consistentes com as análises teóricas feitas por Synge. Para campos simétricos ( $\alpha = 0$ ), obtivemos trajetórias quase circulares consistentes com aquelas obtidas na referência [21]. Em ambos os casos, podemos visualizar claramente os efeitos da auto força entre duas cargas pontuais.

# Referências Bibliográficas

- [1] J. L. Synge and E. T. Whittaker, “On the electromagnetic two-body problem,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 177, no. 968, pp. 118–139, 1940.
- [2] H. Lorentz, *Weiterbildung der Maxwellschen theorie: Elektronentheorie*. Encyklopädie der mathematischen wissenschaften, Teubner, 1903.
- [3] M. Abraham, *Theorie der elektrizitat*. Vol.II: Elektromagnetische theorie der strahlung, Teubner, Leipzig, 1905.
- [4] P. A. M. Dirac, “Classical theory of radiating electrons,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 167, no. 929, pp. 148–169, 1938.
- [5] L. Infeld and P. R. Wallace, “The equations of motion in electrodynamics,” *Phys. Rev.*, vol. 57, pp. 797–806, May 1940.
- [6] D.-A. Deckert and V. Hartenstein, “On the initial value formulation of classical electrodynamics,” *Journal of Physics A: Mathematical and Theoretical*, vol. 49, p. 445202, oct 2016.
- [7] J. D. Jackson, *Classical electrodynamics*. Wiley, 3rd ed ed., 1999.
- [8] G. Smith, *An Introduction to Classical Electromagnetic Radiation*. Cambridge University Press, 1997.
- [9] K. Machado, *Teoria do eletromagnetismo*. No. v. 3 in 03, UEPG, 2006.
- [10] F. Rohrlich, *Classical Charged Particles*. WORLD SCIENTIFIC, 3rd ed., 2007.
- [11] J. A. Wheeler and R. P. Feynman, “Interaction with the absorber as the mechanism of radiation,” *Rev. Mod. Phys.*, vol. 17, pp. 157–181, Apr 1945.
- [12] J. A. Wheeler and R. P. Feynman, “Classical electrodynamics in terms of direct interparticle action,” *Rev. Mod. Phys.*, vol. 21, pp. 425–433, Jul 1949.
- [13] A. Myshkis, *General Theory of Differential Equations with a Retarded Argument*. American Mathematical Society translations, American Mathematical Society, 1951.
- [14] R. D. Driver, *Ordinary and delay differential equations / R. D. Driver*. Springer-Verlag New York, 1977.
- [15] O. Diekmann, S. van Gils, S. Verduyn lunel, and H. Walther, *Delay-Equations: Functional, Complex and Nonlinear Analysis*. Netherlands: Springer, 1995.



- [16] R. D. Driver, "A two-body problem of classical electrodynamics: the one-dimensional case," *Annals of Physics*, vol. 21, no. 1, pp. 122–142, 1963.
- [17] R. D. Driver and M. J. Norris, "Note on uniqueness for a one-dimensional two-body problem of classical electrodynamics," *Annals of Physics*, vol. 42, pp. 347–351, Apr. 1967.
- [18] R. D. DRIVER, "A "backwards" two-body problem of classical relativistic electrodynamics," *Phys. Rev.*, vol. 178, pp. 2051–2057, Feb 1969.
- [19] A. Schild, "Electromagnetic two-body problem," *Phys. Rev.*, vol. 131, pp. 2762–2766, Sep 1963.
- [20] A. Schild and J. A. Schosser, "Electromagnetic two-body problem for particles with spin," *Journal of Mathematical Physics*, vol. 9, no. 6, pp. 913–915, 1968.
- [21] C. M. Andersen and H. C. von Baeyer, "Almost circular orbits in classical action-at-a-distance electrodynamics," *Phys. Rev. D*, vol. 5, pp. 802–813, Feb 1972.
- [22] C. M. Andersen and H. C. von Baeyer, "Solutions of the two-body problem in classical action-at-a-distance electrodynamics: Straight-line motion," *Phys. Rev. D*, vol. 5, pp. 2470–2476, May 1972.
- [23] V. I. Zhdanov, "On the One-Dimensional Symmetric Two-Body Problem of Classical Electrodynamics," *International Journal of Theoretical Physics*, vol. 15, pp. 157–167, Feb. 1976.
- [24] S. P. Travis, "Existence theorem for a backwards two-body problem of electrodynamics," *Phys. Rev. D*, vol. 11, pp. 292–299, Jan 1975.
- [25] D.-p. K. Hsing, "Existence and uniqueness theorem for the one-dimensional backwards two-body problem of electrodynamics," *Phys. Rev. D*, vol. 16, pp. 974–982, Aug 1977.
- [26] R. D. Driver, "Can the future influence the present?," *Phys. Rev. D*, vol. 19, pp. 1098–1107, Feb 1979.
- [27] R. D. Driver, "A neutral system with state-dependent delay," *J. Differential Equations*, vol. 54, pp. 73–86, 1984.
- [28] V. G. Angelov, "On the synge equations in a three-dimensional two-body problem of classical electrodynamics," *Journal of Mathematical Analysis and Applications*, vol. 151, no. 2, pp. 488–511, 1990.
- [29] V. G. Angelov, "Escape trajectories of j. l. synge equations," *Nonlinear Anal.: Real World Appl.*, vol. 1, p. 189–204, jun 2000.
- [30] G. Bauer, D. A. Deckert, and D. Dürr, "On the existence of dynamics in wheeler–feynman electromagnetism," *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 08 2013.
- [31] D.-A. Deckert and G. Hinrichs, "Electrodynamic two-body problem for prescribed initial data on a straight line," *Journal of Differential Equations*, vol. 260, no. 9, pp. 6900–6929, 2016.

- [32] P. Havas, “On the classical equations of motion of point charges,” *Phys. Rev.*, vol. 74, pp. 456–463, Aug 1948.
- [33] A. Sommerfeld and H. Brose, *Atomic Structure and Spectral Lines*. Dutton, 1923.
- [34] J. Franklin and C. LaMont, “On the classical equations of motion of point charges,” *Brazilian Journal of Physics*, vol. 44, pp. 1678–4448, 2014.
- [35] V. Angelov, “On the method of successive approximations for the j. l. synge electromagnetic two-body problem,” *Applied Mathematics E-Notes [electronic only]*, vol. 2, pp. 163–170, 01 2003.
- [36] J. C. Kasher and S. L. Schwebel, “Two-body problem in classical relativistic electrodynamics. i. unlike charges,” *Phys. Rev. D*, vol. 4, pp. 2956–2962, Nov 1971.
- [37] J. Huschilt, W. E. Baylis, D. Leiter, and G. Szamosi, “Numerical solutions to two-body problems in classical electrodynamics: straight-line motion with retarded fields and no radiation reaction,” *Phys. Rev., D, v. 7, no. 10, pp. 2844-2850*, 5 1973.
- [38] R. Ryne, B. Carlsten, C. Mitchell, and J. Qiang, “Self-Consistent Modeling using a Lienard-Wiechert Particle-Mesh Method,” in *Proc. 9th International Particle Accelerator Conference (IPAC’18), Vancouver, BC, Canada, April 29-May 4, 2018*, no. 9 in International Particle Accelerator Conference, (Geneva, Switzerland), pp. 3313–3315, JACoW Publishing, June 2018. <https://doi.org/10.18429/JACoW-IPAC2018-THPAK044>.
- [39] C. Mayes, “Computational approaches to coherent synchrotron radiation in two and three dimensions,” *Journal of Instrumentation*, vol. 16, p. P10010, oct 2021.
- [40] R. D. DRIVER, “A functional-differential system of neutral type arising in a two-body problem of classical electrodynamics,” in *International Symposium on Nonlinear Differential Equations and Nonlinear Mechanics* (J. P. LaSalle and S. Lefschetz, eds.), pp. 474–484, Academic Press, 1963.
- [41] J. A. Murdock, “On the well-posed two-body problem in electrodynamics and special relativity,” *Annals of Physics*, vol. 84, pp. 432–439, May 1974.
- [42] J. A. Murdock, “On the well-posed two-body problem in electrodynamics and special relativity II,” *Annals of Physics*, vol. 119, pp. 90–96, May 1979.
- [43] R. G. Beil, “Alternate formulations of classical electrodynamics,” *Phys. Rev. D*, vol. 12, pp. 2266–2268, Oct 1975.

# Apêndice A

## Equações de movimento no eixo cartesiano

Para o sistema de equações (0.0.32) e (0.0.33) iremos considerar o limite não relativístico e que os tempos  $t_r$  e  $t_a$  são infinitamente pequenos, portanto  $t_r = t_a = t$ . Realizar esta aproximação da instantaneidade implica no acoplamento entre a aceleração no tempo  $t$ , proveniente da força newtoniana, com as acelerações nos tempos  $t_r$  e  $t_a$  dos campos eletromagnéticos. Sendo assim, obtemos um sistema de equações diferenciais de segunda ordem que pode ser dado na forma matricial:

$$\mathcal{M}\mathcal{A} = \mathcal{B}, \quad (1.0.1)$$

Definindo inicialmente as variáveis:

$$\xi_1 = 1/2 + \alpha, \quad (1.0.2)$$

$$\xi_2 = 1/2 - \alpha, \quad (1.0.3)$$

$$\zeta_{11} = (|eX V_{1x} + eY V_{1y} + eZ V_{1z} + 1|)^3, \quad (1.0.4)$$

$$\zeta_{12} = (|eX V_{1x} + eY V_{1y} + eZ V_{1z} - 1|)^3, \quad (1.0.5)$$

$$\zeta_{21} = (|eX V_{2x} + eY V_{2y} + eZ V_{2z} + 1|)^3, \quad (1.0.6)$$

$$\zeta_{22} = (|eX V_{2x} + eY V_{2y} + eZ V_{2z} - 1|)^3, \quad (1.0.7)$$

podemos escrever os elementos da matriz  $\mathcal{M}$  como:

$$m_{11} = \eta, \quad (1.0.8)$$

$$m_{12} = 0, \quad (1.0.9)$$

$$m_{13} = 0, \quad (1.0.10)$$

$$\begin{aligned} m_{14} = & -\frac{S}{r\zeta_{22}\zeta_{21}} \left( ((V_{1z}V_{2z} - 1)eY^2 + ((-V_{1y}V_{2z} - V_{1z}V_{2y})eZ + V_{1y} + V_{2y})eY \right. \\ & + (V_{1y}V_{2y} - 1)eZ^2 + (V_{1z} + V_{2z})eZ - V_{1z}V_{2z} - V_{1y}V_{2y})\xi_2\zeta_{21} \\ & + \zeta_{22}\xi_1 \left( (V_{1z}V_{2z} - 1)eY^2 + ((-V_{1y}V_{2z} - V_{1z}V_{2y})eZ - V_{1y} - V_{2y})eY \right. \\ & \left. \left. + (V_{1y}V_{2y} - 1)eZ^2 + (-V_{2z} - V_{1z})eZ - V_{1z}V_{2z} - V_{1y}V_{2y} \right) \right), \quad (1.0.11) \end{aligned}$$

$$\begin{aligned} m_{15} = & +\frac{S}{r\zeta_{22}\zeta_{21}} \left( ((V_{2x}eZ^2 - V_{2z}eXeZ - V_{2x} + eX)V_{1y} \right. \\ & + ((-V_{1z}eZ + 1)V_{2x} + eX(V_{1z}V_{2z} - 1))eY)\xi_2\zeta_{21} \\ & + \zeta_{22}\xi_1 \left( (V_{2x}eZ^2 - V_{2z}eXeZ - V_{2x} - eX)V_{1y} \right. \\ & \left. \left. + ((-V_{1z}eZ - 1)V_{2x} + eX(V_{1z}V_{2z} - 1))eY \right) \right), \quad (1.0.12) \end{aligned}$$

$$\begin{aligned} m_{16} = & -\frac{S}{r\zeta_{22}\zeta_{21}} \left( ((-V_{2x}eY^2 + V_{2y}eXeY + V_{2x} - eX)V_{1z} \right. \\ & - ((-V_{1y}eY + 1)V_{2x} + eX(V_{1y}V_{2y} - 1))eZ)\xi_2\zeta_{21} \\ & + \zeta_{22}\xi_1 \left( (-V_{2x}eY^2 + V_{2y}eXeY + V_{2x} + eX)V_{1z} \right. \\ & \left. \left. - ((-V_{1y}eY - 1)V_{2x} + eX(V_{1y}V_{2y} - 1))eZ \right) \right), \quad (1.0.13) \end{aligned}$$

$$m_{21} = 0, \quad (1.0.14)$$

$$m_{22} = \eta, \quad (1.0.15)$$

$$m_{23} = 0, \quad (1.0.16)$$

$$\begin{aligned} m_{24} = & +\frac{S}{r\zeta_{22}\zeta_{21}} \left( -V_{1x}V_{2y}(\xi_1\zeta_{22} + \xi_2\zeta_{21})eX^2 \right. \\ & + ((\xi_1\zeta_{22} + \xi_2\zeta_{21})(V_{1z}V_{2z} - 1)eY - V_{2y}(\xi_2(V_{1z}eZ - 1)\zeta_{21} + \xi_1\zeta_{22}(V_{1z}eZ + 1)))eX \\ & \left. - V_{1x}eY(V_{2y}(\xi_1\zeta_{22} + \xi_2\zeta_{21})eY + \xi_2(V_{2z}eZ - 1)\zeta_{21} + \xi_1\zeta_{22}(V_{2z}eZ + 1)) \right), \quad (1.0.17) \end{aligned}$$

$$\begin{aligned} m_{25} = & +\frac{S}{r\zeta_{22}\zeta_{21}} \left( ((V_{2z}(V_{1x}eX - 1) + V_{1z}(V_{2x}eX - 1) - V_{2x}V_{1x}eZ)eZ \right. \\ & + V_{1z}V_{2z}(eY^2 + eZ^2) + (V_{2x} - eX)V_{1x} - V_{2x}eX - eY^2 + 1)\xi_2\zeta_{21} \\ & + \xi_1\zeta_{22} \left( ((V_{2z}(V_{1x}eX + 1) + V_{1z}(V_{2x}eX + 1) - V_{2x}V_{1x}eZ)eZ \right. \\ & \left. \left. + V_{1z}V_{2z}(eY^2 + eZ^2) + (V_{2x} + eX)V_{1x} + V_{2x}eX - eY^2 + 1) \right) \right), \quad (1.0.18) \end{aligned}$$

$$\begin{aligned} m_{26} = & -\frac{S}{r\zeta_{22}\zeta_{21}} \left( V_{1z}V_{2y}(\xi_1\zeta_{22} + \xi_2\zeta_{21})eY^2 \right. \\ & + (eZ(1 - V_{2x}V_{1x})(\xi_1\zeta_{22} + \xi_2\zeta_{21}) + (\xi_2(V_{2x}eX - 1)\zeta_{21} + \xi_1\zeta_{22}(V_{2x}eX + 1))V_{1z})eY \\ & \left. + V_{2y}(eZ(\xi_1\zeta_{22} + \xi_2\zeta_{21})V_{1z} + \xi_2(V_{1x}eX - 1)\zeta_{21} + \xi_1\zeta_{22}(V_{1x}eX + 1))eZ \right), \quad (1.0.19) \end{aligned}$$

$$m_{31} = 0, \quad (1.0.20)$$

$$m_{32} = 0, \quad (1.0.21)$$

$$m_{33} = \eta, \quad (1.0.22)$$

$$\begin{aligned} m_{34} = & -\frac{S}{r\zeta_{22}\zeta_{21}} (V_{2z} V_{1x} (\xi_1 \zeta_{22} + \xi_2 \zeta_{21}) eX^2 \\ & + ((1 - V_{1y} V_{2y}) (\xi_1 \zeta_{22} + \xi_2 \zeta_{21}) eZ + (\xi_2 (V_{1y} eY - 1) \zeta_{21} + \xi_1 \zeta_{22} (V_{1y} eY + 1)) V_{2z}) eX \\ & + V_{1x} (V_{2z} (\xi_1 \zeta_{22} + \xi_2 \zeta_{21}) eZ + \xi_2 (V_{2y} eY - 1) \zeta_{21} + \xi_1 \zeta_{22} (V_{2y} eY + 1)) eZ), \end{aligned} \quad (1.0.23)$$

$$\begin{aligned} m_{35} = & -\frac{S}{r\zeta_{22}\zeta_{21}} (V_{1y} V_{2z} (\xi_1 \zeta_{22} + \xi_2 \zeta_{21}) eY^2 \\ & + (eZ (1 - V_{1x} V_{2x}) (\xi_1 \zeta_{22} + \xi_2 \zeta_{21}) + V_{2z} (\xi_2 (V_{1x} eX - 1) \zeta_{21} + \xi_1 \zeta_{22} (V_{1x} eX + 1))) eY \\ & + V_{1y} (V_{2z} (\xi_1 \zeta_{22} + \xi_2 \zeta_{21}) eZ + \xi_2 (V_{2x} eX - 1) \zeta_{21} + \xi_1 \zeta_{22} (V_{2x} eX + 1)) eZ), \end{aligned} \quad (1.0.24)$$

$$\begin{aligned} m_{36} = & +\frac{S}{r\zeta_{22}\zeta_{21}} (((V_{2y} (V_{1x} eX - 1) + V_{1y} (V_{2x} eX - 1) - V_{2x} V_{1x} eY) eY \\ & + V_{2y} V_{1y} (eY^2 + eZ^2) + (V_{2x} - eX) V_{1x} - V_{2x} eX - eZ^2 + 1) \xi_2 \zeta_{21} \\ & + \xi_1 \zeta_{22} ((V_{2y} (V_{1x} eX + 1) + V_{1y} (V_{2x} eX + 1) - V_{2x} V_{1x} eY) eY \\ & + V_{2y} V_{1y} (eY^2 + eZ^2) + (V_{2x} + eX) V_{1x} + V_{2x} eX - eZ^2 + 1)), \end{aligned} \quad (1.0.25)$$

$$\begin{aligned} m_{41} = & -\frac{S}{r\zeta_{11}\zeta_{12}} (((-eZ (V_{1y} V_{2z} + V_{1z} V_{2y}) + V_{1y} + V_{2y} + (V_{1z} V_{2z} - 1) eY) eY \\ & + (V_{1z} + V_{2z} + (V_{1y} V_{2y} - 1) eZ) eZ - V_{1y} V_{2y} - V_{1z} V_{2z}) \xi_1 \zeta_{11} \\ & + ((-eZ (V_{1y} V_{2z} + V_{1z} V_{2y}) - V_{1y} - V_{2y} + (V_{1z} V_{2z} - 1) eY) eY \\ & + (-V_{2z} - V_{1z} + (V_{1y} V_{2y} - 1) eZ) eZ - V_{1y} V_{2y} - V_{1z} V_{2z}) \xi_2 \zeta_{12}), \end{aligned} \quad (1.0.26)$$

$$\begin{aligned} m_{42} = & +\frac{S}{r\zeta_{11}\zeta_{12}} (((V_{2y} eZ^2 - V_{2z} eY eZ - V_{2y} + eY) V_{1x} \\ & + ((-V_{1z} eZ + 1) V_{2y} + (V_{1z} V_{2z} - 1) eY) eX) \xi_1 \zeta_{11} \\ & + \xi_2 \zeta_{12} ((V_{2y} eZ^2 - V_{2z} eY eZ - V_{2y} - eY) V_{1x} \\ & + ((-V_{1z} eZ - 1) V_{2y} + (V_{1z} V_{2z} - 1) eY) eX)), \end{aligned} \quad (1.0.27)$$

$$\begin{aligned} m_{43} = & -\frac{S}{r\zeta_{11}\zeta_{12}} (((V_{2y} eY eZ - V_{2z} eY^2 + V_{2z} - eZ) V_{1x} \\ & eX (V_{2z} (V_{1y} eY - 1) + (-V_{1y} V_{2y} + 1) eZ)) \xi_1 \zeta_{11} \\ & + \xi_2 \zeta_{12} ((V_{2y} eY eZ - V_{2z} eY^2 + V_{2z} + eZ) V_{1x} \\ & + (V_{2z} (V_{1y} eY + 1) + (-V_{1y} V_{2y} + 1) eZ) eX)), \end{aligned} \quad (1.0.28)$$

$$m_{44} = 1, \quad (1.0.29)$$

$$m_{45} = 0, \quad (1.0.30)$$

$$m_{46} = 0, \quad (1.0.31)$$

$$\begin{aligned}
 m_{51} = & \frac{S}{r\zeta_{11}\zeta_{12}} \left( -V_{1y}V_{2x} (\xi_1\zeta_{11} + \xi_2\zeta_{12}) eX^2 \right. \\
 & + ((\xi_1\zeta_{11} + \xi_2\zeta_{12}) (V_{1z}V_{2z} - 1) eY - V_{1y} (\xi_1 (V_{2z}eZ - 1)\zeta_{11} + \xi_2\zeta_{12} (V_{2z}eZ + 1))) eX \\
 & \left. - V_{2x} (V_{1y} (\xi_1\zeta_{11} + \xi_2\zeta_{12}) eY + \xi_1 (V_{1z}eZ - 1)\zeta_{11} + \xi_2\zeta_{12} (V_{1z}eZ + 1)) eY \right), \quad (1.0.32)
 \end{aligned}$$

$$\begin{aligned}
 m_{52} = & \frac{S}{r\zeta_{11}\zeta_{12}} (((V_{2z} (V_{1x}eX - 1) + V_{1z} (V_{2x}eX - 1) - V_{2x}V_{1x}eZ) eZ \\
 & + V_{1z}V_{2z} (eY^2 + eZ^2) + (V_{2x} - eX) V_{1x} - V_{2x}eX - eY^2 + 1) \xi_1\zeta_{11} \\
 & + \xi_2\zeta_{12} ((V_{2z} (V_{1x}eX + 1) + V_{1z} (V_{2x}eX + 1) - V_{2x}V_{1x}eZ) eZ) \\
 & + V_{1z}V_{2z} (eY^2 + eZ^2) + (V_{2x} + eX) V_{1x} + V_{2x}eX - eY^2 + 1), \quad (1.0.33)
 \end{aligned}$$

$$\begin{aligned}
 m_{53} = & -\frac{S}{r\zeta_{11}\zeta_{12}} (V_{2z}V_{1y} (\xi_1\zeta_{11} + \xi_2\zeta_{12}) eY^2 \\
 & + (eZ (1 - V_{1x}V_{2x}) (\xi_1\zeta_{11} + \xi_2\zeta_{12}) + (\xi_1 (V_{1x}eX - 1)\zeta_{11} + \xi_2\zeta_{12} (V_{1x}eX + 1)) V_{2z}) eY \\
 & + (eZ (\xi_1\zeta_{11} + \xi_2\zeta_{12}) V_{2z} + \xi_1 (V_{2x}eX - 1)\zeta_{11} + \xi_2\zeta_{12} (V_{2x}eX + 1)) V_{1y} eZ), \quad (1.0.34)
 \end{aligned}$$

$$m_{54} = 0, \quad (1.0.35)$$

$$m_{55} = 1, \quad (1.0.36)$$

$$m_{56} = 0, \quad (1.0.37)$$

$$\begin{aligned}
 m_{61} = & -\frac{S}{r\zeta_{11}\zeta_{12}} (V_{1z}V_{2x} (\xi_1\zeta_{11} + \xi_2\zeta_{12}) eX^2 \\
 & + ((\xi_1\zeta_{11} + \xi_2\zeta_{12}) (1 - V_{1y}V_{2y}) eZ + (\xi_1 (V_{2y}eY - 1)\zeta_{11} + \xi_2\zeta_{12} (V_{2y}eY + 1)) V_{1z}) eX \\
 & + V_{2x} (V_{1z} (\xi_1\zeta_{11} + \xi_2\zeta_{12}) eZ + \xi_1 (V_{1y}eY - 1)\zeta_{11} + \xi_2\zeta_{12} (V_{1y}eY + 1)) eZ), \quad (1.0.38)
 \end{aligned}$$

$$\begin{aligned}
 m_{62} = & -\frac{S}{r\zeta_{11}\zeta_{12}} (V_{1z}V_{2y} (\xi_1\zeta_{11} + \xi_2\zeta_{12}) eY^2 \\
 & + (eZ (1 - V_{1x}V_{2x}) (\xi_1\zeta_{11} + \xi_2\zeta_{12}) + V_{1z} (\xi_1 (V_{2x}eX - 1)\zeta_{11} + \xi_2\zeta_{12} (V_{2x}eX + 1))) eY \\
 & + (V_{1z} (\xi_1\zeta_{11} + \xi_2\zeta_{12}) eZ + \xi_1 (V_{1x}eX - 1)\zeta_{11} + \xi_2\zeta_{12} (V_{1x}eX + 1)) V_{2y} eZ), \quad (1.0.39)
 \end{aligned}$$

$$\begin{aligned}
 m_{63} = & \frac{S}{r\zeta_{11}\zeta_{12}} (((V_{2y} (V_{1x}eX - 1) + V_{1y} (V_{2x}eX - 1) - V_{2x}V_{1x}eY) eY \\
 & + V_{2y}V_{1y} (eY^2 + eZ^2) + (V_{2x} - eX) V_{1x} - V_{2x}eX - eZ^2 + 1) \xi_1\zeta_{11} \\
 & + \xi_2\zeta_{12} ((V_{2y} (V_{1x}eX + 1) + V_{1y} (V_{2x}eX + 1) - V_{2x}V_{1x}eY) eY \\
 & + V_{2y}V_{1y} (eY^2 + eZ^2) + (V_{2x} + eX) V_{1x} + V_{2x}eX - eZ^2 + 1)), \quad (1.0.40)
 \end{aligned}$$

$$m_{64} = 0, \quad (1.0.41)$$

$$m_{65} = 0, \quad (1.0.42)$$

$$m_{66} = 1. \quad (1.0.43)$$

Os elementos da matriz  $\mathcal{B}$  são dados por:

$$b_1 = -\frac{S(-V_{2x}^2 - V_{2y}^2 - V_{2z}^2 + 1)}{r^2 \zeta_{22} \zeta_{21}} (\xi_1 \zeta_{22} ((V_{1y} eY + V_{1z} eZ + 1) V_{2x} + eX (-V_{1y} V_{2y} - V_{1z} V_{2z} + 1)) + \zeta_{21} \xi_2 ((V_{1y} eY + V_{1z} eZ - 1) V_{2x} + eX (-V_{1y} V_{2y} - V_{1z} V_{2z} + 1))), \quad (1.0.44)$$

$$b_2 = -\frac{S(-V_{2x}^2 - V_{2y}^2 - V_{2z}^2 + 1)}{r^2 \zeta_{22} \zeta_{21}} (\xi_1 \zeta_{22} ((V_{1x} eX + V_{1z} eZ + 1) V_{2y} + eY (-V_{1x} V_{2x} - V_{1z} V_{2z} + 1)) + \zeta_{21} \xi_2 ((V_{1x} eX + V_{1z} eZ - 1) V_{2y} + eY (-V_{1x} V_{2x} - V_{1z} V_{2z} + 1))), \quad (1.0.45)$$

$$b_3 = -\frac{S(-V_{2x}^2 - V_{2y}^2 - V_{2z}^2 + 1)}{r^2 \zeta_{22} \zeta_{21}} (\xi_1 \zeta_{22} ((V_{1x} eX + V_{1y} eY + 1) V_{2z} + eZ (-V_{1x} V_{2x} - V_{1y} V_{2y} + 1)) + \zeta_{21} \xi_2 ((V_{1x} eX + V_{1y} eY - 1) V_{2z} + eZ (-V_{1x} V_{2x} - V_{1y} V_{2y} + 1))), \quad (1.0.46)$$

$$b_4 = +\frac{S(-V_{1x}^2 - V_{1y}^2 - V_{1z}^2 + 1)}{r^2 \zeta_{11} \zeta_{12}} (\xi_2 \zeta_{12} ((V_{2y} eY + V_{2z} eZ + 1) V_{1x} + eX (-V_{1y} V_{2y} - V_{1z} V_{2z} + 1)) + \xi_1 \zeta_{11} ((V_{2y} eY + V_{2z} eZ - 1) V_{1x} + eX (-V_{1y} V_{2y} - V_{1z} V_{2z} + 1))), \quad (1.0.47)$$

$$b_5 = +\frac{S(-V_{1x}^2 - V_{1y}^2 - V_{1z}^2 + 1)}{r^2 \zeta_{11} \zeta_{12}} (\xi_2 \zeta_{12} ((V_{2x} eX + V_{2z} eZ + 1) V_{1y} + eY (-V_{1x} V_{2x} - V_{1z} V_{2z} + 1)) + \xi_1 \zeta_{11} ((V_{2x} eX + V_{2z} eZ - 1) V_{1y} + eY (-V_{1x} V_{2x} - V_{1z} V_{2z} + 1))), \quad (1.0.48)$$

$$b_6 = +\frac{S(-V_{1x}^2 - V_{1y}^2 - V_{1z}^2 + 1)}{r^2 \zeta_{11} \zeta_{12}} (\xi_2 \zeta_{12} ((V_{2x} eX + V_{2y} eY + 1) V_{1z} + eZ (-V_{1x} V_{2x} - V_{1y} V_{2y} + 1)) + \xi_1 \zeta_{11} ((V_{2x} eX + V_{2y} eY - 1) V_{1z} + eZ (-V_{1x} V_{2x} - V_{1y} V_{2y} + 1))). \quad (1.0.49)$$

Sendo assim, temos o seguinte sistema matricial:

$$\begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} & m_{15} & m_{16} \\ m_{21} & m_{22} & m_{23} & m_{24} & m_{25} & m_{26} \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} \\ m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \end{pmatrix} \begin{pmatrix} A_{1x} \\ A_{1y} \\ A_{1z} \\ A_{2x} \\ A_{2y} \\ A_{2z} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{pmatrix}.$$

Portanto,  $\mathcal{M}$  é uma matriz no formato  $6 \times 6$  que contém os termos que multiplicam as acelerações. A matriz  $\mathcal{A}$ , no formato  $6 \times 1$ , é a matriz das acelerações e por fim a matriz  $\mathcal{B}$ , também no formato  $6 \times 1$ , contém os termos independentes das acelerações. A partir da equação matricial (1.0.1) realizamos a inversão da matriz  $\mathcal{M}$ , obtendo:

$$\mathcal{A} = \mathcal{M}^{-1} \mathcal{B}. \quad (1.0.50)$$

Novamente, como o interesse inicial é o regime de baixas velocidades o conjunto de equações descrita pela equação matricial (1.0.50) será expandida até a segunda ordem nas velocidades. Esta expansão até segunda ordem logo será esclarecido, sendo assim, temos:

$$P_0 = (V_{1x} = 0, V_{1y} = 0, V_{1z} = 0, V_{2x} = 0, V_{2y} = 0, V_{2z} = 0), \quad (1.0.51)$$

por simplificação na notação das derivadas parciais das velocidades iremos utilizar a seguinte regra:

$$\partial_{V_{ij}} = \frac{\partial}{\partial V_{ij}}, \quad i = 1, 2 \text{ e } j = x, y, z. \quad (1.0.52)$$

As equações das acelerações expandidas até segunda ordem nas velocidades em torno de  $P_0$  no eixo cartesiano são dados por:

$$\begin{aligned} A_{1x} = & \mathcal{A}_{1,1}(P_0) + \\ & \partial_{V_{1x}} \mathcal{A}_{1,1}(P_0) V_{1x} + \partial_{V_{1y}} \mathcal{A}_{1,1}(P_0) V_{1y} + \partial_{V_{1z}} \mathcal{A}_{1,1}(P_0) V_{1z} + \\ & \partial_{V_{2x}} \mathcal{A}_{1,1}(P_0) V_{2x} + \partial_{V_{2y}} \mathcal{A}_{1,1}(P_0) V_{2y} + \partial_{V_{2z}} \mathcal{A}_{1,1}(P_0) V_{2z} + \\ & \frac{\partial_{V_{1x}}^2 \mathcal{A}_{1,1}(P_0) V_{1x}^2}{2} + \frac{\partial_{V_{1y}}^2 \mathcal{A}_{1,1}(P_0) V_{1y}^2}{2} + \frac{\partial_{V_{1z}}^2 \mathcal{A}_{1,1}(P_0) V_{1z}^2}{2} + \\ & \frac{\partial_{V_{2x}}^2 \mathcal{A}_{1,1}(P_0) V_{2x}^2}{2} + \frac{\partial_{V_{2y}}^2 \mathcal{A}_{1,1}(P_0) V_{2y}^2}{2} + \frac{\partial_{V_{2z}}^2 \mathcal{A}_{1,1}(P_0) V_{2z}^2}{2} + \\ & \partial_{V_{1y}} \partial_{V_{1x}} \mathcal{A}_{1,1}(P_0) V_{1x} V_{1y} + \partial_{V_{1z}} \partial_{V_{1x}} \mathcal{A}_{1,1}(P_0) V_{1x} V_{1z} + \partial_{V_{1z}} \partial_{V_{1y}} \mathcal{A}_{1,1}(P_0) V_{1y} V_{1z} + \\ & \partial_{V_{2y}} \partial_{V_{2x}} \mathcal{A}_{1,1}(P_0) V_{2x} V_{2y} + \partial_{V_{2z}} \partial_{V_{2x}} \mathcal{A}_{1,1}(P_0) V_{2x} V_{2z} + \partial_{V_{2z}} \partial_{V_{2y}} \mathcal{A}_{1,1}(P_0) V_{2y} V_{2z} + \\ & \partial_{V_{2x}} \partial_{V_{1x}} \mathcal{A}_{1,1}(P_0) V_{1x} V_{2x} + \partial_{V_{2y}} \partial_{V_{1x}} \mathcal{A}_{1,1}(P_0) V_{1x} V_{2y} + \partial_{V_{2z}} \partial_{V_{1x}} \mathcal{A}_{1,1}(P_0) V_{1x} V_{2z} + \\ & \partial_{V_{2x}} \partial_{V_{1y}} \mathcal{A}_{1,1}(P_0) V_{1y} V_{2x} + \partial_{V_{2y}} \partial_{V_{1y}} \mathcal{A}_{1,1}(P_0) V_{1y} V_{2y} + \partial_{V_{2z}} \partial_{V_{1y}} \mathcal{A}_{1,1}(P_0) V_{1y} V_{2z} + \\ & \partial_{V_{2x}} \partial_{V_{1z}} \mathcal{A}_{1,1}(P_0) V_{1z} V_{2x} + \partial_{V_{2y}} \partial_{V_{1z}} \mathcal{A}_{1,1}(P_0) V_{1z} V_{2y} + \partial_{V_{2z}} \partial_{V_{1z}} \mathcal{A}_{1,1}(P_0) V_{1z} V_{2z}, \end{aligned} \quad (1.0.53)$$

$$\begin{aligned} A_{1y} = & \mathcal{A}_{2,1}(P_0) + \\ & \partial_{V_{1x}} \mathcal{A}_{2,1}(P_0) V_{1x} + \partial_{V_{1y}} \mathcal{A}_{2,1}(P_0) V_{1y} + \partial_{V_{1z}} \mathcal{A}_{2,1}(P_0) V_{1z} + \\ & \partial_{V_{2x}} \mathcal{A}_{2,1}(P_0) V_{2x} + \partial_{V_{2y}} \mathcal{A}_{2,1}(P_0) V_{2y} + \partial_{V_{2z}} \mathcal{A}_{2,1}(P_0) V_{2z} + \\ & \frac{\partial_{V_{1x}}^2 \mathcal{A}_{2,1}(P_0) V_{1x}^2}{2} + \frac{\partial_{V_{1y}}^2 \mathcal{A}_{2,1}(P_0) V_{1y}^2}{2} + \frac{\partial_{V_{1z}}^2 \mathcal{A}_{2,1}(P_0) V_{1z}^2}{2} + \\ & \frac{\partial_{V_{2x}}^2 \mathcal{A}_{2,1}(P_0) V_{2x}^2}{2} + \frac{\partial_{V_{2y}}^2 \mathcal{A}_{2,1}(P_0) V_{2y}^2}{2} + \frac{\partial_{V_{2z}}^2 \mathcal{A}_{2,1}(P_0) V_{2z}^2}{2} + \\ & \partial_{V_{1y}} \partial_{V_{1x}} \mathcal{A}_{2,1}(P_0) V_{1x} V_{1y} + \partial_{V_{1z}} \partial_{V_{1x}} \mathcal{A}_{2,1}(P_0) V_{1x} V_{1z} + \partial_{V_{1z}} \partial_{V_{1y}} \mathcal{A}_{2,1}(P_0) V_{1y} V_{1z} + \\ & \partial_{V_{2y}} \partial_{V_{2x}} \mathcal{A}_{2,1}(P_0) V_{2x} V_{2y} + \partial_{V_{2z}} \partial_{V_{2x}} \mathcal{A}_{2,1}(P_0) V_{2x} V_{2z} + \partial_{V_{2z}} \partial_{V_{2y}} \mathcal{A}_{2,1}(P_0) V_{2y} V_{2z} + \\ & \partial_{V_{2x}} \partial_{V_{1x}} \mathcal{A}_{2,1}(P_0) V_{1x} V_{2x} + \partial_{V_{2y}} \partial_{V_{1x}} \mathcal{A}_{2,1}(P_0) V_{1x} V_{2y} + \partial_{V_{2z}} \partial_{V_{1x}} \mathcal{A}_{2,1}(P_0) V_{1x} V_{2z} + \\ & \partial_{V_{2x}} \partial_{V_{1y}} \mathcal{A}_{2,1}(P_0) V_{1y} V_{2x} + \partial_{V_{2y}} \partial_{V_{1y}} \mathcal{A}_{2,1}(P_0) V_{1y} V_{2y} + \partial_{V_{2z}} \partial_{V_{1y}} \mathcal{A}_{2,1}(P_0) V_{1y} V_{2z} + \\ & \partial_{V_{2x}} \partial_{V_{1z}} \mathcal{A}_{2,1}(P_0) V_{1z} V_{2x} + \partial_{V_{2y}} \partial_{V_{1z}} \mathcal{A}_{2,1}(P_0) V_{1z} V_{2y} + \partial_{V_{2z}} \partial_{V_{1z}} \mathcal{A}_{2,1}(P_0) V_{1z} V_{2z}, \end{aligned} \quad (1.0.54)$$





$$\begin{aligned}
 A_{2z} = & \mathcal{A}_{6,1}(P_0) + \\
 & \partial_{V_{1x}} \mathcal{A}_{6,1}(P_0) V_{1x} + \partial_{V_{1y}} \mathcal{A}_{6,1}(P_0) V_{1y} + \partial_{V_{1z}} \mathcal{A}_{6,1}(P_0) V_{1z} + \\
 & \partial_{V_{2x}} \mathcal{A}_{6,1}(P_0) V_{2x} + \partial_{V_{2y}} \mathcal{A}_{6,1}(P_0) V_{2y} + \partial_{V_{2z}} \mathcal{A}_{6,1}(P_0) V_{2z} + \\
 & \frac{\partial_{V_{1x}}^2 \mathcal{A}_{6,1}(P_0) V_{1x}^2}{2} + \frac{\partial_{V_{1y}}^2 \mathcal{A}_{6,1}(P_0) V_{1y}^2}{2} + \frac{\partial_{V_{1z}}^2 \mathcal{A}_{6,1}(P_0) V_{1z}^2}{2} + \\
 & \frac{\partial_{V_{2x}}^2 \mathcal{A}_{6,1}(P_0) V_{2x}^2}{2} + \frac{\partial_{V_{2y}}^2 \mathcal{A}_{6,1}(P_0) V_{2y}^2}{2} + \frac{\partial_{V_{2z}}^2 \mathcal{A}_{6,1}(P_0) V_{2z}^2}{2} + \\
 & \partial_{V_{1y}} \partial_{V_{1x}} \mathcal{A}_{6,1}(P_0) V_{1x} V_{1y} + \partial_{V_{1z}} \partial_{V_{1x}} \mathcal{A}_{6,1}(P_0) V_{1x} V_{1z} + \partial_{V_{1z}} \partial_{V_{1y}} \mathcal{A}_{6,1}(P_0) V_{1y} V_{1z} + \\
 & \partial_{V_{2y}} \partial_{V_{2x}} \mathcal{A}_{6,1}(P_0) V_{2x} V_{2y} + \partial_{V_{2z}} \partial_{V_{2x}} \mathcal{A}_{6,1}(P_0) V_{2x} V_{2z} + \partial_{V_{2z}} \partial_{V_{2y}} \mathcal{A}_{6,1}(P_0) V_{2y} V_{2z} + \\
 & \partial_{V_{2x}} \partial_{V_{1x}} \mathcal{A}_{6,1}(P_0) V_{1x} V_{2x} + \partial_{V_{2y}} \partial_{V_{1x}} \mathcal{A}_{6,1}(P_0) V_{1x} V_{2y} + \partial_{V_{2z}} \partial_{V_{1x}} \mathcal{A}_{6,1}(P_0) V_{1x} V_{2z} + \\
 & \partial_{V_{2x}} \partial_{V_{1y}} \mathcal{A}_{6,1}(P_0) V_{1y} V_{2x} + \partial_{V_{2y}} \partial_{V_{1y}} \mathcal{A}_{6,1}(P_0) V_{1y} V_{2y} + \partial_{V_{2z}} \partial_{V_{1y}} \mathcal{A}_{6,1}(P_0) V_{1y} V_{2z} + \\
 & \partial_{V_{2x}} \partial_{V_{1z}} \mathcal{A}_{6,1}(P_0) V_{1z} V_{2x} + \partial_{V_{2y}} \partial_{V_{1z}} \mathcal{A}_{6,1}(P_0) V_{1z} V_{2y} + \partial_{V_{2z}} \partial_{V_{1z}} \mathcal{A}_{6,1}(P_0) V_{1z} V_{2z}.
 \end{aligned} \tag{1.0.58}$$

A motivação para expansão das acelerações até a segunda ordem nas velocidades resulta do parâmetro  $\alpha$ . Para compreender tal afirmação utilizaremos como exemplo a aceleração da partícula 1 no eixo  $x$ . Como  $A_{1x}$  depende dos parâmetros  $S$ ,  $\alpha$ ,  $\eta$  e das variáveis  $\mathbf{r}$  e  $\mathbf{v}$ , isto é:

$$A_{1x} = A_{1x}(S, \alpha, \eta, \mathbf{r}, \mathbf{v}), \tag{1.0.59}$$

onde:

$$\mathbf{r} = (x_1, y_1, z_1, x_2, y_2, z_2), \tag{1.0.60}$$

$$\mathbf{v} = (V_{1x}, V_{1y}, V_{1z}, V_{2x}, V_{2y}, V_{2z}), \tag{1.0.61}$$

a expansão de  $A_{1x}$  pode ser escrita da seguinte forma:

$$\begin{aligned}
 A_{1x} = & A_{1x}(P_0) \\
 & + \sum_{i=1}^6 \partial_i A_{1x}(P_0) V_i \\
 & + \frac{1}{2} \sum_{i,j=1}^6 \partial_{ij} A_{1x}(P_0) V_i V_j \\
 & + \frac{1}{3!} \sum_{i,j,k=1}^6 \partial_{ijk} A_{1x}(P_0) V_i V_j V_k + \dots
 \end{aligned} \tag{1.0.62}$$

A expansão (1.0.62) em termos do parâmetro  $\alpha$  é dada por:

$$\begin{aligned}
 A_{1x} = & \mathfrak{F}^1(S, \eta, \mathbf{r}) \\
 & + \alpha \sum_{i=1}^6 \mathfrak{F}_i^2(S, \eta, \mathbf{r}) V_i \\
 & + \frac{1}{2} \sum_{i,j=1}^6 (\mathfrak{F}_{ij}^3(S, \eta, \mathbf{r}) + \alpha \mathfrak{F}_{ij}^4(S, \eta, \mathbf{r})) V_i V_j \\
 & + \alpha \frac{1}{3!} \sum_{i,j,k=1}^6 \mathfrak{F}_{ijk}^5(S, \eta, \alpha, \mathbf{r}) V_i V_j V_k + \dots
 \end{aligned} \tag{1.0.63}$$

No caso  $\alpha = 0$ , se considerarmos a expansão até a primeira ordem as velocidades seriam nulas, portanto expandimos até a segunda ordem nas velocidades para que possamos analisar a evolução temporal das partículas sendo dependentes também das velocidades, independentemente do valor do parâmetro  $\alpha$ . Esta justificativa aplica-se para as outras coordenadas da partícula 1, assim como para a partícula 2.

Realizando a inversão do sistema sistema matricial acima e aplicando a expansão até a segunda ordem nas velocidades obtemos as seguintes equações:

$$\begin{aligned}
 A_{1x} = & -\frac{eX S}{\eta r^2} \\
 & -2 \frac{(-\eta r + S) \alpha (eX^2 - 1) V_{1x}}{r^2 \eta (-\eta r^2 + 1)} \\
 & -2 \frac{eX eY \alpha (-\eta r + S) V_{1y}}{\eta r^2 (-\eta r^2 + 1)} \\
 & -2 \frac{eX eZ \alpha (-\eta r + S) V_{1z}}{\eta r^2 (-\eta r^2 + 1)} \\
 & +4 \frac{S (eX^2 + 1/2 r \eta S (eX^2 - 1) - 3/2 (eX^2 - 1/3) r^2 \eta) \alpha V_{2x}}{\eta r^2 (-\eta r^2 + 1)} \\
 & -2 \frac{eY eX S (2 + r (S - 3r) \eta) \alpha V_{2y}}{\eta r^2 (\eta r^2 - 1)} \\
 & -2 \frac{eX eZ S (2 + r (S - 3r) \eta) \alpha V_{2z}}{\eta r^2 (\eta r^2 - 1)} \\
 & -3 \frac{eX (-\eta r + S) (\eta r^2 + 4\alpha^2 - 1) (-eX^2 + 1) V_{1x}^2}{(\eta r^2 - 1)^2 r^2 \eta} \\
 & +3 \frac{eX (-\eta r + S) (r^2 (eY^2 + 4/3 \alpha^2) \eta + (4eY^2 - 4/3) \alpha^2 - eY^2) V_{1y}^2}{\eta r^2 (\eta r^2 - 1)^2} \\
 & +3 \frac{eX (r^2 (eZ^2 + 4/3 \alpha^2) \eta + (4eZ^2 - 4/3) \alpha^2 - eZ^2) (-\eta r + S) V_{1z}^2}{\eta r^2 (\eta r^2 - 1)^2} \\
 & -6 \frac{eY (r^2 (2/3 \alpha^2 + 1/2 - eX^2) \eta + 4 (1/3 - eX^2) \alpha^2 + eX^2 - 1/2) (-\eta r + S) V_{1x} V_{1y}}{\eta r^2 (\eta r^2 - 1)^2} \\
 & -6 \frac{eZ (r^2 (2/3 \alpha^2 + 1/2 - eX^2) \eta + 4 (1/3 - eX^2) \alpha^2 + eX^2 - 1/2) (-\eta r + S) V_{1x} V_{1z}}{\eta r^2 (\eta r^2 - 1)^2} \\
 & +6 \frac{(-\eta r + S) (\eta r^2 + 4\alpha^2 - 1) eY eZ eX V_{1z} V_{1y}}{(\eta r^2 - 1)^2 r^2 \eta} \\
 & -3 \frac{eX S V_{2x}^2}{(\eta r^2 - 1)^2 r^2 \eta} (((2eX^2 - 4/3) r^4 + S r^3 (1 - eX^2)) \eta^2 + eX^2 - 1/3 \\
 & + 4r (((3/4 - \alpha^2) (1 - eX^2) - 1/3) r + S (1 - eX^2) (\alpha^2 - 1/4)) \eta) \\
 & +3 \frac{eX S V_{2y}^2}{(\eta r^2 - 1)^2 r^2 \eta} (((1/3 - 2eY^2) r^4 + S r^3 eY^2) \eta^2 - eY^2 + 1/3 \\
 & + 4r (((3/4 - \alpha^2) eY^2 - 1/6) r + S eY^2 (\alpha^2 - 1/4)) \eta)
 \end{aligned}$$

$$\begin{aligned}
 & +3 \frac{eXSV_{2z}^2}{(\eta r^2 - 1)^2 r^2 \eta} \left( ((1/3 - 2eZ^2) r^4 + Sr^3 eZ^2) \eta^2 - eZ^2 + 1/3 \right. \\
 & +4 \left( ((3/4 - \alpha^2) eZ^2 - 1/6) r + SeZ^2 (\alpha^2 - 1/4) \right) r\eta \\
 & -6 \frac{eYSV_{2y}V_{2x}}{(\eta r^2 - 1)^2 r^2 \eta} \left( r^3 (eX^2 r + (1/2 - eX^2) (S - r)) \eta^2 + eX^2 \right. \\
 & +4r \left( ((\alpha^2 - 1/4) (S - r) + r/2) (1/2 - eX^2) - r/4 \right) \eta \\
 & -6 \frac{eZSV_{2z}V_{2x}}{(\eta r^2 - 1)^2 r^2 \eta} \left( r^3 (eX^2 r + (1/2 - eX^2) (S - r)) \eta^2 + eX^2 \right. \\
 & +4r \left( ((\alpha^2 - 1/4) (S - r) + r/2) (1/2 - eX^2) - r/4 \right) \eta \\
 & +6 \frac{eYeZeXS (r^3 (S - 2r) \eta^2 + ((4\alpha^2 - 1) (S - r) r + 2r^2) \eta - 1) V_{2y}V_{2z}}{(\eta r^2 - 1)^2 r^2 \eta} \\
 & + \frac{(4\alpha^2 (2Sr - 1) - \eta r^2 + 1) (1 - eX^2) eX V_{1x}V_{2x}}{r^3 (\eta r^2 - 1) \eta} \\
 & +8 \frac{eYV_{2y}V_{1x}}{r^3 \eta (\eta r^2 - 1)^2} \left( S (1/8 + (2 - eX^2) \alpha^2) \eta r^3 - 1/8 \eta^2 (8\alpha^2 - eX^2 + 1) r^4 \right. \\
 & -1/2 \eta \left( (1 - eX^2) (\alpha^2 - 1/2) + 1/4 \right) r^2 - (1/8 + (1/2 - eX^2) \alpha^2) Sr - 1/2 eX^2 (\alpha^2 - 1/4) \Big) \\
 & +8 \frac{eZV_{2z}V_{1x}}{r^3 \eta (\eta r^2 - 1)^2} \left( S (1/8 + (2 - eX^2) \alpha^2) \eta r^3 - 1/8 \eta^2 (8\alpha^2 - eX^2 + 1) r^4 \right. \\
 & -1/2 \eta \left( (1 - eX^2) (\alpha^2 - 1/2) + 1/4 \right) r^2 - (1/8 + (1/2 - eX^2) \alpha^2) Sr - 1/2 eX^2 (\alpha^2 - 1/4) \Big) \\
 & - \frac{eYSV_{2x}V_{1y}}{(\eta r^2 - 1)^2 r^3 \eta} \left( \eta^2 r^5 - S\eta^2 (8\alpha^2 + eX^2 + 1) r^4 - 8\eta (1/8 - (3/2 + eX^2) \alpha^2) r^3 \right. \\
 & -4S (eX^2 (\alpha^2 - 1/2) + \alpha^2 - 1/4) \eta r^2 + 4eX^2 (-2\alpha^2 r + S (\alpha^2 - 1/4)) \Big) \\
 & + \frac{eXS (\eta r^3 + (-1 + 4 (-2eY^2 + 1) \alpha^2) r + (\eta r^2 + 4\alpha^2 - 1) (eY^2 - 1) S) V_{1y}V_{2y}}{\eta r^3 (\eta r^2 - 1)} \\
 & - \frac{eZSV_{1z}V_{2x}}{(\eta r^2 - 1)^2 r^3 \eta} \left( \eta^2 r^5 - S\eta^2 (8\alpha^2 + eX^2 + 1) r^4 - 8\eta (1/8 + (-eX^2 - 3/2) \alpha^2) r^3 \right. \\
 & +4 \left( -(eX^2 + 1) \alpha^2 + 1/2 eX^2 + 1/4 \right) S\eta r^2 + eX^2 (4S (\alpha^2 - 1/4) - 8\alpha^2 r) \Big) \\
 & - \frac{(4\alpha^2 (2Sr - 1) - \eta r^2 + 1) eYeXeZeZ V_{1z}V_{2y}}{r^3 \eta (\eta r^2 - 1)} \\
 & + \frac{eXS (\eta r^3 + (4 (1 - 2eZ^2) \alpha^2 - 1) r + (\eta r^2 + 4\alpha^2 - 1) S (eZ^2 - 1)) V_{1z}V_{2z}}{\eta r^3 (\eta r^2 - 1)},
 \end{aligned}$$

(1.0.64)

$$\begin{aligned}
 A_{1y} = & -\frac{eY S}{\eta r^2} \\
 & +2 \frac{(S - \eta r) \alpha eX eY V_{1x}}{\eta r^2 (\eta r^2 - 1)} \\
 & -2 \frac{(eY^2 - 1) (S - \eta r) \alpha V_{1y}}{r^2 \eta (-\eta r^2 + 1)} \\
 & -2 \frac{eY eZ (S - \eta r) \alpha V_{1z}}{\eta r^2 (-\eta r^2 + 1)} \\
 & -2 \frac{eX eY S \alpha (2 + r (S - 3r) \eta) V_{2x}}{\eta r^2 (\eta r^2 - 1)} \\
 & -2 \frac{\alpha S (((1 - 3eY^2) r + S (eY^2 - 1)) r \eta + 2eY^2) V_{2y}}{r^2 \eta (\eta r^2 - 1)} \\
 & -2 \frac{eY S \alpha eZ (2 + r (S - 3r) \eta) V_{2z}}{\eta r^2 (\eta r^2 - 1)} \\
 & -3 \frac{eY (4\alpha^2 (1/3 - eX^2) - \eta r^2 (4/3 \alpha^2 + eX^2) + eX^2) (S - \eta r) V_{1x}^2}{\eta r^2 (\eta r^2 - 1)^2} \\
 & +3 \frac{eY (S - \eta r) (\eta r^2 + 4\alpha^2 - 1) (eY^2 - 1) V_{1y}^2}{(\eta r^2 - 1)^2 r^2 \eta} \\
 & +3 \frac{eY (r^2 (eZ^2 + 4/3 \alpha^2) \eta + (4eZ^2 - 4/3) \alpha^2 - eZ^2) (S - \eta r) V_{1z}^2}{\eta r^2 (\eta r^2 - 1)^2} \\
 & +6 \frac{eX (r^2 (eY^2 - 1/2 - 2/3 \alpha^2) \eta + (4eY^2 - 4/3) \alpha^2 - eY^2 + 1/2) (S - \eta r) V_{1x} V_{1y}}{\eta r^2 (\eta r^2 - 1)^2} \\
 & +6 \frac{(\eta r^2 + 4\alpha^2 - 1) (S - \eta r) eY eX eZ V_{1z} V_{1x}}{r^2 \eta (\eta r^2 - 1)^2} \\
 & +6 \frac{eZ (r^2 (eY^2 - 1/2 - 2/3 \alpha^2) \eta + (4eY^2 - 4/3) \alpha^2 - eY^2 + 1/2) (S - \eta r) V_{1y} V_{1z}}{\eta r^2 (\eta r^2 - 1)^2} \\
 & -3 \frac{eY S V_{2x}^2}{r^2 \eta (\eta r^2 - 1)^2} (((2eX^2 - 1/3) r - eX^2 S) r^3 \eta^2 + eX^2 - 1/3 \\
 & +4 ((1/6 - (3/4 - \alpha^2) eX^2) r - eX^2 S (\alpha^2 - 1/4)) r \eta) \\
 & +3 \frac{eY S V_{2y}^2}{r^2 \eta (\eta r^2 - 1)^2} (r^3 ((4/3 - 2eY^2) r + S (eY^2 - 1)) \eta^2 - eY^2 + 1/3 \\
 & +4 ((eY^2 - 1) (r/2 + (\alpha^2 - 1/4) (S - r)) + r/3) r \eta) \\
 & +3 \frac{eY S V_{2z}^2}{(\eta r^2 - 1)^2 r^2 \eta} (((1/3 - 2eZ^2) r^4 + S r^3 eZ^2) \eta^2 - eZ^2 + 1/3 \\
 & +4 r (((3/4 - \alpha^2) eZ^2 - 1/6) r + S eZ^2 (-1/4 + \alpha^2)) \eta) \\
 & +6 \frac{eX S V_{2x} V_{2y}}{(\eta r^2 - 1)^2 r^2 \eta} (((S - r) (eY^2 - 1/2) - r eY^2) r^3 \eta^2 - eY^2 \\
 & +4 \eta r ((eY^2 - 1/2) (r/2 + (-1/4 + \alpha^2) (S - r)) + r/4)) \\
 & +6 \frac{eX eZ eY S (r^3 (S - 2r) \eta^2 - 1 + (2r^2 + (S - r) r (4\alpha^2 - 1)) \eta) V_{2x} V_{2z}}{(\eta r^2 - 1)^2 r^2 \eta}
 \end{aligned}$$

$$\begin{aligned}
 & +6 \frac{eZ S V_{2y} V_{2z}}{(\eta r^2 - 1)^2 r^2 \eta} \left( ((-2eY^2 + 1/2)r + S(eY^2 - 1/2)) r^3 \eta^2 - eY^2 \right. \\
 & + 4r(1/2 r eY^2 + (-1/4 + \alpha^2)(eY^2 - 1/2)(S - r)) \eta \\
 & \left. + \frac{eY S (r^3 \eta + (4(1 - 2eX^2)\alpha^2 - 1)r + (4S(\alpha^2 - 1/4) + r^2 \eta S)(eX^2 - 1)) V_{1x} V_{2x}}{(\eta r^2 - 1) \eta r^3} \right. \\
 & + \frac{eX S V_{1x} V_{2y}}{(\eta r^2 - 1)^2 r^3 \eta} (S \eta^2 (8\alpha^2 + eY^2 + 1)r^4 - \eta^2 r^5 + (1 + (-8eY^2 - 12)\alpha^2) \eta r^3 \\
 & + 4((eY^2 + 1)\alpha^2 - 1/2 eY^2 - 1/4) S \eta r^2 + 8\alpha^2 eY^2 r + (-4\alpha^2 + 1)eY^2 S) \\
 & \left. - \frac{eZ eX eY (4\alpha^2 (2Sr - 1) - \eta r^2 + 1) V_{1x} V_{2z}}{(\eta r^2 - 1) r^3 \eta} \right. \\
 & - 8 \frac{eX V_{1y} V_{2x}}{(\eta r^2 - 1)^2 \eta r^3} \left( ((Sr - 1/2)\alpha^2 - 1/8 \eta r^2 + 1/8) (\eta r^2 - 1) eY^2 \right. \\
 & - 2r((\eta r^2 (S - 1/2 \eta r) - 1/4 \eta r - S/4)\alpha^2 + 1/16 (\eta r^2 - 1)(S - \eta r)) \\
 & \left. - \frac{(4\alpha^2 (2Sr - 1) - \eta r^2 + 1) eY (eY^2 - 1) V_{1y} V_{2y}}{r^3 (\eta r^2 - 1) \eta} \right. \\
 & - 8 \frac{eZ V_{1y} V_{2z}}{(\eta r^2 - 1)^2 \eta r^3} \left( ((Sr - 1/2)\alpha^2 - 1/8 \eta r^2 + 1/8) (\eta r^2 - 1) eY^2 \right. \\
 & - 2r((\eta r^2 (S - 1/2 \eta r) - 1/4 \eta r - S/4)\alpha^2 + 1/16 (\eta r^2 - 1)(S - \eta r)) \\
 & \left. - \frac{eX eZ eY (4\alpha^2 (2Sr - 1) - \eta r^2 + 1) V_{1z} V_{2x}}{(\eta r^2 - 1) r^3 \eta} \right. \\
 & + \frac{eZ S V_{1z} V_{2y}}{r^3 (\eta r^2 - 1)^2 \eta} \left( ((\eta r^2 - 1)(\eta r^2 + 4\alpha^2 - 1) eY^2 + (8\eta^2 r^4 + 4\eta r^2)\alpha^2 + \eta^2 r^4 - \eta r^2) S \right. \\
 & - (8\alpha^2 (\eta r^2 - 1) eY^2 + \eta r^2 (\eta r^2 + 12\alpha^2 - 1)) r \\
 & \left. + \frac{eY S (r^3 \eta + S \eta (eZ^2 - 1) r^2 + ((4 - 8eZ^2)\alpha^2 - 1)r + (eZ^2 - 1)(4\alpha^2 - 1)S) V_{1z} V_{2z}}{\eta r^3 (\eta r^2 - 1)} \right),
 \end{aligned} \tag{1.0.65}$$

$$\begin{aligned}
 A_{1z} = & -\frac{eZ S}{\eta r^2} \\
 & +2 \frac{\alpha (S - \eta r) eX eZ V_{1x}}{\eta r^2 (\eta r^2 - 1)} \\
 & -2 \frac{eY eZ \alpha (S - \eta r) V_{1y}}{\eta r^2 (-\eta r^2 + 1)} \\
 & +2 \frac{(1 - eZ^2) (S - \eta r) \alpha V_{1z}}{\eta r^2 (-\eta r^2 + 1)} \\
 & +4 \frac{\alpha eZ eX S (1 + 1/2 r \eta S - 3/2 \eta r^2) V_{2x}}{\eta r^2 (-\eta r^2 + 1)} \\
 & -2 \frac{\alpha eY eZ S (2 + r (S - 3r) \eta) V_{2y}}{\eta r^2 (\eta r^2 - 1)} \\
 & +2 \frac{(r ((3eZ^2 - 1)r + S(1 - eZ^2)) \eta - 2eZ^2) S \alpha V_{2z}}{\eta r^2 (\eta r^2 - 1)} \\
 & -3 \frac{eZ (4\alpha^2 (1/3 - eX^2) - \eta r^2 (4/3 \alpha^2 + eX^2) + eX^2) (S - \eta r) V_{1x}^2}{\eta r^2 (\eta r^2 - 1)^2} \\
 & +3 \frac{eZ (S - \eta r) (r^2 (eY^2 + 4/3 \alpha^2) \eta + (4eY^2 - 4/3) \alpha^2 - eY^2) V_{1y}^2}{\eta r^2 (\eta r^2 - 1)^2} \\
 & +3 \frac{eZ (eZ^2 - 1) (S - \eta r) (\eta r^2 + 4\alpha^2 - 1) V_{1z}^2}{(\eta r^2 - 1)^2 r^2 \eta} \\
 & +6 \frac{(S - \eta r) (\eta r^2 + 4\alpha^2 - 1) eY eZ eX V_{1x} V_{1y}}{(\eta r^2 - 1)^2 r^2 \eta} \\
 & +6 \frac{eX (r^2 (eZ^2 - 2/3 \alpha^2 - 1/2) \eta + (4eZ^2 - 4/3) \alpha^2 - eZ^2 + 1/2) (S - \eta r) V_{1x} V_{1z}}{\eta r^2 (\eta r^2 - 1)^2} \\
 & +6 \frac{eY (r^2 (eZ^2 - 2/3 \alpha^2 - 1/2) \eta + (4eZ^2 - 4/3) \alpha^2 - eZ^2 + 1/2) (S - \eta r) V_{1y} V_{1z}}{\eta r^2 (\eta r^2 - 1)^2} \\
 & -3 \frac{eZ S V_{2x}^2}{(\eta r^2 - 1)^2 r^2 \eta} (((2eX^2 - 1/3)r - eX^2 S) r^3 \eta^2 + eX^2 - 1/3 \\
 & +4r (1/2 r (1/3 - eX^2) + (\alpha^2 - 1/4) eX^2 (r - S)) \eta) \\
 & +3 \frac{eZ S V_{2y}^2}{(\eta r^2 - 1)^2 r^2 \eta} (((1/3 - 2eY^2) r^4 + r^3 eY^2 S) \eta^2 - eY^2 + 1/3 \\
 & +4r (((3/4 - \alpha^2) eY^2 - 1/6) r + eY^2 S (\alpha^2 - 1/4)) \eta) \\
 & +3 \frac{eZ S V_{2z}^2}{(\eta r^2 - 1)^2 r^2 \eta} (r^3 ((4/3 - 2eZ^2) r + S (eZ^2 - 1)) \eta^2 - eZ^2 + 1/3 \\
 & +4r (r/3 + (eZ^2 - 1) (r/2 + (\alpha^2 - 1/4) (S - r))) \eta) \\
 & +6 \frac{eZ eX eY S (r^3 (S - 2r) \eta^2 - 1 + ((3 - 4\alpha^2) r^2 + S (4\alpha^2 - 1) r) \eta) V_{2x} V_{2y}}{(\eta r^2 - 1)^2 r^2 \eta} \\
 & +6 \frac{eX S V_{2x} V_{2z}}{(\eta r^2 - 1)^2 r^2 \eta} (r^3 ((S - r) (eZ^2 - 1/2) - eZ^2 r) \eta^2 - eZ^2 \\
 & +4r (r/4 + (eZ^2 - 1/2) (S (\alpha^2 - 1/4) + (3/4 - \alpha^2) r)) \eta)
 \end{aligned}$$

$$\begin{aligned}
 & +6 \frac{eY S V_{2y} V_{2z}}{(\eta r^2 - 1)^2 r^2 \eta} (r^3 ((S - r) (eZ^2 - 1/2) - eZ^2 r) \eta^2 - eZ^2 \\
 & +4r (r/4 + (eZ^2 - 1/2) (S (\alpha^2 - 1/4) + (3/4 - \alpha^2) r)) \eta) \\
 & + \frac{(r^3 \eta + (4 (1 - 2eX^2) \alpha^2 - 1) r + S (1 - \eta r^2 - 4\alpha^2) (1 - eX^2)) eZ S V_{1x} V_{2x}}{r^3 (\eta r^2 - 1) \eta} \\
 & - \frac{(4\alpha^2 (2Sr - 1) - \eta r^2 + 1) eY eZ eX V_{1x} V_{2y}}{r^3 (\eta r^2 - 1) \eta} \\
 & + \frac{eX S V_{1x} V_{2z}}{r^3 \eta (\eta r^2 - 1)^2} (S \eta^2 r^4 (8\alpha^2 + eZ^2 + 1) - \eta^2 r^5 + (1 - (8eZ^2 + 12)\alpha^2) \eta r^3 \\
 & +4\eta ((eZ^2 + 1)\alpha^2 - 1/2 eZ^2 - 1/4) S r^2 + eZ^2 (8r\alpha^2 + S (1 - 4\alpha^2))) \\
 & - \frac{(4\alpha^2 (2Sr - 1) - \eta r^2 + 1) eY eZ eX V_{1y} V_{2x}}{(\eta r^2 - 1) r^3 \eta} \\
 & + \frac{(r^3 \eta + r^2 \eta (eY^2 - 1) S + ((4 - 8eY^2) \alpha^2 - 1) r + (eY^2 - 1) (4\alpha^2 - 1) S) eZ S V_{1y} V_{2y}}{(\eta r^2 - 1) r^3 \eta} \\
 & + \frac{eY S V_{1y} V_{2z}}{r^3 \eta (\eta r^2 - 1)^2} (S \eta^2 r^4 (8\alpha^2 + eZ^2 + 1) - \eta^2 r^5 + (1 - (8eZ^2 + 12)\alpha^2) \eta r^3 \\
 & +4\eta ((eZ^2 + 1)\alpha^2 - 1/2 eZ^2 - 1/4) S r^2 + 8r\alpha^2 eZ^2 + (1 - 4\alpha^2) eZ^2 S) \\
 & -8 \frac{eX V_{1z} V_{2x}}{r^3 \eta (\eta r^2 - 1)^2} (-1/8 \eta^2 (-8\alpha^2 + eZ^2 - 1) r^4 + \eta (-1/8 + (eZ^2 - 2)\alpha^2) S r^3 - 1/8 eZ^2 \\
 & -1/2 \eta ((eZ^2 - 1)\alpha^2 - 1/2 eZ^2 + 1/4) r^2 - S ((eZ^2 - 1/2)\alpha^2 - 1/8) r + 1/2 \alpha^2 eZ^2) \\
 & -8 \frac{eY V_{1z} V_{2y}}{r^3 \eta (\eta r^2 - 1)^2} (-1/8 \eta^2 (-8\alpha^2 + eZ^2 - 1) r^4 + \eta (-1/8 + (eZ^2 - 2)\alpha^2) S r^3 - 1/8 eZ^2 \\
 & -1/2 \eta ((eZ^2 - 1)\alpha^2 - 1/2 eZ^2 + 1/4) r^2 - S ((eZ^2 - 1/2)\alpha^2 - 1/8) r + 1/2 \alpha^2 eZ^2) \\
 & - \frac{(4\alpha^2 (2Sr - 1) - \eta r^2 + 1) eZ (eZ^2 - 1) V_{1z} V_{2z}}{(\eta r^2 - 1) r^3 \eta},
 \end{aligned}$$

(1.0.66)



$$\begin{aligned}
 A_{2x} = & + \frac{eX S}{r^2} \\
 & + 4 \frac{\alpha S (1/2 r (eX^2 - 1) S - 3/2 \eta r^2 (eX^2 - 1/3) + eX^2) V_{1x}}{r^2 (-\eta r^2 + 1)} \\
 & + 4 \frac{eY eX \alpha S (1 + 1/2 r S - 3/2 \eta r^2) V_{1y}}{r^2 (-\eta r^2 + 1)} \\
 & + 4 \frac{eZ eX \alpha S (1 + 1/2 r S - 3/2 \eta r^2) V_{1z}}{r^2 (-\eta r^2 + 1)} \\
 & - 2 \frac{(eX^2 - 1) (S - r) \alpha V_{2x}}{r^2 (-\eta r^2 + 1)} \\
 & + 2 \frac{eY eX \alpha (S - r) V_{2y}}{r^2 (\eta r^2 - 1)} \\
 & + 2 \frac{eZ eX \alpha (S - r) V_{2z}}{r^2 (\eta r^2 - 1)} \\
 & + 3 \frac{eX S V_{1x}^2}{r^2 (\eta r^2 - 1)^2} (S \eta (1 - eX^2) r^3 - 2 (2/3 - eX^2) \eta^2 r^4 + eX^2 - 1/3 \\
 & - 4 ((\alpha^2 - 3/4) (1 - eX^2) + 1/3) \eta r^2 + 4 (\alpha^2 - 1/4) S (1 - eX^2) r) \\
 & - 3 \frac{eX S V_{1y}^2}{r^2 (\eta r^2 - 1)^2} (S r^3 eY^2 \eta - 2 r^4 (eY^2 - 1/6) \eta^2 - eY^2 + 1/3 \\
 & + ((3 - 4\alpha^2) eY^2 - 2/3) \eta r^2 + eY^2 S (4\alpha^2 - 1) r) \\
 & - 3 \frac{eX S V_{1z}^2}{r^2 (\eta r^2 - 1)^2} (S r^3 \eta eZ^2 - 2 (eZ^2 - 1/6) \eta^2 r^4 - eZ^2 + 1/3 \\
 & + ((3 - 4\alpha^2) eZ^2 - 2/3) \eta r^2 + S eZ^2 (4\alpha^2 - 1) r) \\
 & + 6 \frac{eY S V_{1x} V_{1y}}{r^2 (\eta r^2 - 1)^2} (S \eta (1/2 - eX^2) r^3 - 2 \eta^2 (1/4 - eX^2) r^4 + eX^2 \\
 & - 4 ((1/2 - eX^2) (\alpha^2 - 3/4) + 1/4) \eta r^2 + 4 S (\alpha^2 - 1/4) (1/2 - eX^2) r) \\
 & + 6 \frac{eZ S V_{1x} V_{1z}}{r^2 (\eta r^2 - 1)^2} (S \eta (1/2 - eX^2) r^3 - 2 \eta^2 (1/4 - eX^2) r^4 + eX^2 \\
 & - 4 ((1/2 - eX^2) (\alpha^2 - 3/4) + 1/4) \eta r^2 + 4 S (\alpha^2 - 1/4) (1/2 - eX^2) r) \\
 & - 6 \frac{eX eZ eY S (-1 - 2 r^4 \eta^2 + r^3 \eta S + (-4 \alpha^2 + 3) \eta r^2 + S (4 \alpha^2 - 1) r) V_{1y} V_{1z}}{r^2 (\eta r^2 - 1)^2} \\
 & + 3 \frac{eX (S - r) (\eta r^2 + 4 \alpha^2 - 1) (-eX^2 + 1) V_{2x}^2}{r^2 (\eta r^2 - 1)^2} \\
 & - 3 \frac{(S - r) eX (r^2 (eY^2 + 4/3 \alpha^2) \eta + (4 eY^2 - 4/3) \alpha^2 - eY^2) V_{2y}^2}{r^2 (\eta r^2 - 1)^2} \\
 & - 3 \frac{(S - r) eX (r^2 (eZ^2 + 4/3 \alpha^2) \eta + (4 eZ^2 - 4/3) \alpha^2 - eZ^2) V_{2z}^2}{r^2 (\eta r^2 - 1)^2} \\
 & + 6 \frac{eY (S - r) (\eta (2/3 \alpha^2 + 1/2 - eX^2) r^2 + (1 - eX^2) (4 \alpha^2 - 1) - 8/3 \alpha^2 + 1/2) V_{2x} V_{2y}}{r^2 (\eta r^2 - 1)^2}
 \end{aligned}$$

$$\begin{aligned}
 & +6 \frac{eZ (S - r) (\eta (2/3 \alpha^2 + 1/2 - eX^2) r^2 + (1 - eX^2) (4 \alpha^2 - 1) - 8/3 \alpha^2 + 1/2) V_{2x} V_{2z}}{r^2 (\eta r^2 - 1)^2} \\
 & -6 \frac{(\eta r^2 + 4 \alpha^2 - 1) (S - r) eY eX eZ V_{2y} V_{2z}}{r^2 (\eta r^2 - 1)^2} \\
 & - \frac{eX (8r\eta \alpha^2 S - \eta r^2 - 4 \alpha^2 + 1) (1 - eX^2) V_{1x} V_{2x}}{\eta (\eta r^2 - 1) r^3} \\
 & + \frac{eY S V_{1x} V_{2y}}{r^3 (\eta r^2 - 1)^2 \eta} (\eta^3 r^5 + \eta^2 r^3 ((8eX^2 + 12) \alpha^2 - rS (8\alpha^2 + eX^2 + 1) - 1)) \\
 & + 4 (((1/2 - \alpha^2) (eX^2 + 1) - 1/4) Sr - 2\alpha^2 eX^2) r\eta + 4eX^2 S (\alpha^2 - 1/4)) \\
 & + \frac{eZ S V_{1x} V_{2z}}{r^3 (\eta r^2 - 1)^2 \eta} (\eta^3 r^5 + \eta^2 r^3 ((8eX^2 + 12) \alpha^2 - rS (8\alpha^2 + eX^2 + 1) - 1)) \\
 & + 4 (((1/2 - \alpha^2) (eX^2 + 1) - 1/4) Sr - 2\alpha^2 eX^2) r\eta + 4eX^2 S (\alpha^2 - 1/4)) \\
 & - 8 \frac{eY V_{1y} V_{2x}}{r^3 (\eta r^2 - 1)^2 \eta} (((1/8 eX^2 - 1/8 - \alpha^2) r + S (1/8 + (2 - eX^2) \alpha^2)) r^3 \eta^2 \\
 & - r ((1/2 (1 - eX^2) (\alpha^2 - 1/2) + 1/8) r + S (1/8 + (1/2 - eX^2) \alpha^2)) \eta - 1/2 eX^2 (\alpha^2 - 1/4)) \\
 & - \frac{eX S (\eta^2 r^3 + (S (eY^2 - 1) r - 1 + (-8eY^2 + 4) \alpha^2) r\eta + (eY^2 - 1) (4\alpha^2 - 1) S) V_{1y} V_{2y}}{r^3 \eta (\eta r^2 - 1)} \\
 & + \frac{(4\alpha^2 (2Sr\eta - 1) - \eta r^2 + 1) eY eX eZ V_{1y} V_{2z}}{r^3 (\eta r^2 - 1) \eta} \\
 & - 8 \frac{eZ V_{1z} V_{2x}}{r^3 (\eta r^2 - 1)^2 \eta} (((1/8 eX^2 - 1/8 - \alpha^2) r + S (1/8 + (2 - eX^2) \alpha^2)) r^3 \eta^2 \\
 & - r ((1/2 (1 - eX^2) (\alpha^2 - 1/2) + 1/8) r + S (1/8 + (1/2 - eX^2) \alpha^2)) \eta - 1/2 eX^2 (\alpha^2 - 1/4)) \\
 & + \frac{(4\alpha^2 (2Sr\eta - 1) - \eta r^2 + 1) eY eX eZ V_{1z} V_{2y}}{r^3 \eta (\eta r^2 - 1)} \\
 & - \frac{eX S (\eta^2 r^3 + (S (eZ^2 - 1) r - 1 + (4 - 8eZ^2) \alpha^2) r\eta + (eZ^2 - 1) (4\alpha^2 - 1) S) V_{1z} V_{2z}}{r^3 \eta (\eta r^2 - 1)},
 \end{aligned} \tag{1.0.67}$$

$$\begin{aligned}
 A_{2y} = & + \frac{eY S}{r^2} \\
 & - 2 \frac{S\alpha eY eX (rS - 3\eta r^2 + 2) V_{1x}}{r^2 (\eta r^2 - 1)} \\
 & + 4 \frac{S\alpha (eY^2 + 1/2 S (eY^2 - 1) r - 3/2 \eta r^2 (eY^2 - 1/3)) V_{1y}}{r^2 (-\eta r^2 + 1)} \\
 & + 4 \frac{eZ eY S\alpha (1 + 1/2 rS - 3/2 \eta r^2) V_{1z}}{r^2 (-\eta r^2 + 1)} \\
 & - 2 \frac{eY eX \alpha (S - r) V_{2x}}{r^2 (-\eta r^2 + 1)} \\
 & + 2 \frac{(eY^2 - 1) (S - r) \alpha V_{2y}}{r^2 (\eta r^2 - 1)} \\
 & + 2 \frac{eZ eY \alpha (S - r) V_{2z}}{r^2 (\eta r^2 - 1)} \\
 & + 3 \frac{eY S V_{1x}^2}{r^2 (\eta r^2 - 1)^2} (-2 (1/6 - eX^2) \eta^2 r^4 - r^3 \eta eX^2 S + eX^2 - 1/3 \\
 & - 4 \eta (-\alpha^2 - 3/4) eX^2 - 1/6) r^2 - 4 eX^2 S (\alpha^2 - 1/4) r) \\
 & - 3 \frac{eY S V_{1y}^2}{r^2 (\eta r^2 - 1)^2} (-2 \eta^2 (eY^2 - 2/3) r^4 + r^3 \eta (eY^2 - 1) S - eY^2 + 1/3 \\
 & - 4 ((\alpha^2 - 3/4) (eY^2 - 1) - 1/3) \eta r^2 + (eY^2 - 1) (4\alpha^2 - 1) S r) \\
 & - 3 \frac{eY S V_{1z}^2}{r^2 (\eta r^2 - 1)^2} (-2 (eZ^2 - 1/6) \eta^2 r^4 + r^3 \eta S eZ^2 - eZ^2 + 1/3 \\
 & + (-2/3 + (-4\alpha^2 + 3) eZ^2) \eta r^2 + S eZ^2 (4\alpha^2 - 1) r) \\
 & - 6 \frac{eX S V_{1x} V_{1y}}{r^2 (\eta r^2 - 1)^2} (\eta^2 (1/2 - 2eY^2) r^4 + S \eta (eY^2 - 1/2) r^3 - eY^2 \\
 & - 4 ((eY^2 - 1/2) (\alpha^2 - 3/4) - 1/4) \eta r^2 + 4 r S (eY^2 - 1/2) (\alpha^2 - 1/4)) \\
 & - 6 \frac{eY eZ eX S (r^3 \eta S + (3 - 4\alpha^2) \eta r^2 + S (4\alpha^2 - 1) r - 1 - 2 r^4 \eta^2) V_{1x} V_{1z}}{r^2 (\eta r^2 - 1)^2} \\
 & - 6 \frac{eZ S V_{1y} V_{1z}}{r^2 (\eta r^2 - 1)^2} (\eta^2 (1/2 - 2eY^2) r^4 + S \eta (eY^2 - 1/2) r^3 - eY^2 \\
 & - 4 ((eY^2 - 1/2) (\alpha^2 - 3/4) - 1/4) \eta r^2 + 4 r S (eY^2 - 1/2) (\alpha^2 - 1/4)) \\
 & + 3 \frac{eY (S - r) (-\eta (4/3 \alpha^2 + eX^2) r^2 + 4 (1/3 - eX^2) \alpha^2 + eX^2) V_{2x}^2}{r^2 (\eta r^2 - 1)^2} \\
 & - 3 \frac{(\eta r^2 + 4\alpha^2 - 1) (S - r) eY (eY^2 - 1) V_{2y}^2}{r^2 (\eta r^2 - 1)^2} \\
 & - 3 \frac{eY (S - r) (r^2 (eZ^2 + 4/3 \alpha^2) \eta + (4eZ^2 - 4/3) \alpha^2 - eZ^2) V_{2z}^2}{r^2 (\eta r^2 - 1)^2} \\
 & - 6 eX \frac{(S - r) (\eta (eY^2 - 2/3 \alpha^2 - 1/2) r^2 + (4eY^2 - 4/3) \alpha^2 - eY^2 + 1/2) V_{2x} V_{2y}}{r^2 (\eta r^2 - 1)^2}
 \end{aligned}$$

$$\begin{aligned}
 & -6 \frac{(\eta r^2 + 4\alpha^2 - 1)(S - r) eX eZ eY V_{2x} V_{2z}}{r^2 (\eta r^2 - 1)^2} \\
 & -6 \frac{eZ (S - r) (\eta (eY^2 - 2/3\alpha^2 - 1/2) r^2 + (4eY^2 - 4/3)\alpha^2 - eY^2 + 1/2) V_{2y} V_{2z}}{r^2 (\eta r^2 - 1)^2} \\
 & - \frac{eY S V_{1x} V_{2x}}{r^3 \eta (\eta r^2 - 1)} (\eta^2 r^3 + (r (eX^2 - 1) S - 1 + 4(1 - 2eX^2)\alpha^2) r \eta) \\
 & - 4(1 - eX^2)(\alpha^2 - 1/4) S \\
 & 8 \frac{eX V_{1x} V_{2y}}{r^3 \eta (\eta r^2 - 1)^2} ((\eta r^2 - 1) ((Sr\eta - 1/2)\alpha^2 - 1/8\eta r^2 + 1/8) eY^2 \\
 & - 2r\eta ((r^2 (S - r/2)\eta - S/4 - r/4)\alpha^2 + 1/16 (\eta r^2 - 1) (S - r))) \\
 & + \frac{(4\alpha^2 (2Sr\eta - 1) - \eta r^2 + 1) eZ eX eY V_{1x} V_{2z}}{r^3 (\eta r^2 - 1) \eta} \\
 & - \frac{eX S V_{1y} V_{2x}}{r^3 \eta (\eta r^2 - 1)^2} (((\eta r^2 - 1) (\eta r^2 + 4\alpha^2 - 1) eY^2 + (8r^4\eta^2 + 4\eta r^2)\alpha^2 + r^4\eta^2 - \eta r^2) S \\
 & - (8\alpha^2 (\eta r^2 - 1) eY^2 + \eta r^2 (\eta r^2 + 12\alpha^2 - 1)) \eta r) \\
 & + \frac{(4\alpha^2 (2Sr\eta - 1) - \eta r^2 + 1) eY (eY^2 - 1) V_{1y} V_{2y}}{\eta r^3 (\eta r^2 - 1)} \\
 & - \frac{eX S V_{1y} V_{2z}}{r^3 \eta (\eta r^2 - 1)^2} (((\eta r^2 - 1) (\eta r^2 + 4\alpha^2 - 1) eY^2 + (8r^4\eta^2 + 4\eta r^2)\alpha^2 + r^4\eta^2 - \eta r^2) S \\
 & - (8\alpha^2 (\eta r^2 - 1) eY^2 + \eta r^2 (\eta r^2 + 12\alpha^2 - 1)) \eta r) \\
 & + \frac{(4\alpha^2 (2Sr\eta - 1) - \eta r^2 + 1) eZ eX eY V_{1z} V_{2x}}{r^3 (\eta r^2 - 1) \eta} \\
 & + 8 \frac{eZ V_{1z} V_{2y}}{r^3 \eta (\eta r^2 - 1)^2} ((\eta r^2 - 1) ((Sr\eta - 1/2)\alpha^2 - 1/8\eta r^2 + 1/8) eY^2 \\
 & - 2r\eta ((r^2 (S - r/2)\eta - S/4 - r/4)\alpha^2 + 1/16 (\eta r^2 - 1) (S - r))) \\
 & - \frac{eY S V_{1z} V_{2z}}{r^3 \eta (\eta r^2 - 1)} (\eta^2 r^3 + (S (eZ^2 - 1) r - 1 + (-8eZ^2 + 4)\alpha^2) r \eta) \\
 & + (eZ^2 - 1) (4\alpha^2 - 1) S,
 \end{aligned}$$

(1.0.68)

$$\begin{aligned}
 A_{2z} = & + \frac{eZ S}{r^2} \\
 & - 2 \frac{S(-3\eta r^2 + rS + 2) \alpha eX eZ V_{1x}}{r^2 (\eta r^2 - 1)} \\
 & + 4 \frac{eY eZ S (1 + 1/2 rS - 3/2 \eta r^2) \alpha V_{1y}}{r^2 (-\eta r^2 + 1)} \\
 & - 4 \frac{\alpha S (1/2 r (1 - eZ^2) S - eZ^2 - 3/2 \eta (1/3 - eZ^2) r^2) V_{1z}}{r^2 (-\eta r^2 + 1)} \\
 & - 2 \frac{eX eZ (S - r) \alpha V_{2x}}{r^2 (-\eta r^2 + 1)} \\
 & + 2 \frac{eZ eY (S - r) \alpha V_{2y}}{r^2 (\eta r^2 - 1)} \\
 & - 2 \frac{(1 - eZ^2) (S - r) \alpha V_{2z}}{r^2 (\eta r^2 - 1)} \\
 & + 3 \frac{eZ S V_{1x}^2}{(\eta r^2 - 1)^2 r^2} (-2\eta^2 (1/6 - eX^2) r^4 - r^3 eX^2 \eta S + eX^2 - 1/3 \\
 & + 4 ((\alpha^2 - 3/4) eX^2 + 1/6) \eta r^2 - 4 (\alpha^2 - 1/4) S eX^2 r) \\
 & - 3 \frac{eZ S V_{1y}^2}{(\eta r^2 - 1)^2 r^2} (-2 (-1/6 + eY^2) \eta^2 r^4 + r^3 eY^2 \eta S - eY^2 + 1/3 \\
 & + (-2/3 + (-4\alpha^2 + 3) eY^2) \eta r^2 + S eY^2 (4\alpha^2 - 1) r) \\
 & - 3 \frac{eZ S V_{1z}^2}{(\eta r^2 - 1)^2 r^2} (-2\eta^2 (eZ^2 - 2/3) r^4 + S \eta (eZ^2 - 1) r^3 - eZ^2 + 1/3 \\
 & - 4\eta ((\alpha^2 - 3/4) (eZ^2 - 1) - 1/3) r^2 + S (4\alpha^2 - 1) (eZ^2 - 1) r) \\
 & - 6 \frac{eX eY eZ S (-1 - 2r^4 \eta^2 + S \eta r^3 + (-4\alpha^2 + 3) \eta r^2 + S (4\alpha^2 - 1) r) V_{1x} V_{1y}}{(\eta r^2 - 1)^2 r^2} \\
 & - 6 \frac{eX S V_{1x} V_{1z}}{(\eta r^2 - 1)^2 r^2} (\eta^2 (-2eZ^2 + 1/2) r^4 + S \eta (eZ^2 - 1/2) r^3 - eZ^2 \\
 & - 4\eta ((\alpha^2 - 3/4) eZ^2 - 1/2 \alpha^2 + 1/8) r^2 + 4S (\alpha^2 - 1/4) (eZ^2 - 1/2) r) \\
 & - 6 \frac{eY S V_{1y} V_{1z}}{(\eta r^2 - 1)^2 r^2} (\eta^2 (-2eZ^2 + 1/2) r^4 + S \eta (eZ^2 - 1/2) r^3 - eZ^2 \\
 & - 4\eta ((\alpha^2 - 3/4) eZ^2 - 1/2 \alpha^2 + 1/8) r^2 + 4S (\alpha^2 - 1/4) (eZ^2 - 1/2) r) \\
 & + 3 \frac{eZ (-\eta (eX^2 + 4/3 \alpha^2) r^2 + 4 (1/3 - eX^2) \alpha^2 + eX^2) (S - r) V_{2x}^2}{r^2 (\eta r^2 - 1)^2} \\
 & - 3 \frac{(\eta (eY^2 + 4/3 \alpha^2) r^2 + (4eY^2 - 4/3) \alpha^2 - eY^2) (S - r) eZ V_{2y}^2}{r^2 (\eta r^2 - 1)^2} \\
 & - 3 \frac{(S - r) (\eta r^2 + 4\alpha^2 - 1) eZ (eZ^2 - 1) V_{2z}^2}{(\eta r^2 - 1)^2 r^2} \\
 & - 6 \frac{(S - r) (\eta r^2 + 4\alpha^2 - 1) eY eX eZ V_{2x} V_{2y}}{(\eta r^2 - 1)^2 r^2}
 \end{aligned}$$

$$\begin{aligned}
 & -6 \frac{(\eta (eZ^2 - 2/3\alpha^2 - 1/2) r^2 + (4eZ^2 - 4/3)\alpha^2 - eZ^2 + 1/2) eX (S - r) V_{2x} V_{2z}}{r^2 (\eta r^2 - 1)^2} \\
 & -6 \frac{(S - r) (\eta (eZ^2 - 2/3\alpha^2 - 1/2) r^2 + (4eZ^2 - 4/3)\alpha^2 - eZ^2 + 1/2) eY V_{2y} V_{2z}}{r^2 (\eta r^2 - 1)^2} \\
 & - \frac{eZ S V_{1x} V_{2x}}{r^3 (\eta r^2 - 1) \eta} ((\eta^2 r^3 - 4S(-eX^2 + 1)(\alpha^2 - 1/4)) \\
 & + (S(eX^2 - 1)r - 1 + 4(-2eX^2 + 1)\alpha^2) r\eta) \\
 & + \frac{eY eZ eX (4\alpha^2(2S\eta r - 1) - \eta r^2 + 1) V_{1x} V_{2y}}{(\eta r^2 - 1) r^3 \eta} \\
 & + 8 \frac{eX V_{1x} V_{2z}}{(\eta r^2 - 1)^2 r^3 \eta} (((-1/8 eZ^2 + \alpha^2 + 1/8) r + S(-1/8 + (eZ^2 - 2)\alpha^2)) r^3 \eta^2 \\
 & - (1/2 r (eZ^2 - 1)\alpha^2 - S/8 + (eZ^2 - 1/2)(S\alpha^2 - r/4)) r\eta + 1/2 eZ^2 (\alpha^2 - 1/4)) \\
 & + \frac{eY eZ eX (4\alpha^2(2S\eta r - 1) - \eta r^2 + 1) V_{1y} V_{2x}}{(\eta r^2 - 1) r^3 \eta} \\
 & - \frac{eZ S V_{1y} V_{2y}}{r^3 (\eta r^2 - 1) \eta} (\eta^2 r^3 + (4\alpha^2 - 1)(eY^2 - 1) S \\
 & + (S(eY^2 - 1)r - 1 + 4(-2eY^2 + 1)\alpha^2) r\eta) \\
 & + 8 \frac{eY V_{1y} V_{2z}}{(\eta r^2 - 1)^2 r^3 \eta} (((-1/8 eZ^2 + \alpha^2 + 1/8) r + S(-1/8 + (eZ^2 - 2)\alpha^2)) r^3 \eta^2 \\
 & - (1/2 r (eZ^2 - 1)\alpha^2 - S/8 + (eZ^2 - 1/2)(S\alpha^2 - r/4)) r\eta + 1/2 eZ^2 (\alpha^2 - 1/4)) \\
 & - \frac{eX S V_{1z} V_{2x}}{(\eta r^2 - 1)^2 r^3 \eta} (-r^5 \eta^3 + (S(8\alpha^2 + eZ^2 + 1)r + 1 + (-8eZ^2 - 12)\alpha^2) r^3 \eta^2 \\
 & + 4(S((eZ^2 + 1)\alpha^2 - 1/2 eZ^2 - 1/4)r + 2\alpha^2 eZ^2) r\eta + (-4\alpha^2 + 1)eZ^2 S) \\
 & - \frac{eY S V_{1z} V_{2y}}{(\eta r^2 - 1)^2 r^3 \eta} (-r^5 \eta^3 + (S(8\alpha^2 + eZ^2 + 1)r + 1 + (-8eZ^2 - 12)\alpha^2) r^3 \eta^2 \\
 & + 4(S((eZ^2 + 1)\alpha^2 - 1/2 eZ^2 - 1/4)r + 2\alpha^2 eZ^2) r\eta + (-4\alpha^2 + 1)eZ^2 S) \\
 & + \frac{(4\alpha^2(2S\eta r - 1) - \eta r^2 + 1) eZ (eZ^2 - 1) V_{1z} V_{2z}}{(\eta r^2 - 1) r^3 \eta},
 \end{aligned} \tag{1.0.69}$$

onde os termos  $r$ ,  $eX$ ,  $eY$  e  $eZ$  são dados por:

$$\left\{ \begin{array}{l} r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}, \\ eX = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}, \\ eY = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}, \\ eZ = \frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}. \end{array} \right.$$

Por uma questão de simplicidade iremos impor a condição de solução planar zerando todas as componentes no eixo  $z$ . Portanto, o conjunto de equações, no plano  $xy$ , de duas partículas pontuais que interagem entre si através do campo eletromagnético tomando os tempos  $t_r$  e  $t_a$  como infinitamente pequenos são:

$$\begin{aligned}
 A_{1x} = & -\frac{eXS}{\eta r^2} + \frac{2S\alpha eY^2(S\eta r - 1)V_{1x}}{\eta r^2(\eta r^2 - 1)} + \frac{2S\alpha(((2 - 3eY^2)r^2 + SreY^2)\eta - 2eX^2)V_{2x}}{\eta r^2(\eta r^2 - 1)} \\
 & - \frac{2S\alpha eXeY(S\eta - 1)V_{1y}}{\eta r^2(\eta r^2 - 1)} - \frac{2S\alpha eXeY(2 + r(S - 3r)\eta)V_{2y}}{\eta r^2(\eta r^2 - 1)} \\
 & + \frac{3SeXeY^2(\eta r^2 + 4\alpha^2 - 1)(S\eta r - 1)V_{1x}^2}{\eta r^2(\eta r^2 - 1)^2} \\
 & - \frac{3eXS(r^2(eY^2 + (4/3)\alpha^2)\eta + (4eY^2 - 4/3)\alpha^2 - eY^2)(S\eta r - 1)V_{1y}^2}{\eta r^2(\eta r^2 - 1)^2} \\
 & + \frac{6eYS(r^2(eY^2 - 1/2 + (2/3)\alpha^2)\eta + (4\alpha^2 - 1)eY^2 - (8/3)\alpha^2 + 1/2)(S\eta r - 1)V_{1x}V_{1y}}{\eta r^2(\eta r^2 - 1)^2} \\
 & - \frac{3eXS}{\eta r^2(\eta r^2 - 1)^2} (((2/3 - 2eY^2)r^4 + Sr^3eY^2)\eta^2 - eY^2 + 2/3 \\
 & + 4\eta r (((3/4 - \alpha^2)eY^2 - 1/3)r + eY^2S(\alpha^2 - 1/4))) V_{2x}^2 \\
 & + \frac{3eXS}{\eta r^2(\eta r^2 - 1)^2} (((1/3 - 2eY^2)r^4 + Sr^3eY^2)\eta^2 - eY^2 + 1/3 \\
 & + 4\eta r (((3/4 - \alpha^2)eY^2 - 1/6)r + eY^2S(\alpha^2 - 1/4))) V_{2y}^2 \\
 & - \frac{6eYS}{\eta r^2(\eta r^2 - 1)^2} (((3/2 - 2eY^2)r + S(eY^2 - 1/2))r^3\eta^2 + eX^2 \\
 & + 4\eta r (((3/4 - \alpha^2)eY^2 + (1/2)\alpha^2 - 5/8)r + S(\alpha^2 - 1/4)(eY^2 - 1/2))) V_{2x}V_{2y} \\
 & - \frac{eXeY^2S((\eta r^2 + 4\alpha^2 - 1)S - 8\alpha^2r)V_{1x}V_{2x}}{\eta r^3(\eta r^2 - 1)} \\
 & - \frac{eYS}{\eta r^3(\eta r^2 - 1)^2} (S\eta^2r^4(8\alpha^2 + eY^2) - \eta r^3(1 + 8\alpha^2(eY^2 + 1))) \\
 & + S\eta r^2(4\alpha^2eY^2 - 2eY^2 + 1) + r(1 + \alpha^2(8eY^2 - 4)) + eX^2S(4\alpha^2 - 1)) V_{1x}V_{2y} \\
 & - \frac{eYS}{\eta r^3(\eta r^2 - 1)^2} (\eta^2r^5 + S\eta^2r^4(eY^2 - 8\alpha^2 - 2) + \eta r^3(\alpha^2(20 - 8eY^2) - 1)) \\
 & + S\eta r^2(4\alpha^2(eY^2 - 2) - 2eY^2 + 3) - 8eX^2\alpha^2r + eX^2S(4\alpha^2 - 1)) V_{1y}V_{2x} \\
 & + \frac{eXS}{\eta r^3(\eta r^2 - 1)} (\eta r^3 - S\eta r^2eX^2 + r(\alpha^2(4 - 8eY^2) - 1) + eX^2S(1 - 4\alpha^2)) V_{1y}V_{2y},
 \end{aligned} \tag{1.0.70}$$

$$\begin{aligned}
 A_{1y} = & -\frac{eYS}{\eta r^2} - \frac{2SaeXeY(S\eta r - 1)V_{1x}}{\eta r^2(\eta r^2 - 1)} - \frac{2SaeXeY(2 + r(S - 3r)\eta)V_{2x}}{\eta r^2(\eta r^2 - 1)} \\
 & + \frac{2SaeX^2(S\eta r - 1)V_{1y}}{\eta r^2(\eta r^2 - 1)} - \frac{2S\alpha(\eta r(r(1 - 3eY^2) + SeY^2 - S) + 2eY^2)V_{2y}}{\eta r^2(\eta r^2 - 1)} \\
 & + \frac{3SeY(S\eta r - 1)(1 - \eta r^2((4/3)\alpha^2 + eX^2) + (4\alpha^2 - 1)eY^2 - (8/3)\alpha^2)V_{1x}^2}{\eta r^2(\eta r^2 - 1)^2} \\
 & + \frac{3SeYeX^2(\eta r^2 + 4\alpha^2 - 1)(S\eta r - 1)V_{1y}^2}{\eta r^2(\eta r^2 - 1)^2} \\
 & - \frac{6eXS(\eta r^2(eY^2 - 1/2 - (2/3)\alpha^2) + (4eY^2 - 4/3)\alpha^2 - eY^2 + 1/2)(S\eta r - 1)V_{1x}V_{1y}}{\eta r^2(\eta r^2 - 1)^2} \\
 & - \frac{3eYS}{\eta r^2(\eta r^2 - 1)^2} (\eta^2 r^3 (r (5/3 - 2eY^2) - eX^2 S) - eY^2 + 2/3 \\
 & \eta r ((4r ((3/4 - \alpha^2) eY^2 + \alpha^2 - 7/12)) - 4eX^2 S (\alpha^2 - 1/4))) V_{2x}^2 \\
 & + \frac{3eYS}{\eta r^2(\eta r^2 - 1)^2} (\eta^2 r^3 (r (4/3 - 2eY^2) - eX^2 S) - eY^2 + 1/3 \\
 & \eta r ((4r ((3/4 - \alpha^2) eY^2 + \alpha^2 - 5/12)) - 4eX^2 S (\alpha^2 - 1/4))) V_{2y}^2 \\
 & + \frac{6eXS}{\eta r^2(\eta r^2 - 1)^2} (\eta^2 r^3 (r (1/2 - 2eY^2) + S (eY^2 - 1/2)) - eY^2 \\
 & + 4\eta r (r ((3/4 - \alpha^2) eY^2 + (1/2)\alpha^2 - 1/8) + S (\alpha^2 - 1/4) (eY^2 - 1/2))) V_{2x}V_{2y} \\
 & + \frac{eYS(\eta r^3 - S\eta eY^2 r^2 + r(4\alpha^2(2eY^2 - 1) - 1) + (1 - 4\alpha^2)eY^2 S)V_{1x}V_{2x}}{\eta r^3(\eta r^2 - 1)} \\
 & + \frac{eXS}{\eta r^3(\eta r^2 - 1)^2} (S\eta^2 r^4 (8\alpha^2 + eY^2 + 1) - \eta^2 r^5 + \eta r^3 (1 - \alpha^2 (8eY^2 + 12))) \\
 & + 4S\eta r^2 ((eY^2 + 1) \alpha^2 - (1/2)eY^2 - 1/4) + 8r\alpha^2 eY^2 + eY^2 S(1 - 4\alpha^2)) V_{1x}V_{2y} \\
 & + \frac{eXS}{\eta r^3(\eta r^2 - 1)^2} (\eta r^3 (1 + \alpha^2 (16 - 8eY^2)) - S\eta^2 r^4 (8\alpha^2 + eX^2) + eY^2 S(1 - 4\alpha^2) \\
 & 2S\eta r^2 (1/2 - 2\alpha^2 eX^2 - eY^2) + r (\alpha^2 (8eY^2 - 4) - 1)) V_{1y}V_{2x} \\
 & - \frac{eX^2 eYS (S (\eta r^2 + 4\alpha^2 - 1) - 8\alpha^2 r) V_{1y}V_{2y}}{\eta r^3(\eta r^2 - 1)}, \tag{1.0.71}
 \end{aligned}$$



$$\begin{aligned}
 A_{2x} = & \frac{eXS}{r^2} + \frac{2S\alpha(\eta r^2(2 - 3eY^2) + SreY^2 - 2eX^2)V_{1x}}{r^2(\eta r^2 - 1)} + \frac{(2eY^2S\alpha(Sr - 1))V_{2x}}{\eta r^2 - 1} \\
 & - \frac{2S\alpha eXeY(2 - 3\eta r^2 + Sr)V_{1y}}{r^2(\eta r^2 - 1)} - \frac{2S\alpha eXeY(Sr - 1)V_{2y}}{r^2(\eta r^2 - 1)} \\
 & + \frac{3eXS}{r^2(\eta r^2 - 1)^2} (S\eta eY^2 r^3 - 2\eta^2 r^4 (eY^2 - 1/3) - eY^2 + 2/3 \\
 & + \eta r^2((3 - 4\alpha^2)eY^2 - 4/3) + SreY^2(4\alpha^2 - 1)) V_{1x}^2 \\
 & - \frac{3eXS}{r^2(\eta r^2 - 1)^2} (S\eta eY^2 r^3 - 2\eta^2 r^4 (eY^2 - 1/6) - eY^2 + 1/3 \\
 & + \eta r^2((3 - 4\alpha^2)eY^2 - 2/3) + SreY^2(4\alpha^2 - 1)) V_{1y}^2 \\
 & + \frac{6SeY}{r^2(\eta r^2 - 1)^2} (S\eta r^3 (eY^2 - 1/2) - 2\eta^2 r^4 (eY^2 - 3/4) + eX^2 \\
 & - \eta r^2(4eY^2(\alpha^2 - 3/4) - 2\alpha^2 + 5/2) + 4Sr(eY^2 - 1/2)(\alpha^2 - 1/4)) V_{1x}V_{1y} \\
 & - \frac{3eXeY^2S(\eta r^2 + 4\alpha^2 - 1)(Sr - 1)V_{2x}^2}{r^2(\eta r^2 - 1)^2} \\
 & + \frac{3eXS(\eta r^2(eY^2 + (4/3)\alpha^2) + (4eY^2 - 4/3)\alpha^2 - eY^2)(Sr - 1)V_{2y}^2}{r^2(\eta r^2 - 1)^2} \\
 & - \frac{6eYS(\eta r^2(eY^2 - 1/2 + (2/3)\alpha^2) + (4\alpha^2 - 1)eY^2 - (8/3)\alpha^2 + 1/2)(Sr - 1)V_{2x}V_{2y}}{r^2(\eta r^2 - 1)^2} \\
 & + \frac{eXeY^2S(S(\eta r^2 + 4\alpha^2 - 1) - 8\alpha^2\eta r)V_{1x}V_{2x}}{\eta r^3(\eta r^2 - 1)} \\
 & + \frac{eYS}{\eta r^3(\eta r^2 - 1)^2} (\eta^3 r^5 + \eta^2 r^3 (Sr (eY^2 - 8\alpha^2 - 2) - 1 + (20 - 8eY^2) \alpha^2) \\
 & + 4\eta r (Sr ((eY^2 - 2) \alpha^2 - (1/2)eY^2 + 3/4) - 2\alpha^2 eX^2) + eX^2 S(4\alpha^2 - 1)) V_{1x}V_{2y} \\
 & + \frac{eYS}{\eta r^3(\eta r^2 - 1)^2} (\eta^2 r^3 (Sr (8\alpha^2 + eY^2) - 1 - 8\alpha^2 (eY^2 + 1)) + eX^2 S(4\alpha^2 - 1) \\
 & + 4\eta r (Sr ((eY^2 \alpha^2 - (1/2)eY^2 + 1/4) + 1/4 + \alpha^2(2eY^2 - 1))) V_{1y}V_{2x}, \quad (1.0.72)
 \end{aligned}$$

$$\begin{aligned}
 A_{2y} = & \frac{eYS}{r^2} - \frac{2S\alpha eYeX(Sr - 3\eta r^2 + 2)V_{1x}}{r^2(\eta r^2 - 1)} - \frac{2S\alpha eYeX(Sr - 1)V_{2x}}{r^2(\eta r^2 - 1)} \\
 & - \frac{2S\alpha(2eY^2 - eX^2Sr + \eta r^2(1 - 3eY^2))V_{1y}}{r^2(\eta r^2 - 1)} + \frac{2S\alpha eX^2(Sr - 1)V_{2y}}{r^2(\eta r^2 - 1)} \\
 & + \frac{3eYS}{r^2(\eta r^2 - 1)^2} (\eta^2 r^4 (5/3 - 2eY^2) - S\eta eX^2 r^3 - eY^2 + 2/3 \\
 & - \eta r^2 (4eY^2 (\alpha^2 - 3/4) - 4\alpha^2 + 7/3) + r (eY^2 S (4\alpha^2 - 1) + S (1 - 4\alpha^2))) V_{1x}^2 \\
 & - \frac{3eYS}{r^2(\eta r^2 - 1)^2} (2\eta^2 r^4 (2/3 - eY^2) - S\eta eX^2 r^3 - eY^2 + 1/3 \\
 & - 4\eta r^2 (eY^2 (\alpha^2 - 3/4) - \alpha^2 + 5/12) + r (eY^2 S (4\alpha^2 - 1) + S (1 - 4\alpha^2))) V_{1y}^2 \\
 & - \frac{6eXS}{r^2(\eta r^2 - 1)^2} (\eta^2 r^4 (1/2 - 2eY^2) + S\eta r^3 (eY^2 - 1/2) - eY^2 \\
 & - \eta r^2 ((4eY^2 (\alpha^2 - 3/4)) - 2\alpha^2 + 1/2) + 4Sr (eY^2 - 1/2) (\alpha^2 - 1/4)) V_{1x}V_{1y} \\
 & - \frac{3eYS(\eta r^2 (eY^2 - (4/3)\alpha^2 - 1) + eY^2 (4\alpha^2 - 1) - (8/3)\alpha^2 + 1) (Sr - 1) V_{2x}^2}{r^2(\eta r^2 - 1)^2} \\
 & - \frac{3eYeX^2S(\eta r^2 + 4\alpha^2 - 1)(Sr - 1)V_{2y}^2}{r^2(\eta r^2 - 1)^2} \\
 & + \frac{6eXS(\eta r^2 (eY^2 - 1/2 - (2/3)\alpha^2) + (4eY^2 - 4/3)\alpha^2 - eY^2 + 1/2) (Sr - 1) V_{2x}V_{2y}}{r^2(\eta r^2 - 1)^2} \\
 & - \frac{eYS(\eta^2 r^3 - \eta r (SreY^2 + 4\alpha^2 (1 - 2eY^2) + 1) + eY^2 S (1 - 4\alpha^2)) V_{1x}V_{2x}}{\eta r^3(\eta r^2 - 1)} \\
 & - \frac{eXS}{\eta r^3(\eta r^2 - 1)^2} (\eta^2 r^3 (1 - Sr (8\alpha^2 + eX^2) + (16 - 8eY^2) \alpha^2) + eY^2 S (1 - 4\alpha^2) \\
 & + 2\eta r (Sr (1/2 - 2\alpha^2 eX^2 - eY^2) + (4eY^2 - 2) \alpha^2 - 1/2)) V_{1x}V_{2y} \\
 & - \frac{eXS}{\eta r^3(\eta r^2 - 1)^2} (\eta^2 r^3 (Sr (8\alpha^2 + eY^2 + 1) + 1 - (8eY^2 + 12) \alpha^2) - \eta^3 r^5 \\
 & + 4\eta r (Sr ((eY^2 + 1) \alpha^2 - (1/2)eY^2 - 1/4) + 2eY^2 \alpha^2) + eY^2 S (1 - 4\alpha^2)) V_{1y}V_{2x} \\
 & + \frac{eYeX^2S(S(\eta r^2 + 4\alpha^2 - 1) - 8\alpha^2 \eta r) V_{1y}V_{2y}}{\eta r^3(\eta r^2 - 1)}, \tag{1.0.73}
 \end{aligned}$$

onde os termos  $r$ ,  $eX$  e  $eY$  são dados por:

$$\left\{ \begin{array}{l} r = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \\ eX = \frac{x_2 - x_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}, \\ eY = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}. \end{array} \right.$$

O sistema de duas partículas pontuais carregadas, com  $\alpha = 0$ , no plano cartesiano  $xy$  é dado por:

$$\begin{aligned}
 A_{1x} = & -\frac{eXS}{\eta r^2} + \frac{3eXeY^2(S - \eta r)V_{1x}^2}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{3eXeY^2(S - \eta r)V_{1y}^2}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{3eY(S - \eta r)(2eX^2 - 1)V_{1x}V_{1y}}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{eXS(\eta r(2r(2eY^2 - eX^2) - 3SeY^2) + 3eX^2 - 1)V_{2x}^2}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{eXS(\eta(3SreY^2 - r^2(6eY^2 - 1)) + 3eY^2 - 1)V_{2y}^2}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{3eYS(\eta r(r(1 - 4eX^2) + S(2eX^2 - 1)) + 2eX^2)V_{2x}V_{2y}}{\eta r^2(1 - \eta r^2)} \\
 & + \frac{eXS(r - eX^2S)V_{1y}V_{2y}}{\eta r^3} - \frac{eXeY^2V_{1x}V_{2x}}{\eta r^3} \\
 & + \frac{eY(\eta r^2eY^2 - Sr + eX^2)V_{1x}V_{2y}}{\eta r^3(1 - \eta r^2)} \\
 & + \frac{eYS(eX^2S - \eta r^2((eX^2 + 1)S - r))V_{1y}V_{2x}}{\eta r^3(1 - \eta r^2)}, \tag{1.0.74}
 \end{aligned}$$

$$\begin{aligned}
 A_{1y} = & -\frac{eYS}{\eta r^2} - \frac{3eYeX^2(S - \eta r)V_{1x}^2}{\eta r^2(1 - \eta r^2)} \\
 & + \frac{3eYeX^2(S - \eta r)V_{1y}^2}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{(3eX(2eY^2 - 1))(S - \eta r)V_{1x}V_{1y}}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{eYS(\eta(3SreX^2 - (6eX^2 - 1)r^2) + 3eX^2 - 1)V_{2x}^2}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{eYS(\eta r(2r(2eX^2 - eY^2) - 3eX^2S) + 3eY^2 - 1)V_{2y}^2}{\eta r^2(1 - \eta r^2)} \\
 & - \frac{3eXS(\eta r(r(1 - 4eY^2) + S(2eY^2 - 1)) + 2eY^2)V_{2x}V_{2y}}{\eta r^2(1 - \eta r^2)} \\
 & + \frac{eYS(r - eY^2S)V_{1x}V_{2x}}{\eta r^3} - \frac{eYeX^2V_{1y}V_{2y}}{\eta r^3} \\
 & + \frac{eXS(eY^2S - \eta r^2((eY^2 + 1)S - r))V_{1x}V_{2y}}{\eta r^3(1 - \eta r^2)} \\
 & + \frac{eX(eX^2\eta r^2 - Sr + eY^2)V_{1y}V_{2x}}{\eta r^3(1 - \eta r^2)}, \tag{1.0.75}
 \end{aligned}$$

$$\begin{aligned}
 A_{2x} = & \frac{eXS}{\eta r^2} \\
 & + \frac{eXS(3eX^2 - 1 - 3SreY^2 - 2(3eX^2 - 2)\eta r^2)V_{1x}^2}{r^2(1 - \eta r^2)} \\
 & + \frac{eXS(3SreY^2 + 3eY^2 - 1 - (6eY^2 - 1)\eta r^2)V_{1y}^2}{r^2(1 - \eta r^2)} \\
 & + \frac{3eYS(\eta r^2(1 - 4eX^2) + Sr(2eX^2 - 1) + 2eX^2)V_{1x}V_{1y}}{r^2(1 - \eta r^2)} \\
 & - \frac{3eXeY^2(S - r)V_{2x}^2}{r^2(1 - \eta r^2)} + \frac{3eXeY^2(S - r)V_{2y}^2}{r^2(1 - \eta r^2)} \\
 & + \frac{3eY(2eX^2 - 1)(S - r)V_{2x}V_{2y}}{r^2(1 - \eta r^2)} \\
 & + \frac{eXeY^2V_{1x}V_{2x}}{\eta r^3} - \frac{eXS(\eta r - eX^2S)V_{1y}V_{2y}}{\eta r^3} \\
 & - \frac{eYS(\eta^2 r^3 - S\eta r^2(eX^2 + 1) + eX^2S)V_{1x}V_{2y}}{\eta r^3(1 - \eta r^2)} \\
 & - \frac{eY(eX^2 + \eta(eY^2 r^2 - Sr))V_{1y}V_{2x}}{\eta r^3(1 - \eta r^2)}, \tag{1.0.76}
 \end{aligned}$$

$$\begin{aligned}
 A_{2y} = & \frac{eYS}{r^2} \\
 & + \frac{eYS(3SreX^2 + 3eX^2 - 1 - \eta r^2(6eX^2 - 1))V_{1x}^2}{r^2(1 - \eta r^2)} \\
 & + \frac{eYS(3eY^2 - 3SreX^2 - 1 - 2\eta r^2(eY^2 - 2eX^2))V_{1y}^2}{r^2(1 - \eta r^2)} \\
 & + \frac{3eXS(\eta r^2(1 - 4eY^2) + Sr(2eY^2 - 1) + 2eY^2)V_{1x}V_{1y}}{r^2(1 - \eta r^2)} \\
 & + \frac{3eYeX^2(S - r)V_{2x}^2}{r^2(1 - \eta r^2)} - \frac{3eYeX^2(S - r)V_{2y}^2}{r^2(1 - \eta r^2)} \\
 & + \frac{3eX(2eY^2 - 1)(S - r)V_{2x}V_{2y}}{r^2(1 - \eta r^2)} \\
 & - \frac{eYS(\eta r - eY^2S)V_{1x}V_{2x}}{\eta r^3} + \frac{eYeX^2V_{1y}V_{2y}}{\eta r^3} \\
 & - \frac{eX(\eta r(eX^2 r - S) + eY^2)V_{1x}V_{2y}}{\eta r^3(1 - \eta r^2)} \\
 & - \frac{eXS(\eta^2 r^3 - S\eta r^2(eY^2 + 1) + eY^2S)V_{1y}V_{2x}}{\eta r^3(1 - \eta r^2)}. \tag{1.0.77}
 \end{aligned}$$

# Apêndice B

## Transformação De Coordenadas

### 2.1 Vetor do centro de massa e relativo

Para um sistema de massas discreto, formado por um conjunto de massas pontuais, o centro de massa é definido como:

$$\mathbf{r}_{cm} = \frac{\sum_i (\mathbf{r}_i m_i)}{\sum_i m_i}$$

- $m_i$  - Massa da  $i$ -ésima partícula,
- $\mathbf{r}_i$  - Posição da  $i$ -ésima partícula.

Portanto,

$$\begin{aligned}x_{cm} &= \frac{m_1 x_1 + m_2 x_2}{m_1 + m_2}, \\y_{cm} &= \frac{m_1 y_1 + m_2 y_2}{m_1 + m_2}, \\z_{cm} &= \frac{m_1 z_1 + m_2 z_2}{m_1 + m_2},\end{aligned}$$

substituindo

$$\frac{m_1}{m_2} = \eta,$$

obtemos

$$x_{cm} = \frac{\eta x_1 + x_2}{\eta + 1}, \tag{2.1.1}$$

$$y_{cm} = \frac{\eta y_1 + y_2}{\eta + 1}, \tag{2.1.2}$$

$$z_{cm} = \frac{\eta z_1 + z_2}{\eta + 1}. \tag{2.1.3}$$

A partir das equações (2.1.1), (2.1.2) e (2.1.3) obtemos a velocidade e a aceleração do centro de massa no plano  $xyz$ :

$$V_x = \frac{dx_{cm}}{dt} = \frac{\eta V_{1x} + V_{2x}}{\eta + 1}, \quad (2.1.4)$$

$$V_y = \frac{dy_{cm}}{dt} = \frac{\eta V_{1y} + V_{2y}}{\eta + 1}, \quad (2.1.5)$$

$$V_z = \frac{dz_{cm}}{dt} = \frac{\eta V_{1z} + V_{2z}}{\eta + 1}, \quad (2.1.6)$$

$$A_x = \frac{d^2x_{cm}}{dt^2} = \frac{\eta A_{1x} + A_{2x}}{\eta + 1}, \quad (2.1.7)$$

$$A_y = \frac{d^2y_{cm}}{dt^2} = \frac{\eta A_{2x} + A_{2y}}{\eta + 1}, \quad (2.1.8)$$

$$A_z = \frac{d^2z_{cm}}{dt^2} = \frac{\eta A_{2z} + A_{2z}}{\eta + 1}. \quad (2.1.9)$$

Na forma matricial temos a posição, velocidade e aceleração do centro de massa como:

$$\mathbf{r}_{cm} = \frac{\eta \mathbf{r}_1 + \mathbf{r}_2}{\eta + 1} = \begin{pmatrix} x_{cm} \\ y_{cm} \\ z_{cm} \end{pmatrix},$$

$$\mathbf{V} = \frac{\eta \mathbf{V}_1 + \mathbf{V}_2}{\eta + 1} = \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix},$$

$$\mathbf{A} = \frac{\eta \mathbf{A}_1 + \mathbf{A}_2}{\eta + 1} = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix}.$$

Definimos o vetor da posição relativa entre duas partículas como:

$$x = x_2 - x_1, \quad (2.1.10)$$

$$y = y_2 - y_1, \quad (2.1.11)$$

$$z = z_2 - z_1. \quad (2.1.12)$$

Derivando as equações (2.1.10), (2.1.11) e (2.1.12) com relação ao tempo obtemos as velocidades e derivando novamente obtêm-se as acelerações, portanto:

$$v_x = V_{2x} - V_{1x}, \quad (2.1.13)$$

$$v_y = V_{2y} - V_{1y}, \quad (2.1.14)$$

$$v_z = V_{2z} - V_{1z}, \quad (2.1.15)$$

$$a_x = A_{2x} - A_{1x}, \quad (2.1.16)$$

$$a_y = A_{2y} - A_{1y}, \quad (2.1.17)$$

$$a_z = A_{2z} - A_{1z}. \quad (2.1.18)$$

Na forma matricial temos a posição, velocidade e aceleração do vetor relativo como:

$$\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1 = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}.$$

Por fim, a última relação que obtemos é a relação das velocidades das partículas em função do vetor centro de massa e o vetor relativo que é expresso da seguinte maneira:

$$V_{1x} = \frac{-v_x + V_x\eta + V_x}{\eta + 1}, \quad (2.1.19)$$

$$V_{1y} = \frac{-v_y + V_y\eta + V_y}{\eta + 1}, \quad (2.1.20)$$

$$V_{1z} = \frac{-v_z + V_z\eta + V_z}{\eta + 1}, \quad (2.1.21)$$

$$V_{2x} = \frac{V_x\eta + V_x + \eta v_x}{\eta + 1}, \quad (2.1.22)$$

$$V_{2y} = \frac{V_y\eta + V_y + \eta v_y}{\eta + 1}, \quad (2.1.23)$$

$$V_{2z} = \frac{V_z\eta + V_z + \eta v_z}{\eta + 1}. \quad (2.1.24)$$

Com estas relações definidas neste apêndice podemos reescrever completamente o sistema de equações diferenciais para as duas cargas pontuais em função do vetor centro de massa e o vetor relativo.

## 2.2 Coordenadas polares

As equações (1.0.70), (1.0.71), (1.0.72) e (1.0.73) do apêndice (A) podem ser expressas pelos vetores do centro de massa, relativo e projetados nas coordenadas polares. Consideremos o plano  $xy$  onde vetor posição para as coordenadas cartesianas  $x$  e  $y$  em função das coordenadas polares  $r$  e  $\theta$  é descrita pela seguinte relação trigonométrica:

$$x = r \cos(\theta), \quad (2.2.25)$$

$$y = r \sin(\theta), \quad (2.2.26)$$

O vetor  $\mathbf{r}$  na coordenada cartesiana é:

$$\mathbf{r} = x\hat{x} + y\hat{y}, \quad (2.2.27)$$

e o versor  $\hat{r}$  dado como:

$$\hat{r} = \frac{\mathbf{r}}{r}, \quad (2.2.28)$$

portanto podemos reescrevê-lo da seguinte forma:

$$\hat{r} = \hat{x} \cos(\theta) + \hat{y} \sin(\theta).$$

O vetor velocidade, dado por:

$$\mathbf{v} = \frac{d\mathbf{r}}{dt},$$

na qual substituindo a expressão (2.2.28) nos leva a seguinte relação para a velocidade:

$$\begin{aligned} \mathbf{v} &= \frac{d(r\hat{r})}{dt}, \\ \mathbf{v} &= \hat{r} \frac{dr}{dt} + r \frac{d\hat{r}}{dt}. \end{aligned} \quad (2.2.29)$$

A derivada temporal de  $\hat{r}$  é

$$\begin{aligned} \frac{d\hat{r}}{dt} &= \hat{x} \frac{d \cos(\theta)}{dt} + \hat{y} \frac{d \sin(\theta)}{dt}, \\ &= -\hat{x} \sin(\theta) \frac{d\theta}{dt} + \hat{y} \cos(\theta) \frac{d\theta}{dt}, \\ &= \frac{d\theta}{dt} \underbrace{(-\hat{x} \sin(\theta) + \hat{y} \cos(\theta))}_{=\hat{\theta}}, \\ &= \frac{d\theta}{dt} \hat{\theta}. \end{aligned}$$

Assim, a velocidade da partícula em função de  $r$  e  $\theta$  pode ser escrita como:

$$\mathbf{v} = \hat{r} \frac{dr}{dt} + r \frac{d\hat{r}}{dt} = \hat{r} \frac{dr}{dt} + r \frac{d\theta}{dt} \hat{\theta},$$

ou em termos de  $v_x$  e  $v_y$ , da equação  $\mathbf{v} = \hat{x}(dx/dt) + \hat{y}(dy/dt)$ , temos:

$$\begin{cases} v_x = \frac{dr}{dt} \cos(\theta) - r \sin(\theta) \frac{d\theta}{dt}, \\ v_y = \frac{dr}{dt} \sin(\theta) + r \cos(\theta) \frac{d\theta}{dt}. \end{cases} \quad (2.2.30)$$

$$\begin{cases} v_x = \frac{dr}{dt} \cos(\theta) - r \sin(\theta) \frac{d\theta}{dt}, \\ v_y = \frac{dr}{dt} \sin(\theta) + r \cos(\theta) \frac{d\theta}{dt}. \end{cases} \quad (2.2.31)$$



Com as equações (2.2.30) e (2.2.31) podemos escrever as componentes das velocidades dos vetores centro de massa e relativo em função de  $r$  e  $\theta$ , portanto temos:

- Velocidade do centro de massa:

$$\begin{cases} V_x = \frac{dr_{cm}}{dt} \cos(\theta) - r_{cm} \frac{d\theta}{dt} \sin(\theta), \\ V_y = \frac{dr_{cm}}{dt} \sin(\theta) + r_{cm} \frac{d\theta}{dt} \cos(\theta), \end{cases} \quad (2.2.32)$$

$$\begin{cases} V_x = \underbrace{\frac{dr_{cm}}{dt}}_{V_r} \underbrace{\sin(\theta)}_{eY} + \underbrace{r_{cm} \frac{d\theta}{dt}}_{V_\theta} \underbrace{\cos(\theta)}_{eX}, \\ V_y = \underbrace{\frac{dr_{cm}}{dt}}_{V_r} \underbrace{\cos(\theta)}_{eX} - \underbrace{r_{cm} \frac{d\theta}{dt}}_{V_\theta} \underbrace{\sin(\theta)}_{eY}, \end{cases} \quad (2.2.33)$$

$$\begin{cases} V_x = V_r eX - V_\theta eY, \\ V_y = V_r eY + V_\theta eX. \end{cases} \quad (2.2.34)$$

$$\begin{cases} V_x = V_r eX - V_\theta eY, \\ V_y = V_r eY + V_\theta eX. \end{cases} \quad (2.2.35)$$

- Velocidade relativa:

$$\begin{cases} v_x = \frac{dr}{dt} \cos(\theta) - r \frac{d\theta}{dt} \sin(\theta), \\ v_y = \frac{dr}{dt} \sin(\theta) + r \frac{d\theta}{dt} \cos(\theta), \end{cases} \quad (2.2.36)$$

$$\begin{cases} v_x = \underbrace{\frac{dr}{dt}}_{v_r} \underbrace{\sin(\theta)}_{eY} + \underbrace{r \frac{d\theta}{dt}}_{v_\theta} \underbrace{\cos(\theta)}_{eX}, \\ v_y = \underbrace{\frac{dr}{dt}}_{v_r} \underbrace{\cos(\theta)}_{eX} - \underbrace{r \frac{d\theta}{dt}}_{v_\theta} \underbrace{\sin(\theta)}_{eY}, \end{cases} \quad (2.2.37)$$

o termo  $d\theta/dt$  pode ser substituído por:

$$\frac{d\theta}{dt} = \frac{\eta + 1}{\eta} \frac{l}{r^2}, \quad (2.2.38)$$

em que  $l$  é o momento angular relativo que será demonstrado na seção (3.1) do apêndice (C), sendo assim, temos:

$$\begin{cases} v_x = v_r eX - \frac{eY(\eta + 1)l}{\eta r}, \\ v_y = v_r eY + \frac{eX(\eta + 1)l}{\eta r}. \end{cases} \quad (2.2.39)$$

$$\begin{cases} v_x = v_r eX - \frac{eY(\eta + 1)l}{\eta r}, \\ v_y = v_r eY + \frac{eX(\eta + 1)l}{\eta r}. \end{cases} \quad (2.2.40)$$

Calculando a derivada temporal do versor  $\hat{\theta}$ , temos:

$$\frac{d\hat{\theta}}{dt} = -\hat{x} \cos(\theta) \frac{d\theta}{dt} - \hat{y} \sin(\theta) \frac{d\theta}{dt},$$

$$\frac{d\hat{\theta}}{dt} = (-\hat{x} \cos(\theta) - \hat{y} \sin(\theta)) \frac{d\theta}{dt},$$

$$\frac{d\hat{\theta}}{dt} = -\hat{r} \frac{d\theta}{dt}.$$

Realizado as derivadas de segunda ordem no tempo dos versores  $\hat{r}$  e  $\hat{\theta}$  chegamos as seguintes expressões:

$$\begin{aligned} \frac{d^2 \hat{r}}{dt^2} &= \frac{d}{dt} \left( \frac{d\theta}{dt} \hat{\theta} \right), \\ &= \frac{d^2 \theta}{dt^2} \hat{\theta} + \frac{d\theta}{dt} \frac{d\hat{\theta}}{dt}, \\ &= \frac{d^2 \theta}{dt^2} \hat{\theta} + \frac{d\theta}{dt} \left( -\hat{r} \frac{d\theta}{dt} \right), \\ &= \hat{\theta} \frac{d^2 \theta}{dt^2} - \hat{r} \left( \frac{d\theta}{dt} \right)^2, \end{aligned}$$

$$\begin{aligned}
\frac{d^2\hat{\theta}}{dt^2} &= \frac{d}{dt} \left( \frac{d\hat{\theta}}{dt} \right) = \frac{d}{dt} \left( -\hat{r} \frac{d\theta}{dt} \right), \\
&= -\hat{r} \frac{d^2\theta}{dt^2} - \frac{d\theta}{dt} \frac{d\hat{r}}{dt}, \\
&= -\hat{r} \frac{d^2\theta}{dt^2} - \frac{d\theta}{dt} \frac{d\theta}{dt} \hat{\theta}, \\
&= -\hat{r} \frac{d^2\theta}{dt^2} - \hat{\theta} \left( \frac{d\theta}{dt} \right)^2,
\end{aligned}$$

que possibilita escrever a aceleração na coordenada polar da seguinte maneira:

$$\begin{aligned}
\mathbf{a} &= \frac{d^2\mathbf{r}}{dt^2} = \frac{d}{dt} \left( \frac{d\mathbf{r}}{dt} \right), \\
&= \frac{d}{dt} \left( \hat{r} \frac{dr}{dt} + r \frac{d\theta}{dt} \hat{\theta} \right), \\
&= \frac{d\hat{r}}{dt} \frac{dr}{dt} + \hat{r} \frac{d^2r}{dt^2} + \frac{d}{dt} \left( r \frac{d\theta}{dt} \right) \hat{\theta} + r \frac{d\theta}{dt} \frac{d\hat{\theta}}{dt}, \\
&= \hat{\theta} \frac{d\theta}{dt} \frac{dr}{dt} + \hat{r} \frac{d^2r}{dt^2} + \frac{dr}{dt} \frac{d\theta}{dt} \hat{\theta} + r \hat{\theta} \frac{d^2\theta}{dt^2} + r \frac{d\theta}{dt} \left( -\hat{r} \frac{d\theta}{dt} \right), \\
&= \hat{r} \left( \frac{d^2r}{dt^2} - r \left( \frac{d\theta}{dt} \right)^2 \right) + \hat{\theta} \left( 2 \frac{d\theta}{dt} \frac{dr}{dt} + r \frac{d^2\theta}{dt^2} \right), \\
\mathbf{a} &= \hat{r} \left( \frac{d^2r}{dt^2} - r \left( \frac{d\theta}{dt} \right)^2 \right) + \hat{\theta} \left( 2 \frac{d\theta}{dt} \frac{dr}{dt} + r \frac{d^2\theta}{dt^2} \right). \tag{2.2.41}
\end{aligned}$$

Dado os versores  $\hat{r}$  e  $\hat{\theta}$  em função dos versores  $\hat{x}$  e  $\hat{y}$ ,

$$\begin{cases} \hat{r} = \cos(\theta)\hat{x} + \sin(\theta)\hat{y} \\ \hat{\theta} = -\sin(\theta)\hat{x} + \cos(\theta)\hat{y} \end{cases}$$

resolvemos o sistema acima para termos dos versores  $\hat{x}$  e  $\hat{y}$  em função dos versores  $\hat{r}$  e  $\hat{\theta}$ , ou seja,

$$\begin{cases} \hat{x} = \cos(\theta)\hat{r} - \sin(\theta)\hat{\theta} \\ \hat{y} = \sin(\theta)\hat{r} + \cos(\theta)\hat{\theta} \end{cases}$$

Escrevendo a aceleração como:

$$\mathbf{a} = a_x\hat{x} + a_y\hat{y},$$

e substituímos os termos  $\hat{x}$  e  $\hat{y}$  temos:

$$\mathbf{a} = a_x(-\sin(\theta)\hat{\theta} + \cos(\theta)\hat{r}) + a_y(\cos(\theta)\hat{\theta} + \sin(\theta)\hat{r}), \tag{2.2.42}$$

em que os termos  $a_x$  e  $a_y$  são as componentes das acelerações no plano cartesiano  $x$  e  $y$  respectivamente.

Para obter as equações que descrevem o movimento das partículas com relação aos vetores de centro de massa e relativo nas coordenadas polares substituímos nas equações (1.0.70), (1.0.71), (1.0.72) e (1.0.73), do apêndice (A), primeiramente as equações (2.1.19), (2.1.20), (2.1.22) e (2.1.23), da seção (2.1)(Apêndice (B)), que descrevem as velocidades das partículas nas coordenadas cartesianas em função da velocidade do centro

de massa e da velocidade relativa. Posteriormente substituímos nestas equações as expressões (2.2.34), (2.2.35), (2.2.39) e (2.2.40) que descrevem as velocidades do centro de massa e relativo em função de  $V_r$ ,  $V_\theta$ ,  $v_r$  e  $l$ .

Igualando as acelerações dada pelas equações (2.2.41) e (2.2.42) e substituindo primeiramente os termos  $a_x$  e  $a_y$  pelas acelerações relativas, (2.1.16) e (2.1.17), obtemos as equações:

$$\begin{aligned} \frac{dv_r}{dt} = & \frac{(\eta + 1)^2 l^2}{\eta^2 r^3} - \frac{\eta + 1}{\eta r^2} + \frac{8\alpha v_r}{(\eta + 1)r^2} \\ & - \frac{4\alpha(\eta - 1)V_r}{\eta r^2} - \frac{2v_r^2}{r^2(\eta + 1)} - \frac{2(\eta + 1)V_r^2}{\eta r^2} \\ & + \left( \frac{4\alpha^2(\eta - 1)^2(\eta r + \eta + 1)}{\eta^3 r^4(\eta r^2 - 1)} + \frac{8\alpha^2 - 2}{\eta^2 r^5} \right) l^2 \\ & + \left( \frac{8\alpha^2(\eta r + r + 2)}{r(\eta r^2 - 1)} - \frac{2(\eta r + r + 1)(4\alpha^2 - 1)}{\eta r^3} \right) V_\theta^2 \\ & + \left( \frac{4\alpha^2(\eta - 1)(\eta^2 r^2 + 2 + (r^2 + 4r + 2)\eta)}{\eta^2 r^3(\eta r^2 - 1)} - \frac{(\eta - 1)(\eta r + r + 2)(4\alpha^2 - 1)}{\eta^2 r^4} \right) lV_\theta, \end{aligned} \quad (2.2.43)$$

$$\begin{aligned} \frac{dl}{dt} = & \frac{2(\eta - 1)(\eta r^2 + 1)\alpha V_\theta}{r(\eta + 1)(\eta r^2 - 1)} - \frac{2(2\eta^2 r^2 - \eta^2 - 1)\alpha l}{\eta r^2(\eta r^2 - 1)(\eta + 1)} \\ & + \left( \frac{2(\eta - 1)(4\alpha^2 - 1)}{\eta r^2(\eta r^2 - 1)} - \frac{4(\eta - 1)\alpha^2(-1 + (r - 1)\eta)}{(\eta r^2(\eta r^2 - 1)(\eta + 1))} \right) lV_r \\ & + \left( \frac{8\alpha^2(1/2 + \eta^2 r^2 + (r^2 - r + 1/2)\eta)}{r(\eta + 1)(\eta r^2 - 1)} + \frac{-8\alpha^2 + 2}{r} \right) V_r V_\theta \\ & + \frac{4\alpha^2}{\eta r^2(\eta r^2 - 1)^2(\eta + 1)^2} (1 + (4r^4 - r^3 - 4r^2)\eta^4 + (4r^4 - 3r^2 + r + 1)\eta^3 \\ & + (2 - r^3 - 3r^2)\eta^2 + (2 - 4r^2 + r)\eta) lv_r \\ & - \frac{(4\eta^2 r^2 - 3\eta^2 + 2\eta - 3)(4\alpha^2 - 1)lv_r}{\eta r^2(\eta r^2 - 1)(\eta + 1)} \\ & + \left( \frac{(\eta r^2 + 3)(\eta - 1)(4\alpha^2 - 1)}{r(\eta + 1)(\eta r^2 - 1)} \right. \\ & \left. - \frac{4\alpha^2(\eta - 1)(\eta^3 r^4 - 1 + r^2(r^2 + r + 6)\eta^2 + (6r^2 - r - 1)\eta)}{r(\eta r^2 - 1)^2(\eta + 1)^2} \right) v_r V_\theta, \end{aligned} \quad (2.2.44)$$

que descrevem a aceleração relativa radial e a variação do momento angular relativo. Substituindo  $a_x$  por (2.1.7) e  $a_y$  por (2.1.8) obtemos as equações:

$$\begin{aligned} \frac{dV_r}{dt} = & \frac{(\eta + 1)lV_\theta}{\eta r^2} - \frac{4\alpha(\eta - 1)v_r}{r^2(\eta + 1)^2} + \frac{2(\eta - 1)v_r^2}{r^2(\eta + 1)^2} - \frac{8\alpha V_r}{r^2(\eta + 1)} \\ & + \left( \frac{(\eta - 1)(\eta r + r - 1)(4\alpha^2 - 1)}{\eta^2 r^5(\eta + 1)} - \frac{4(\eta - 1)\alpha^2(\eta^2 r^2 - 2 + (r^2 - 2r - 2)\eta)}{r^4 \eta^2(\eta r^2 - 1)(\eta + 1)} \right) l^2 \\ & + \left( \frac{(2\eta^2 r + \eta^2 + 2\eta r - 2\eta + 1)(4\alpha^2 - 1)}{\eta^2 r^4(\eta + 1)} - \frac{4\alpha^2(\eta r + r - 4)(2\eta r + \eta + 1)}{r^3 \eta(\eta + 1)(\eta r^2 - 1)} \right) lV_\theta \\ & + \left( \frac{4\alpha^2(\eta - 1) - \eta + 1}{\eta r^3(\eta + 1)} - \frac{8\alpha^2(\eta - 1)}{r(\eta + 1)(\eta r^2 - 1)} \right) V_\theta^2 + \left( \frac{16\alpha^2}{r^2(\eta + 1)} + \frac{-16\alpha^2 + 4}{r^2(\eta + 1)} \right) v_r V_r, \end{aligned} \quad (2.2.45)$$

$$\begin{aligned}
\frac{dV_\theta}{dt} = & \frac{2(r+1)(\eta-1)(\eta r+1)\alpha l}{\eta r^3(\eta+1)(\eta r^2-1)} - \frac{(\eta+1)lV_r}{\eta r^2} \\
& + \frac{4(r+1)(\eta r+1)\alpha V_\theta}{r^2(\eta+1)(\eta r^2-1)} + \frac{4\alpha^2(\eta-1)V_r V_\theta}{r(\eta+1)(\eta r^2-1)} \\
& + \left( \frac{2(r+1)(\eta r+1)(4\alpha^2-1)}{\eta r^3(\eta r^2-1)} - \frac{4\alpha^2(-1+2\eta^2 r^2+(2r^2+2r-1)\eta)}{\eta r^3(\eta+1)(\eta r^2-1)} \right) lV_r \\
& + \frac{3(r+1)(\eta-1)(\eta r+1)(4\alpha^2-1)lv_r}{\eta r^3(\eta+1)(\eta r^2-1)} \\
& - \frac{4\alpha^2(\eta-1)}{\eta r^3(\eta r^2-1)^2(\eta+1)^2} (-1+(3r^4+3r^3)\eta^3 \\
& + (3r^4+7r^3+4r^2)\eta^2 + (3r^3+4r^2-r-1)\eta) lv_r \\
& + \frac{4(r+1)(\eta r+1)(4\alpha^2-1)v_r V_\theta}{r^2(\eta+1)(\eta r^2-1)} \\
& - \frac{4\alpha^2}{r^2(\eta r^2-1)^2(\eta+1)^2} (-3+(4r^4+6r^3)\eta^3 \\
& + (4r^4+14r^3+11r^2)\eta^2 + (6r^3+11r^2-2r-3)\eta) v_r V_\theta. \tag{2.2.46}
\end{aligned}$$

que descrevem a aceleração do centro de massa radial e a aceleração do centro de massa tangencial respectivamente.

# Apêndice C

## Transformação De Escala

### 3.1 Momento angular

Da definição do momento angular para um sistema de duas partículas temos:

$$\mathbf{L} = m_1 \mathbf{r}_1 \times \mathbf{v}_1 + m_2 \mathbf{r}_2 \times \mathbf{v}_2, \quad (3.1.1)$$

onde os vetores posição  $r_1$  e  $r_2$  e os vetores velocidades  $v_1$  e  $v_2$  são substituídos pelos vetores do centro de massa e relativo tanto nas posições quanto nas velocidades de forma que:

$$\begin{aligned} \mathbf{L} &= m_1 \left( \mathbf{R} - \frac{m_2}{m_1 + m_2} \mathbf{r} \right) \times \left( \mathbf{V} - \frac{m_2}{m_1 + m_2} \mathbf{v} \right) \\ &\quad + m_2 \left( \mathbf{R} + \frac{m_1}{m_1 + m_2} \mathbf{r} \right) \times \left( \mathbf{V} + \frac{m_1}{m_1 + m_2} \mathbf{v} \right), \\ \mathbf{L} &= (m_1 + m_2) \mathbf{R} \times \mathbf{V} + \frac{m_1 m_2}{m_1 + m_2} \mathbf{r} \times \mathbf{v}. \end{aligned}$$

Aplicando as transformações de escalas:

$$\begin{aligned} r &\rightarrow Lr, \\ v &\rightarrow cv, \\ V &\rightarrow cV, \end{aligned}$$

no momento angular temos a seguinte expressão:

$$\begin{aligned} &(m_1 + m_2) L \mathbf{R} \times c \mathbf{V} + \frac{m_1 m_2}{m_1 + m_2} L \mathbf{r} \times c \mathbf{v}, \\ &\frac{|q_1| |q_2|}{4\pi \epsilon_0 m_2 c^2} c (m_1 + m_2) \mathbf{R} \times \mathbf{V} + \frac{|q_1| |q_2|}{4\pi \epsilon_0 m_2 c^2} c \frac{m_1 m_2}{m_1 + m_2} \mathbf{r} \times \mathbf{v}, \\ &\frac{|q_1| |q_2|}{4\pi \epsilon_0 c} (\eta + 1) \mathbf{R} \times \mathbf{V} + \frac{|q_1| |q_2|}{4\pi \epsilon_0 c} \frac{\eta}{\eta + 1} \mathbf{r} \times \mathbf{v}, \\ &\frac{|q_1| |q_2|}{4\pi \epsilon_0 c} \left( (\eta + 1) \mathbf{R} \times \mathbf{V} + \frac{\eta}{\eta + 1} \mathbf{r} \times \mathbf{v} \right), \end{aligned}$$

que nos leva a seguinte relação de transformação:

$$\mathbf{L} \rightarrow \frac{|q_1| |q_2|}{4\pi \epsilon_0 c} \mathbf{L}. \quad (3.1.2)$$

Da relação acima temos que o momento angular nas unidades adimensionais é dado por:

$$\mathbf{L} = (\eta + 1) \mathbf{R} \times \mathbf{V} + \frac{\eta}{\eta + 1} \mathbf{r} \times \mathbf{v}. \quad (3.1.3)$$

Se considerarmos o momento angular na coordenada polar com o centro de massa nulo e que as partículas encontram-se no plano  $xy$  obtemos a seguinte equação:

$$\begin{aligned} \mathbf{L} &= \frac{\eta}{\eta + 1} \mathbf{r} \times \mathbf{v}, \\ \mathbf{L} &= \frac{\eta}{\eta + 1} (r\hat{r}) \times \left( \hat{r} \frac{dr}{dt} + r \frac{d\hat{\theta}}{dt} \right), \\ \mathbf{L} &= \frac{\eta}{\eta + 1} r^2 \frac{d\theta}{dt} \underbrace{(\hat{r} \times \hat{\theta})}_{\hat{z}}. \end{aligned}$$

Como a projeção do momento é no eixo  $z$  podemos reescrever a expressão acima da seguinte maneira:

$$l_z = l = \frac{\eta}{\eta + 1} r^2 \frac{d\theta}{dt}. \quad (3.1.4)$$

Portanto, temos uma relação entre a velocidade angular e o momento angular dada por:

$$r \frac{d\theta}{dt} = \frac{\eta + 1}{\eta} \frac{l}{r}. \quad (3.1.5)$$

# Apêndice D

## Condições Iniciais

### 4.1 Condições iniciais para o sistema dinâmico

Para determinar as condições iniciais do conjunto de equações (1.0.70), (1.0.71), (1.0.72) e (1.0.73), do apêndice (A), no plano cartesiano iremos considerar  $S = -1$ , pois o sistema de interesse é o atrativo. Estabelecemos a distância entre as partículas no tempo inicial,  $t_0$ , como  $r_0 = 18789.0$  (Raio de Bohr para a órbita  $n = 1$ ). Para determinar a velocidade angular inicial consideramos a equação (2.2.43) do apêndice (B) com o centro de massa em repouso, portanto  $V_r = V_\theta = 0$ . Assumimos  $\alpha = 0$  e  $v_r = 0$ , isto é, a força que atua no sistema tem caráter puramente coulombiana (gerando órbitas circulares), portanto:

$$\begin{aligned} 0 &= -\frac{\eta + 1}{\eta r^2} + \frac{(\eta + 1)^2 l^2}{\eta^2 r^3} - \frac{2l^2}{\eta^2 r^5}, \\ \frac{\eta + 1}{\eta r^2} &= \frac{(\eta + 1)^2 l^2}{\eta^2 r^3} - \frac{2l^2}{\eta^2 r^5}, \end{aligned}$$

como os termos que multiplicam  $l^2$  a direita da igualdade acima possuem valores aproximados de  $(\eta + 1)^2 / \eta^2 r^3 \approx 10^{-13}$  e  $2 / \eta^2 r^5 \approx 10^{-28}$  considerando  $r = r_0$  e  $\eta = 1836.0$  (razão entre a massa do próton e a massa do elétron) podemos desconsiderar o segundo termo em detrimento do primeiro que apresenta uma maior contribuição ao momento angular relativo, sendo assim,

$$\begin{aligned} 1 &= \frac{(\eta + 1)l^2}{\eta r}, \\ \frac{\eta r}{(\eta + 1)} &= l^2, \\ \frac{\eta r}{\eta + 1} &= \left(\frac{\eta}{\eta + 1}\right)^2 r^4 \left(\frac{d\theta}{dt}\right)^2, \\ \left(\frac{\eta + 1}{\eta}\right) \frac{1}{r^3} &= \left(\frac{d\theta}{dt}\right)^2. \end{aligned}$$

A velocidade angular inicial, portanto, é dado da seguinte maneira:

$$\left.\frac{d\theta}{dt}\right|_{t=0} = \left(\frac{\eta + 1}{\eta}\right)^{\frac{1}{2}} \frac{1}{r_0^{\frac{3}{2}}}. \quad (4.1.1)$$

Caso tivéssemos adotado alfa igual a  $1/2$  ou  $-1/2$  chegaríamos ao mesmo resultado para a velocidade angular inicial com a diferença que para alfa diferente de zero o segundo

termo que multiplica  $l^2$  dado por  $2/\eta^2 r^5$  seria substituído pela expressão:

$$\frac{(\eta - 1)^2 (\eta r + \eta + 1)}{r^4 (\eta r^2 - 1) \eta^3},$$

cujos valores aproximados são da ordem de  $10^{-25}$  dados os mesmos valores para  $r$  e  $\eta$ , sendo assim, este termo também seria desprezado devido a sua pequena contribuição para o momento angular relativo.

Com o vetor da posição do centro de massa (Seção(2.1) do apêndice (B)) :

$$\mathbf{r}_{cm} = \frac{\eta \mathbf{r}_1 + \mathbf{r}_2}{\eta + 1}, \quad (4.1.2)$$

e o vetor da posição relativa (Seção(2.1) do apêndice (B)):

$$\mathbf{r} = \mathbf{r}_2 - \mathbf{r}_1, \quad (4.1.3)$$

podemos expressar os vetores  $\mathbf{r}_1$  e  $\mathbf{r}_2$  em função do vetor centro de massa e relativo, portanto temos:

$$\begin{aligned} (\eta + 1)\mathbf{r}_{cm} &= \eta \mathbf{r}_1 + \mathbf{r}_2 + \mathbf{r}_1 - \mathbf{r}_1, \\ (\eta + 1)\mathbf{r}_{cm} &= (\eta + 1)\mathbf{r}_1 + \mathbf{r}, \end{aligned} \quad (4.1.4)$$

$$\begin{aligned} (\eta + 1)\mathbf{r}_{cm} &= \eta \mathbf{r}_1 + \mathbf{r}_2 + \eta \mathbf{r}_2 - \eta \mathbf{r}_2, \\ (\eta + 1)\mathbf{r}_{cm} &= (\eta + 1)\mathbf{r}_2 - \eta \mathbf{r}. \end{aligned} \quad (4.1.5)$$

Assumindo para o tempo inicial que  $\mathbf{r}_{cm} = 0$ , temos as seguintes relações:

$$\mathbf{r}_1 = -\frac{\mathbf{r}}{\eta + 1}, \quad (4.1.6)$$

$$\mathbf{r}_2 = \frac{\eta}{\eta + 1}\mathbf{r}, \quad (4.1.7)$$

e dado que  $\mathbf{r} = (r_0, 0)$  as posições iniciais no plano cartesiano são:

$$x_1(t_0) = -\frac{r_0}{\eta + 1}, \quad (4.1.8)$$

$$y_1(t_0) = 0, \quad (4.1.9)$$

$$x_2(t_0) = \frac{\eta r_0}{\eta + 1}, \quad (4.1.10)$$

$$y_2(t_0) = 0. \quad (4.1.11)$$

Para determinar as velocidades iniciais consideremos as equações (2.2.25), (2.2.26), (2.2.39) e (2.2.40) (Seção (2.2) do apêndice (B)) reescrevendo da seguinte maneira:

$$v_x = \frac{d\mathbf{r}}{dt} \frac{x}{r} - y \frac{d\theta}{dt}, \quad (4.1.12)$$

$$v_y = \frac{d\mathbf{r}}{dt} \frac{y}{r} + x \frac{d\theta}{dt}. \quad (4.1.13)$$

Para o tempo inicial, em que  $d\mathbf{r}/dt = 0$ , temos:

$$v_x = -y \frac{d\theta}{dt}, \quad (4.1.14)$$

$$v_y = x \frac{d\theta}{dt}. \quad (4.1.15)$$



Portanto, as condições iniciais para as velocidades são:

$$V_{1x}(t_0) = -y_1 \left. \frac{d\theta}{dt} \right|_{t=0} = 0,$$

$$V_{1y}(t_0) = x_1 \left. \frac{d\theta}{dt} \right|_{t=0},$$

$$V_{2x}(t_0) = -y_2 \left. \frac{d\theta}{dt} \right|_{t=0} = 0,$$

$$V_{2y}(t_0) = x_2 \left. \frac{d\theta}{dt} \right|_{t=0}.$$

# Apêndice E

## Código Fonte

Abaixo temos os parâmetros utilizados em algumas das simulações apresentadas no capítulo III.

$\eta$	$\alpha$	$H_n$	$r_0$	$n$	$m$	Tempo de simulação
1	1/2	$H^0$	50	95	1	00 : 00 : 01
1	1/2	$H^1$	50	60	25000	00 : 00 : 16
1	1/2	$H^2$	50	60	25000	00 : 01 : 17
1	1/2	$H^3$	50	60	25000	00 : 10 : 59
1	0	$H^0$	50	100	250	00 : 00 : 05
1	0	$H^1$	50	100	250	00 : 00 : 48
2	1/2	$H^0$	50	95	1	00 : 00 : 01
2	1/2	$H^1$	50	60	25000	00 : 00 : 28
2	1/2	$H^2$	50	60	25000	00 : 02 : 14
2	1/2	$H^3$	50	60	25000	00 : 17 : 54
2	0	$H^0$	50	100	4500	00 : 01 : 44
2	0	$H^1$	50	100	4500	00 : 14 : 21
100	1/2	$H^0$	50	95	45	00 : 00 : 06
100	1/2	$H^1$	50	60	25000	00 : 16 : 29
100	1/2	$H^2$	50	60	25000	01 : 23 : 06
100	1/2	$H^3$	50	60	25000	10 : 50 : 29
100	0	$H^0$	50	160	11500	00 : 06 : 30
100	0	$H^1$	50	160	11500	00 : 59 : 17

Tabela E.1: Tabela dos parâmetros das simulações. O parâmetro  $\alpha$  indica a proporção entre os campos calculados no tempo de retardo e no tempo avançado. Dado o valor alfa igual a meio o campo é calculado somente no tempo de retardo e quando alfa é zero existe a contribuição dos dois campos de forma idêntica. O valor de  $\eta$  indica a razão entre as massas das partículas e  $r_0$  a distância inicial entre as partículas. O campo vetorial  $H^n$  indica em qual ordem de convergência  $n$  estamos simulando as equações. Os parâmetros  $n$  e  $m$ , trabalham juntos para obter os valores das posições, velocidades e acelerações calculados durante as simulações. Sendo assim,  $n$  indica a divisão do círculo em  $n$  partes e  $m$  é o número de círculos que desejamos calcular, por fim temos os tempos de simulação.

No Pseudocódigo, apresentado abaixo, temos a estrutura principal de como se deve realizar o processo de simulação do sistema. O que muda de uma simulação para outra é a ordem do campo  $H^n$  que estamos avaliando.

Como exemplo quando estamos considerando a aproximação de ordem zero,  $H^0$ , aplicamos o modelo de runge-kutta tradicional conhecido na literatura. Ou seja, aplicamos o modelo de runge-kutta nas equações instâneas, desenvolvidas no apêndice (A), obtemos as posições e velocidades das partículas. Quando realizamos a aproximação de primeira ordem,  $H^1$ , calculamos os tempos de retardo e/ou os tempos avançados de maneira a encontrar as posições e velocidades, utilizando o runge-kutta desenvolvido para aproximação de ordem zero, das posições e velocidades dos tempos de retardo e/ou dos tempos avançados. Assim, podemos calcular as acelerações retardadas e/ou avançadas para obter as acelerações no tempo  $t$  e assim aplicar o método de runge-kutta nessas acelerações. Desta forma, podemos aumentar o valor de  $n$  do campo vetorial  $H^n$  para encontrar a aproximação do campo  $H^{n+1}$  considerando o runge-kutta desenvolvido para o campo  $H^n$ .

```

1 /* Loop pra os m circulos */
2 for (j = 1; j <= m; j++):
3     /* Loop para o cálculo do Runge Kutta e
4         retornar o passo do círculo dividido em
5         n partes */
6     for (i = 1; i < n; i++):
7
8         t += dt
9
10        dt_atual, x1, y1, x2, y2, V1x, V1y, V2x, V2y <- Runge-Kutta do campo  $H^n$ 
11
12        dt = dt_atual
13
14        Salva o tempo, posições e velocidades <- t x1 y1 x2 y2 V1x V1y V2x V2y
15
16        /* Condição de parada do programa*/
17        if (sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (
18            sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)) < 2
19        ):
20            break
21
22        /* Atualização do passo para o Próximo ciclos*/
23        vx = V2x - V1x
24        vy = V2y - V1y
25        rr = np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
26        ex = (x2 - x1) / (rr)
27        ey = (y2 - y1) / (rr)
28        DT = (-vx * ey + vy * ex) / (rr)
29        dt = (2.0 * np.pi) / (DT * n)
30
31        /* Condição de parada do programa*/
32        if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (rr < 2):
33            break

```

Código E.1: Pseudocódigo

Por fim, temos os códigos, escrito em Python, para a solução numérica apresentada no capítulo III.

```

1 # -*- coding: utf-8 -*-
2 """
3 @author: rodrigo
4 """
5
6 # Bibliotecas
7
8 import numpy as np # Biblioteca com funções matemáticas
9 from tqdm import tqdm # Biblioteca para calcular o tempo no loop
10
11 # ### Funções ###
12
13 # ### Equações com os tempos de retardo ###
14
15 # Aceleração da partícula 1 no eixo x para o tempo retardado
16 def f_alx_tr(S, eta, x1, y1, X2, Y2, v1x, v1y, V2x, V2y, A2x, A2y):
17
18     r12 = np.sqrt((-x1 + X2) ** 2 + (-y1 + Y2) ** 2)
19     e2X = (-x1 + X2) / r12
20     e2Y = (-y1 + Y2) / r12
21
22     fflx = (
23         -(v1x ** 2) - v1y ** 2 + 1) ** 2
24         * (
25             -(v1x ** 2) + 1)
26             * (
27                 -S
28                 * (V2y + e2Y)
29                 * (v1y + e2Y)
30                 * A2x
31                 / (r12 * (V2x * e2X + V2y * e2Y + 1) ** 3)
32                 + S
33                 * (V2x + e2X)
34                 * (v1y + e2Y)
35                 * A2y
36                 / (r12 * (V2x * e2X + V2y * e2Y + 1) ** 3)
37                 + S
38                 * (V2x ** 2 + V2y ** 2 - 1)
39                 * (V2x * e2Y * v1y - V2y * e2X * v1y + V2x + e2X)
40                 / (r12 ** 2 * (V2x * e2X + V2y * e2Y + 1) ** 3)
41             )
42         - v1x
43         * v1y
44         * (
45             S
46             * (V2y + e2Y)
47             * (v1x + e2X)
48             * A2x
49             / (r12 * (V2x * e2X + V2y * e2Y + 1) ** 3)
50             - S
51             * (V2x + e2X)
52             * (v1x + e2X)
53             * A2y
54             / (r12 * (V2x * e2X + V2y * e2Y + 1) ** 3)
55             - S
56             * (V2x ** 2 + V2y ** 2 - 1)
57             * (V2x * e2Y * v1x - V2y * e2X * v1x - V2y - e2Y)
58             / (r12 ** 2 * (V2x * e2X + V2y * e2Y + 1) ** 3)

```

```

59     )
60     )
61     / eta
62 )
63 return ff1x
64
65
66 # Aceleração da partícula 1 no eixo y para o tempo retardado
67 def f_a1y_tr(S, eta, x1, y1, X2, Y2, v1x, v1y, V2x, V2y, A2x, A2y):
68
69     r12 = np.sqrt((-x1 + X2) ** 2 + (-y1 + Y2) ** 2)
70     e2X = (-x1 + X2) / r12
71     e2Y = (-y1 + Y2) / r12
72
73     ff1y = (
74         -(v1x ** 2) - v1y ** 2 + 1) ** 2
75         * (
76             -v1x
77             * v1y
78             * (
79                 -S
80                 * (V2y + e2Y)
81                 * (v1y + e2Y)
82                 * A2x
83                 / (r12 * (V2x * e2X + V2y * e2Y + 1) ** 3)
84                 + S
85                 * (V2x + e2X)
86                 * (v1y + e2Y)
87                 * A2y
88                 / (r12 * (V2x * e2X + V2y * e2Y + 1) ** 3)
89                 + S
90                 * (V2x ** 2 + V2y ** 2 - 1)
91                 * (V2x * e2Y * v1y - V2y * e2X * v1y + V2x + e2X)
92                 / (r12 ** 2 * (V2x * e2X + V2y * e2Y + 1) ** 3)
93             )
94         + (-(v1y ** 2) + 1)
95         * (
96             S
97             * (V2y + e2Y)
98             * (v1x + e2X)
99             * A2x
100            / (r12 * (V2x * e2X + V2y * e2Y + 1) ** 3)
101            - S
102            * (V2x + e2X)
103            * (v1x + e2X)
104            * A2y
105            / (r12 * (V2x * e2X + V2y * e2Y + 1) ** 3)
106            - S
107            * (V2x ** 2 + V2y ** 2 - 1)
108            * (V2x * e2Y * v1x - V2y * e2X * v1x - V2y - e2Y)
109            / (r12 ** 2 * (V2x * e2X + V2y * e2Y + 1) ** 3)
110        )
111     )
112     / eta
113 )
114
115 return ff1y
116
117
118 # Aceleração da partícula 2 no eixo x para o tempo retardado

```

```

119 def f_a2x_tr(S, X1, Y1, x2, y2, V1x, V1y, v2x, v2y, A1x, A1y):
120
121     r21 = np.sqrt((X1 - x2) ** 2 + (Y1 - y2) ** 2)
122     e1X = (-X1 + x2) / r21
123     e1Y = (-Y1 + y2) / r21
124
125     ff2x = ((-v2x ** 2) - v2y ** 2 + 1) ** 2 * (
126         ((-v2x ** 2) + 1)
127         * (
128             -S
129             * (v2y - e1Y)
130             * (V1y - e1Y)
131             * A1x
132             / (r21 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
133             + S
134             * (v2y - e1Y)
135             * (V1x - e1X)
136             * A1y
137             / (r21 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
138             - S
139             * (V1x ** 2 + V1y ** 2 - 1)
140             * (V1x * e1Y * v2y - V1y * e1X * v2y - V1x + e1X)
141             / (r21 ** 2 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
142         )
143     - v2x
144     * v2y
145     * (
146         S
147         * (v2x - e1X)
148         * (V1y - e1Y)
149         * A1x
150         / (r21 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
151         - S
152         * (v2x - e1X)
153         * (V1x - e1X)
154         * A1y
155         / (r21 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
156         + S
157         * (V1x ** 2 + V1y ** 2 - 1)
158         * (V1x * e1Y * v2x - V1y * e1X * v2x + V1y - e1Y)
159         / (r21 ** 2 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
160     )
161 )
162
163 return ff2x
164
165 # Aceleração da partícula 2 no eixo y para o tempo retardado
166 def f_a2y_tr(S, X1, Y1, x2, y2, V1x, V1y, v2x, v2y, A1x, A1y):
167
168     r21 = np.sqrt((X1 - x2) ** 2 + (Y1 - y2) ** 2)
169     e1X = (-X1 + x2) / r21
170     e1Y = (-Y1 + y2) / r21
171
172     ff2y = ((-v2x ** 2) - v2y ** 2 + 1) ** 2 * (
173         -v2x
174         * v2y
175         * (
176             -S
177             * (v2y - e1Y)
178             * (V1y - e1Y)

```

```

179         * A1x
180         / (r21 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
181         + S
182         * (v2y - e1Y)
183         * (V1x - e1X)
184         * A1y
185         / (r21 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
186         - S
187         * (V1x ** 2 + V1y ** 2 - 1)
188         * (V1x * e1Y * v2y - V1y * e1X * v2y - V1x + e1X)
189         / (r21 ** 2 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
190     )
191 + (-(v2y ** 2) + 1)
192 * (
193     S
194     * (v2x - e1X)
195     * (V1y - e1Y)
196     * A1x
197     / (r21 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
198     - S
199     * (v2x - e1X)
200     * (V1x - e1X)
201     * A1y
202     / (r21 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
203     + S
204     * (V1x ** 2 + V1y ** 2 - 1)
205     * (V1x * e1Y * v2x - V1y * e1X * v2x + V1y - e1Y)
206     / (r21 ** 2 * (-V1x * e1X - V1y * e1Y + 1) ** 3)
207 )
208 )
209
210 return ff2y
211
212
213 # # Equações com os tempos de retardo com aproximação instantânea (tr -> t) #
214
215 # Aceleração da partícula 1 no eixo x para o tempo instantâneo
216 def d_V1x_inst(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
217
218     aix = (
219         np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
220         * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
221         * (
222             -(
223                 (
224                     (V2x)
225                     * (x2 - x1)
226                     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
227                     + (V2y)
228                     * (y2 - y1)
229                     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
230                     + 1
231                 )
232                 ** 3
233             )
234         * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
235         * eta
236         * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
237         * (
238             (V1x) * (V1y) * (V2y)

```

```

239         + (V1x)
240         * (V1y)
241         * (y2 - y1)
242         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
243         - (V1y) ** 2 * (V2x)
244         - (V1y) ** 2
245         * (x2 - x1)
246         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
247         - (V1y)
248         * (V2x)
249         * (y2 - y1)
250         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
251         + (V1y)
252         * (V2y)
253         * (x2 - x1)
254         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
255         - (V2x) * (-((V1x) ** 2) - (V1y) ** 2 + 1)
256         - (-((V1x) ** 2) - (V1y) ** 2 + 1)
257         * (x2 - x1)
258         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
259     )
260     * (
261         -(V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
262         - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
263         + 1
264     )
265     ** 3
266     - S
267     * eta
268     * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
269     * (-((V1x) ** 2) - (V1y) ** 2 + 1)
270     * (
271         (V1x)
272         * (V1y)
273         * (x2 - x1)
274         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
275         + (V1y) ** 2
276         * (y2 - y1)
277         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
278         + (-((V1x) ** 2) - (V1y) ** 2 + 1)
279         * (y2 - y1)
280         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
281         + V1y
282     )
283     * (
284         (V1x)
285         * (V2x)
286         * (V2y)
287         * (x2 - x1)
288         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
289         - (V1x)
290         * (V2x)
291         * (x2 - x1)
292         * (y2 - y1)
293         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
294         + (V1x)
295         * (V2y) ** 2
296         * (y2 - y1)
297         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
298         - (V1x)

```



```

299         * (V2y)
300         * (y2 - y1) ** 2
301         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
302         + 2
303         * (V1x)
304         * (-((V2x) ** 2) - (V2y) ** 2 + 1)
305         * (y2 - y1)
306         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
307         - (V1y)
308         * (V2x) ** 2
309         * (x2 - x1)
310         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
311         - (V1y)
312         * (V2x)
313         * (V2y)
314         * (y2 - y1)
315         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
316         + (V1y)
317         * (V2x)
318         * (x2 - x1) ** 2
319         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
320         + (V1y)
321         * (V2y)
322         * (x2 - x1)
323         * (y2 - y1)
324         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
325         - 2
326         * (V1y)
327         * (-((V2x) ** 2) - (V2y) ** 2 + 1)
328         * (x2 - x1)
329         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
330         + (V2x) ** 2
331         * (x2 - x1)
332         * (y2 - y1)
333         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
334         - (V2x)
335         * (V2y)
336         * (x2 - x1) ** 2
337         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
338         + (V2x)
339         * (V2y)
340         * (y2 - y1) ** 2
341         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
342         - (V2y) ** 2
343         * (x2 - x1)
344         * (y2 - y1)
345         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
346         + (V1x) * (V2y)
347         - (V1x) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
348         - (V1y) * (V2x)
349         + (V1y) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
350         + (V2x) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
351         - (V2y) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
352     )
353     * (
354         (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
355         + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
356         + 1
357     )
358     ** 3

```

```

359     - S ** 2
360     * (-((V2x) ** 2) - (V2y) ** 2 + 1)
361     * (-((V1x) ** 2) - (V1y) ** 2 + 1) ** (3 / 2)
362     * (
363         (V2x) ** 2 * (x2 - x1) ** 2 / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
364         + 2
365         * (V2x)
366         * (V2y)
367         * (x2 - x1)
368         * (y2 - y1)
369         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
370         + (V2y) ** 2
371         * (y2 - y1) ** 2
372         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
373         - 2 * (V2x) ** 2
374         - 2 * (V2y) ** 2
375         + 1
376     )
377     * (V1x - (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2))
378     * (
379         (V1x)
380         * (V2x)
381         * (y2 - y1) ** 2
382         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
383         - (V1x)
384         * (V2y)
385         * (x2 - x1)
386         * (y2 - y1)
387         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
388         - (V1y)
389         * (V2x)
390         * (x2 - x1)
391         * (y2 - y1)
392         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
393         + (V1y)
394         * (V2y)
395         * (x2 - x1) ** 2
396         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
397         - (V1x) * (V2x)
398         - (V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
399         - (V1y) * (V2y)
400         - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
401         - (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
402         - (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
403         - 1
404     )
405 )
406 * S
407 / (
408     ((x2 - x1) ** 2 + (y2 - y1) ** 2)
409     * eta
410     * (
411         (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
412         + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
413         + 1
414     )
415 ** 3
416 * (
417     -((x2 - x1) ** 2 + (y2 - y1) ** 2)
418     * eta

```

```

419     * (
420         (V2x)
421         * (x2 - x1)
422         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
423         + (V2y)
424         * (y2 - y1)
425         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
426         + 1
427     )
428     ** 3
429     * (
430         -(V1x)
431         * (x2 - x1)
432         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
433         - (V1y)
434         * (y2 - y1)
435         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
436         + 1
437     )
438     ** 3
439     + S ** 2
440     * (
441         (
442             (V2x) ** 2
443             + (V2x)
444             * (x2 - x1)
445             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
446             + (V2y) ** 2
447             + (V2y)
448             * (y2 - y1)
449             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
450         )
451         * (
452             (V2x) ** 2
453             - (V2x)
454             * (x2 - x1)
455             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
456             + (V2y) ** 2
457             - (V2y)
458             * (y2 - y1)
459             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
460         )
461         * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
462         - (-((V2x) ** 2) - (V2y) ** 2 + 1) ** (5 / 2)
463     )
464     * (
465         (
466             (V1x) ** 2
467             + (V1x)
468             * (x2 - x1)
469             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
470             + (V1y) ** 2
471             + (V1y)
472             * (y2 - y1)
473             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
474         )
475         * (
476             (V1x) ** 2
477             - (V1x)
478             * (x2 - x1)

```

```

479         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
480         + (V1y) ** 2
481         - (V1y)
482         * (y2 - y1)
483         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
484     )
485     * np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
486     - (-((V1x) ** 2) - (V1y) ** 2 + 1) ** (5 / 2)
487 )
488 )
489 )
490 )
491
492 return a1x
493
494
495 # Aceleração da partícula 1 no eixo y para o tempo instantâneo
496 def d_V1y_inst(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
497
498     a1y = (
499         np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
500         * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
501         * S
502         * (
503             (
504                 (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
505                 + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
506                 + 1
507             )
508             ** 3
509             * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
510             * eta
511             * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
512             * (
513                 (V1x) ** 2 * (V2y)
514                 + (V1x) ** 2
515                 * (y2 - y1)
516                 / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
517                 - (V1x) * (V1y) * (V2x)
518                 - (V1x)
519                 * (V1y)
520                 * (x2 - x1)
521                 / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
522                 - (V1x)
523                 * (V2x)
524                 * (y2 - y1)
525                 / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
526                 + (V1x)
527                 * (V2y)
528                 * (x2 - x1)
529                 / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
530                 + (V2y) * (-((V1x) ** 2) - (V1y) ** 2 + 1)
531                 + (-((V1x) ** 2) - (V1y) ** 2 + 1)
532                 * (y2 - y1)
533                 / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
534             )
535             * (
536                 -(V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
537                 - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
538                 + 1

```

```

539 )
540 ** 3
541 + (
542     (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
543     + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
544     + 1
545 )
546 ** 3
547 * S
548 * eta
549 * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
550 * (-((V1x) ** 2) - (V1y) ** 2 + 1)
551 * (
552     (V1x) ** 2
553     * (x2 - x1)
554     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
555     + (V1x)
556     * (V1y)
557     * (y2 - y1)
558     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
559     + (-((V1x) ** 2) - (V1y) ** 2 + 1)
560     * (x2 - x1)
561     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
562     + V1x
563 )
564 * (
565     (V1x)
566     * (V2x)
567     * (V2y)
568     * (x2 - x1)
569     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
570     - (V1x)
571     * (V2x)
572     * (x2 - x1)
573     * (y2 - y1)
574     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
575     + (V1x)
576     * (V2y) ** 2
577     * (y2 - y1)
578     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
579     - (V1x)
580     * (V2y)
581     * (y2 - y1) ** 2
582     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
583     + 2
584     * (V1x)
585     * (-((V2x) ** 2) - (V2y) ** 2 + 1)
586     * (y2 - y1)
587     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
588     - (V1y)
589     * (V2x) ** 2
590     * (x2 - x1)
591     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
592     - (V1y)
593     * (V2x)
594     * (V2y)
595     * (y2 - y1)
596     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
597     + (V1y)
598     * (V2x)

```

```

599         * (x2 - x1) ** 2
600         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
601         + (V1y)
602         * (V2y)
603         * (x2 - x1)
604         * (y2 - y1)
605         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
606         - 2
607         * (V1y)
608         * (-((V2x) ** 2) - (V2y) ** 2 + 1)
609         * (x2 - x1)
610         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
611         + (V2x) ** 2
612         * (x2 - x1)
613         * (y2 - y1)
614         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
615         - (V2x)
616         * (V2y)
617         * (x2 - x1) ** 2
618         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
619         + (V2x)
620         * (V2y)
621         * (y2 - y1) ** 2
622         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
623         - (V2y) ** 2
624         * (x2 - x1)
625         * (y2 - y1)
626         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
627         + (V1x) * (V2y)
628         - (V1x) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
629         - (V1y) * (V2x)
630         + (V1y) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
631         + (V2x) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
632         - (V2y) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
633     )
634     - S ** 2
635     * (-((V2x) ** 2) - (V2y) ** 2 + 1)
636     * (-((V1x) ** 2) - (V1y) ** 2 + 1) ** (3 / 2)
637     * (
638         (V2x) ** 2 * (x2 - x1) ** 2 / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
639         + 2
640         * (V2x)
641         * (V2y)
642         * (x2 - x1)
643         * (y2 - y1)
644         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
645         + (V2y) ** 2
646         * (y2 - y1) ** 2
647         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
648         - 2 * (V2x) ** 2
649         - 2 * (V2y) ** 2
650         + 1
651     )
652     * (V1y - (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2))
653     * (
654         (V1x)
655         * (V2x)
656         * (y2 - y1) ** 2
657         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
658         - (V1x)

```

```

659         * (V2y)
660         * (x2 - x1)
661         * (y2 - y1)
662         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
663         - (V1y)
664         * (V2x)
665         * (x2 - x1)
666         * (y2 - y1)
667         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
668         + (V1y)
669         * (V2y)
670         * (x2 - x1) ** 2
671         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
672         - (V1x) * (V2x)
673         - (V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
674         - (V1y) * (V2y)
675         - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
676         - (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
677         - (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
678         - 1
679     )
680 )
681 / (
682     ((x2 - x1) ** 2 + (y2 - y1) ** 2)
683     * eta
684     * (
685         (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
686         + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
687         + 1
688     )
689     ** 3
690     * (
691         -((x2 - x1) ** 2 + (y2 - y1) ** 2)
692         * eta
693         * (
694             (V2x)
695             * (x2 - x1)
696             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
697             + (V2y)
698             * (y2 - y1)
699             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
700             + 1
701         )
702         ** 3
703         * (
704             -(V1x)
705             * (x2 - x1)
706             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
707             - (V1y)
708             * (y2 - y1)
709             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
710             + 1
711         )
712         ** 3
713     + S ** 2
714     * (
715         (
716             (V2x) ** 2
717             + (V2x)
718             * (x2 - x1)

```

```

719         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
720         + (V2y) ** 2
721         + (V2y)
722         * (y2 - y1)
723         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
724     )
725     * (
726         (V2x) ** 2
727         - (V2x)
728         * (x2 - x1)
729         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
730         + (V2y) ** 2
731         - (V2y)
732         * (y2 - y1)
733         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
734     )
735     * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
736     - (-((V2x) ** 2) - (V2y) ** 2 + 1) ** (5 / 2)
737 )
738 * (
739     (
740         (V1x) ** 2
741         + (V1x)
742         * (x2 - x1)
743         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
744         + (V1y) ** 2
745         + (V1y)
746         * (y2 - y1)
747         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
748     )
749     * (
750         (V1x) ** 2
751         - (V1x)
752         * (x2 - x1)
753         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
754         + (V1y) ** 2
755         - (V1y)
756         * (y2 - y1)
757         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
758     )
759     * np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
760     - (-((V1x) ** 2) - (V1y) ** 2 + 1) ** (5 / 2)
761 )
762 )
763 )
764 )
765
766     return a1y
767
768
769 # Aceleração da partícula 2 no eixo x para o tempo instantâneo
770 def d_V2x_inst(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
771
772     a2x = (
773         np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
774         * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
775         * S
776         * (
777             (
778                 -(V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)

```



```

779         - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
780         + 1
781     )
782     ** 3
783     * np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
784     * eta
785     * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
786     * (
787         (V1x) * (V2y) ** 2
788         - (V1x)
789         * (V2y)
790         * (y2 - y1)
791         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
792         + (V1x) * (-((V2x) ** 2) - (V2y) ** 2 + 1)
793         - (V1y) * (V2x) * (V2y)
794         + (V1y)
795         * (V2y)
796         * (x2 - x1)
797         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
798         + (V2x)
799         * (V2y)
800         * (y2 - y1)
801         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
802         - (V2y) ** 2
803         * (x2 - x1)
804         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
805         - (-((V2x) ** 2) - (V2y) ** 2 + 1)
806         * (x2 - x1)
807         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
808     )
809     * (
810         (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
811         + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
812         + 1
813     )
814     ** 3
815     + (
816         -(V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
817         - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
818         + 1
819     )
820     ** 3
821     * S
822     * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
823     * (-((V2x) ** 2) - (V2y) ** 2 + 1)
824     * (
825         (V2x)
826         * (V2y)
827         * (x2 - x1)
828         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
829         + (V2y) ** 2
830         * (y2 - y1)
831         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
832         + (-((V2x) ** 2) - (V2y) ** 2 + 1)
833         * (y2 - y1)
834         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
835         - (V2y)
836     )
837     * (
838         (V1x) ** 2

```

```

839     * (V2y)
840     * (x2 - x1)
841     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
842     + (V1x) ** 2
843     * (x2 - x1)
844     * (y2 - y1)
845     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
846     - (V1x)
847     * (V1y)
848     * (V2x)
849     * (x2 - x1)
850     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
851     + (V1x)
852     * (V1y)
853     * (V2y)
854     * (y2 - y1)
855     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
856     - (V1x)
857     * (V1y)
858     * (x2 - x1) ** 2
859     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
860     + (V1x)
861     * (V1y)
862     * (y2 - y1) ** 2
863     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
864     - (V1x)
865     * (V2x)
866     * (x2 - x1)
867     * (y2 - y1)
868     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
869     + (V1x)
870     * (V2y)
871     * (x2 - x1) ** 2
872     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
873     - (V1y) ** 2
874     * (V2x)
875     * (y2 - y1)
876     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
877     - (V1y) ** 2
878     * (x2 - x1)
879     * (y2 - y1)
880     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
881     - (V1y)
882     * (V2x)
883     * (y2 - y1) ** 2
884     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
885     + (V1y)
886     * (V2y)
887     * (x2 - x1)
888     * (y2 - y1)
889     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
890     - 2
891     * (V2x)
892     * (-((V1x) ** 2) - (V1y) ** 2 + 1)
893     * (y2 - y1)
894     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
895     + 2
896     * (V2y)
897     * (-((V1x) ** 2) - (V1y) ** 2 + 1)
898     * (x2 - x1)

```

```

899         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
900         - (V1x) * (V2y)
901         - (V1x) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
902         + (V1y) * (V2x)
903         + (V1y) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
904         + (V2x) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
905         - (V2y) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
906     )
907     - S ** 2
908     * (-((V1x) ** 2) - (V1y) ** 2 + 1)
909     * (-((V2x) ** 2) - (V2y) ** 2 + 1) ** (3 / 2)
910     * (
911         (V1x) ** 2 * (x2 - x1) ** 2 / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
912         + 2
913         * (V1x)
914         * (V1y)
915         * (x2 - x1)
916         * (y2 - y1)
917         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
918         + (V1y) ** 2
919         * (y2 - y1) ** 2
920         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
921         - 2 * (V1x) ** 2
922         - 2 * (V1y) ** 2
923         + 1
924     )
925     * (V2x + (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2))
926     * (
927         (V1x)
928         * (V2x)
929         * (y2 - y1) ** 2
930         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
931         - (V1x)
932         * (V2y)
933         * (x2 - x1)
934         * (y2 - y1)
935         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
936         - (V1y)
937         * (V2x)
938         * (x2 - x1)
939         * (y2 - y1)
940         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
941         + (V1y)
942         * (V2y)
943         * (x2 - x1) ** 2
944         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
945         - (V1x) * (V2x)
946         + (V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
947         - (V1y) * (V2y)
948         + (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
949         + (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
950         + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
951         - 1
952     )
953 )
954 / (
955     ((x2 - x1) ** 2 + (y2 - y1) ** 2)
956     * (
957         -(V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
958         - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)

```

```

959         + 1
960     )
961     ** 3
962     * (
963         -((x2 - x1) ** 2 + (y2 - y1) ** 2)
964         * eta
965         * (
966             (V2x)
967             * (x2 - x1)
968             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
969             + (V2y)
970             * (y2 - y1)
971             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
972             + 1
973         )
974     ** 3
975     * (
976         -(V1x)
977         * (x2 - x1)
978         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
979         - (V1y)
980         * (y2 - y1)
981         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
982         + 1
983     )
984     ** 3
985     + S ** 2
986     * (
987         (
988             (V2x) ** 2
989             + (V2x)
990             * (x2 - x1)
991             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
992             + (V2y) ** 2
993             + (V2y)
994             * (y2 - y1)
995             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
996         )
997         * (
998             (V2x) ** 2
999             - (V2x)
1000            * (x2 - x1)
1001            / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1002            + (V2y) ** 2
1003            - (V2y)
1004            * (y2 - y1)
1005            / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1006        )
1007        * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
1008        - (-((V2x) ** 2) - (V2y) ** 2 + 1) ** (5 / 2)
1009    )
1010    * (
1011        (
1012            (V1x) ** 2
1013            + (V1x)
1014            * (x2 - x1)
1015            / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1016            + (V1y) ** 2
1017            + (V1y)
1018            * (y2 - y1)

```

```

1019         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1020     )
1021     * (
1022         (V1x) ** 2
1023         - (V1x)
1024         * (x2 - x1)
1025         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1026         + (V1y) ** 2
1027         - (V1y)
1028         * (y2 - y1)
1029         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1030     )
1031     * np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
1032     - (-((V1x) ** 2) - (V1y) ** 2 + 1) ** (5 / 2)
1033     )
1034     )
1035     )
1036 )
1037
1038 return a2x
1039
1040
1041 # Aceleração da partícula 2 no eixo y para o tempo instantâneo
1042 def d_V2y_inst(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
1043
1044     a2y = (
1045         np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
1046         * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
1047         * (
1048             -(
1049                 (
1050                     -(V1x)
1051                     * (x2 - x1)
1052                     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1053                     - (V1y)
1054                     * (y2 - y1)
1055                     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1056                     + 1
1057                 )
1058                 ** 3
1059             )
1060         * np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
1061         * eta
1062         * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1063         * (
1064             (V1x) * (V2x) * (V2y)
1065             - (V1x)
1066             * (V2x)
1067             * (y2 - y1)
1068             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1069             - (V1y) * (V2x) ** 2
1070             + (V1y)
1071             * (V2x)
1072             * (x2 - x1)
1073             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1074             - (V1y) * (-((V2x) ** 2) - (V2y) ** 2 + 1)
1075             + (V2x) ** 2
1076             * (y2 - y1)
1077             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1078             - (V2x)

```

```

1079         * (V2y)
1080         * (x2 - x1)
1081         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1082         + (-((V2x) ** 2) - (V2y) ** 2 + 1)
1083         * (y2 - y1)
1084         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1085     )
1086     * (
1087         (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1088         + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1089         + 1
1090     )
1091     ** 3
1092     - (
1093         -(V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1094         - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1095         + 1
1096     )
1097     ** 3
1098     * S
1099     * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1100     * (-((V2x) ** 2) - (V2y) ** 2 + 1)
1101     * (
1102         (V2x) ** 2
1103         * (x2 - x1)
1104         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1105         + (V2x)
1106         * (V2y)
1107         * (y2 - y1)
1108         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1109         + (-((V2x) ** 2) - (V2y) ** 2 + 1)
1110         * (x2 - x1)
1111         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1112         - (V2x)
1113     )
1114     * (
1115         (V1x) ** 2
1116         * (V2y)
1117         * (x2 - x1)
1118         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1119         + (V1x) ** 2
1120         * (x2 - x1)
1121         * (y2 - y1)
1122         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1123         - (V1x)
1124         * (V1y)
1125         * (V2x)
1126         * (x2 - x1)
1127         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1128         + (V1x)
1129         * (V1y)
1130         * (V2y)
1131         * (y2 - y1)
1132         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1133         - (V1x)
1134         * (V1y)
1135         * (x2 - x1) ** 2
1136         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1137         + (V1x)
1138         * (V1y)

```

```

1139     * (y2 - y1) ** 2
1140     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1141     - (V1x)
1142     * (V2x)
1143     * (x2 - x1)
1144     * (y2 - y1)
1145     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1146     + (V1x)
1147     * (V2y)
1148     * (x2 - x1) ** 2
1149     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1150     - (V1y) ** 2
1151     * (V2x)
1152     * (y2 - y1)
1153     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1154     - (V1y) ** 2
1155     * (x2 - x1)
1156     * (y2 - y1)
1157     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1158     - (V1y)
1159     * (V2x)
1160     * (y2 - y1) ** 2
1161     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1162     + (V1y)
1163     * (V2y)
1164     * (x2 - x1)
1165     * (y2 - y1)
1166     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1167     - 2
1168     * (V2x)
1169     * (-(V1x) ** 2) - (V1y) ** 2 + 1)
1170     * (y2 - y1)
1171     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1172     + 2
1173     * (V2y)
1174     * (-(V1x) ** 2) - (V1y) ** 2 + 1)
1175     * (x2 - x1)
1176     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1177     - (V1x) * (V2y)
1178     - (V1x) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1179     + (V1y) * (V2x)
1180     + (V1y) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1181     + (V2x) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1182     - (V2y) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1183 )
1184 - S ** 2
1185 * (-(V1x) ** 2) - (V1y) ** 2 + 1)
1186 * (-(V2x) ** 2) - (V2y) ** 2 + 1) ** (3 / 2)
1187 * (
1188     (V1x) ** 2 * (x2 - x1) ** 2 / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1189     + 2
1190     * (V1x)
1191     * (V1y)
1192     * (x2 - x1)
1193     * (y2 - y1)
1194     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1195     + (V1y) ** 2
1196     * (y2 - y1) ** 2
1197     / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1198     - 2 * (V1x) ** 2

```

```

1199         - 2 * (V1y) ** 2
1200         + 1
1201     )
1202     * (V2y + (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2))
1203     * (
1204         (V1x)
1205         * (V2x)
1206         * (y2 - y1) ** 2
1207         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1208         - (V1x)
1209         * (V2y)
1210         * (x2 - x1)
1211         * (y2 - y1)
1212         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1213         - (V1y)
1214         * (V2x)
1215         * (x2 - x1)
1216         * (y2 - y1)
1217         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1218         + (V1y)
1219         * (V2y)
1220         * (x2 - x1) ** 2
1221         / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1222         - (V1x) * (V2x)
1223         + (V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1224         - (V1y) * (V2y)
1225         + (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1226         + (V2x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1227         + (V2y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1228         - 1
1229     )
1230 )
1231 * S
1232 / (
1233     ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1234     * (
1235         -(V1x) * (x2 - x1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1236         - (V1y) * (y2 - y1) / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1237         + 1
1238     )
1239 ** 3
1240 * (
1241     -((x2 - x1) ** 2 + (y2 - y1) ** 2)
1242     * eta
1243     * (
1244         (V2x)
1245         * (x2 - x1)
1246         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1247         + (V2y)
1248         * (y2 - y1)
1249         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1250         + 1
1251     )
1252 ** 3
1253 * (
1254     -(V1x)
1255     * (x2 - x1)
1256     / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1257     - (V1y)
1258     * (y2 - y1)

```



```

1259         / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1260         + 1
1261     )
1262     ** 3
1263     + S ** 2
1264     * (
1265         (
1266             (V2x) ** 2
1267             + (V2x)
1268             * (x2 - x1)
1269             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1270             + (V2y) ** 2
1271             + (V2y)
1272             * (y2 - y1)
1273             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1274         )
1275         * (
1276             (V2x) ** 2
1277             - (V2x)
1278             * (x2 - x1)
1279             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1280             + (V2y) ** 2
1281             - (V2y)
1282             * (y2 - y1)
1283             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1284         )
1285         * np.sqrt(-((V2x) ** 2) - (V2y) ** 2 + 1)
1286         - (-((V2x) ** 2) - (V2y) ** 2 + 1) ** (5 / 2)
1287     )
1288     * (
1289         (
1290             (V1x) ** 2
1291             + (V1x)
1292             * (x2 - x1)
1293             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1294             + (V1y) ** 2
1295             + (V1y)
1296             * (y2 - y1)
1297             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1298         )
1299         * (
1300             (V1x) ** 2
1301             - (V1x)
1302             * (x2 - x1)
1303             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1304             + (V1y) ** 2
1305             - (V1y)
1306             * (y2 - y1)
1307             / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1308         )
1309         * np.sqrt(-((V1x) ** 2) - (V1y) ** 2 + 1)
1310         - (-((V1x) ** 2) - (V1y) ** 2 + 1) ** (5 / 2)
1311     )
1312 )
1313 )
1314 )
1315
1316     return a2y
1317
1318

```

```

1319 # Equações com os tempos de retardo com aproximação instantânea
1320 # lineares nas velocidades (tr -> t)
1321
1322 # Aceleração da partícula 1 no eixo x instantâneo linear nas velocidades
1323 def d_V1x_inst_linear(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
1324
1325     a1x = (
1326         -S * (x2 - x1) / (((x2 - x1) ** 2 + (y2 - y1) ** 2) ** (3 / 2) * eta)
1327         + (y2 - y1) ** 2
1328         * (eta * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2) - S)
1329         * V1x
1330         / (
1331             ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1332             * eta
1333             * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1334         )
1335         - (y2 - y1)
1336         * (x2 - x1)
1337         * (eta * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2) - S)
1338         * V1y
1339         / (
1340             ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1341             * eta
1342             * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1343         )
1344         + (
1345             S * (y2 - y1) ** 2 * eta / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1346             - 3 * eta * (y2 - y1) ** 2
1347             + 2 * eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1348             + 2 * (y2 - y1) ** 2 / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1349             - 2
1350         )
1351         * S
1352         * V2x
1353         / (
1354             eta
1355             * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1356             * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1357         )
1358         - (y2 - y1)
1359         * (x2 - x1)
1360         * (
1361             S * eta * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1362             - 3 * eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1363             + 2
1364         )
1365         * S
1366         * V2y
1367         / (
1368             ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1369             * eta
1370             * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1371         )
1372     )
1373
1374     return a1x
1375
1376
1377 # Aceleração da partícula 1 no eixo y instantâneo linear nas velocidades
1378 def d_V1y_inst_linear(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):

```

```

1379
1380     a1y = (
1381         -S * (y2 - y1) / (((x2 - x1) ** 2 + (y2 - y1) ** 2) ** (3 / 2) * eta)
1382         - (y2 - y1)
1383         * (x2 - x1)
1384         * (eta * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2) - S)
1385         * V1x
1386         / (
1387             ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1388             * eta
1389             * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1390         )
1391         + (x2 - x1) ** 2
1392         * (eta * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2) - S)
1393         * V1y
1394         / (
1395             ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1396             * eta
1397             * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1398         )
1399         - (y2 - y1)
1400         * (x2 - x1)
1401         * (
1402             S * eta * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1403             - 3 * eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1404             + 2
1405         )
1406         * S
1407         * V2x
1408         / (
1409             ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1410             * eta
1411             * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1412         )
1413         + (
1414             S * (x2 - x1) ** 2 * eta / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1415             - 3 * eta * (x2 - x1) ** 2
1416             + 2 * eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1417             + 2 * (x2 - x1) ** 2 / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1418             - 2
1419         )
1420         * S
1421         * V2y
1422         / (
1423             eta
1424             * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1425             * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1426         )
1427     )
1428
1429     return a1y
1430
1431
1432 # Aceleração da partícula 2 no eixo x instantâneo linear nas velocidades
1433 def d_V2x_inst_linear(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
1434
1435     a2x = (
1436         S * (x2 - x1) / ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** (3 / 2)
1437         - (
1438             3 * eta * (y2 - y1) ** 2

```

```

1439     - S * (y2 - y1) ** 2 / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1440     - 2 * eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1441     - 2 * (y2 - y1) ** 2 / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1442     + 2
1443 )
1444 * S
1445 * V1x
1446 / (
1447     ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1448     * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1449 )
1450 + (y2 - y1)
1451 * (x2 - x1)
1452 * (
1453     3 * eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1454     - S * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1455     - 2
1456 )
1457 * S
1458 * V1y
1459 / (
1460     ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1461     * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1462 )
1463 - (y2 - y1) ** 2
1464 * (S - np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2))
1465 * V2x
1466 / (
1467     ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1468     * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1469 )
1470 + (y2 - y1)
1471 * (x2 - x1)
1472 * (S - np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2))
1473 * V2y
1474 / (
1475     ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1476     * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1477 )
1478 )
1479
1480 return a2x
1481
1482
1483 # Aceleração da partícula 2 no eixo y instantâneo linear nas velocidades
1484 def d_V2y_inst_linear(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
1485
1486     a2y = (
1487         S * (y2 - y1) / ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** (3 / 2)
1488         + (y2 - y1)
1489         * (x2 - x1)
1490         * (
1491             3 * eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1492             - S * np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1493             - 2
1494         )
1495         * S
1496         * V1x
1497         / (
1498             ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2

```

```

1499         * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1500     )
1501     - S
1502     * (
1503         3 * eta * (x2 - x1) ** 2
1504         - S * (x2 - x1) ** 2 / np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
1505         - 2 * eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1506         - 2 * (x2 - x1) ** 2 / ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1507         + 2
1508     )
1509     * V1y
1510     / (
1511         ((x2 - x1) ** 2 + (y2 - y1) ** 2)
1512         * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1513     )
1514     + (y2 - y1)
1515     * (x2 - x1)
1516     * (S - np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2))
1517     * V2x
1518     / (
1519         ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1520         * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1521     )
1522     - (x2 - x1) ** 2
1523     * (S - np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2))
1524     * V2y
1525     / (
1526         ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** 2
1527         * (eta * ((x2 - x1) ** 2 + (y2 - y1) ** 2) - 1)
1528     )
1529 )
1530
1531 return a2y
1532
1533 # ### Modelos de Runge-Kutta para os campos vetorias ###
1534
1535 # Runge-Kutta 4 ordem instantâneo - Aproximação de ordem zero do campo
1536 def rk4_0(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, step):
1537
1538     x1_1 = V1x * step
1539     y1_1 = V1y * step
1540     x2_1 = V2x * step
1541     y2_1 = V2y * step
1542
1543     V1x_1 = d_V1x_inst(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y) * step
1544     V1y_1 = d_V1y_inst(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y) * step
1545     V2x_1 = d_V2x_inst(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y) * step
1546     V2y_1 = d_V2y_inst(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y) * step
1547
1548     x1_k = x1 + x1_1 * 0.5
1549     y1_k = y1 + y1_1 * 0.5
1550     x2_k = x2 + x2_1 * 0.5
1551     y2_k = y2 + y2_1 * 0.5
1552     V1x_k = V1x + V1x_1 * 0.5
1553     V1y_k = V1y + V1y_1 * 0.5
1554     V2x_k = V2x + V2x_1 * 0.5
1555     V2y_k = V2y + V2y_1 * 0.5
1556
1557     x1_2 = V1x_k * step
1558     y1_2 = V1y_k * step

```

```
1559     x2_2 = V2x_k * step
1560     y2_2 = V2y_k * step
1561
1562     V1x_2 = (
1563         d_V1x_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1564         * step
1565     )
1566     V1y_2 = (
1567         d_V1y_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1568         * step
1569     )
1570     V2x_2 = (
1571         d_V2x_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1572         * step
1573     )
1574     V2y_2 = (
1575         d_V2y_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1576         * step
1577     )
1578
1579     x1_k = x1 + x1_2 * 0.5
1580     y1_k = y1 + y1_2 * 0.5
1581     x2_k = x2 + x2_2 * 0.5
1582     y2_k = y2 + y2_2 * 0.5
1583
1584     V1x_k = V1x + V1x_2 * 0.5
1585     V1y_k = V1y + V1y_2 * 0.5
1586     V2x_k = V2x + V2x_2 * 0.5
1587     V2y_k = V2y + V2y_2 * 0.5
1588
1589     x1_3 = V1x_k * step
1590     y1_3 = V1y_k * step
1591     x2_3 = V2x_k * step
1592     y2_3 = V2y_k * step
1593
1594     V1x_3 = (
1595         d_V1x_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1596         * step
1597     )
1598     V1y_3 = (
1599         d_V1y_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1600         * step
1601     )
1602     V2x_3 = (
1603         d_V2x_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1604         * step
1605     )
1606     V2y_3 = (
1607         d_V2y_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1608         * step
1609     )
1610
1611     x1_k = x1 + x1_3
1612     y1_k = y1 + y1_3
1613     x2_k = x2 + x2_3
1614     y2_k = y2 + y2_3
1615
1616     V1x_k = V1x + V1x_3
1617     V1y_k = V1y + V1y_3
1618     V2x_k = V2x + V2x_3
```

```

1619     V2y_k = V2y + V2y_3
1620
1621     x1_4 = V1x_k * step
1622     y1_4 = V1y_k * step
1623     x2_4 = V2x_k * step
1624     y2_4 = V2y_k * step
1625
1626     V1x_4 = (
1627         d_V1x_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1628         * step
1629     )
1630     V1y_4 = (
1631         d_V1y_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1632         * step
1633     )
1634     V2x_4 = (
1635         d_V2x_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1636         * step
1637     )
1638     V2y_4 = (
1639         d_V2y_inst(S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k)
1640         * step
1641     )
1642
1643     x1 = x1 + (x1_1 + 2 * (x1_2 + x1_3) + x1_4) / 6
1644     y1 = y1 + (y1_1 + 2 * (y1_2 + y1_3) + y1_4) / 6
1645     x2 = x2 + (x2_1 + 2 * (x2_2 + x2_3) + x2_4) / 6
1646     y2 = y2 + (y2_1 + 2 * (y2_2 + y2_3) + y2_4) / 6
1647
1648     V1x = V1x + (V1x_1 + 2 * (V1x_2 + V1x_3) + V1x_4) / 6
1649     V1y = V1y + (V1y_1 + 2 * (V1y_2 + V1y_3) + V1y_4) / 6
1650     V2x = V2x + (V2x_1 + 2 * (V2x_2 + V2x_3) + V2x_4) / 6
1651     V2y = V2y + (V2y_1 + 2 * (V2y_2 + V2y_3) + V2y_4) / 6
1652
1653     return step, x1, y1, x2, y2, V1x, V1y, V2x, V2y
1654
1655
1656 # Aplicação do método Runge Kutta - Aproximação de ordem zero
1657 def rk4_method_inst(
1658     path, n, m, f, t, dt, S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
1659 ):
1660
1661     with open(path, "w") as outfile:
1662         outfile.write(f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n")
1663         for _ in tqdm(range(1, m + 1, 1)):
1664             for _ in range(1, n + 1, 1):
1665                 t += dt
1666                 dt_atual, x1, y1, x2, y2, V1x, V1y, V2x, V2y = rk4_0(
1667                     S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, dt
1668                 )
1669                 dt = dt_atual
1670
1671                 outfile.write(
1672                     f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n"
1673                 )
1674
1675                 if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (
1676                     np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)) < 2
1677                 ):
1678                     break

```

```

1679
1680     vx = V2x - V1x
1681     vy = V2y - V1y
1682     rr = np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
1683     ex = (x2 - x1) / (rr)
1684     ey = (y2 - y1) / (rr)
1685     DT = (-vx * ey + vy * ex) / (rr)
1686     dt = (2.0 * np.pi) / (DT * n) / f
1687     if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (rr < 2):
1688         break
1689
1690
1691 # Campo para calcular no tempo de retardo - Aproximação de ordem 1
1692 def campo_w_1(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
1693     tr = -np.sqrt((x2 - x1) ** 2.0 + (y2 - y1) ** 2)
1694     _, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr = rk4_0(
1695         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, tr
1696     )
1697
1698     A1x_tr = d_V1x_inst_linear(
1699         S, eta, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr
1700     )
1701     A1y_tr = d_V1y_inst_linear(
1702         S, eta, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr
1703     )
1704     A2x_tr = d_V2x_inst_linear(
1705         S, eta, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr
1706     )
1707     A2y_tr = d_V2y_inst_linear(
1708         S, eta, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr
1709     )
1710
1711     A1x = f_a1x_tr(
1712         S, eta, x1, y1, x2_tr, y2_tr, V1x, V1y, V2x_tr, V2y_tr, A2x_tr, A2y_tr
1713     )
1714     A1y = f_a1y_tr(
1715         S, eta, x1, y1, x2_tr, y2_tr, V1x, V1y, V2x_tr, V2y_tr, A2x_tr, A2y_tr
1716     )
1717     A2x = f_a2x_tr(
1718         S, x1_tr, y1_tr, x2, y2, V1x_tr, V1y_tr, V2x, V2y, A1x_tr, A1y_tr
1719     )
1720     A2y = f_a2y_tr(
1721         S, x1_tr, y1_tr, x2, y2, V1x_tr, V1y_tr, V2x, V2y, A1x_tr, A1y_tr
1722     )
1723
1724     return A1x, A1y, A2x, A2y
1725
1726
1727 # Runge-Kutta 4 ordem com retardo - Aproximação de ordem 1 do campo
1728 def rk4_tr_1(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, h):
1729
1730     x1_1 = V1x * h
1731     y1_1 = V1y * h
1732     x2_1 = V2x * h
1733     y2_1 = V2y * h
1734
1735     V1x_1, V1y_1, V2x_1, V2y_1 = campo_w_1(
1736         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
1737     )
1738     V1x_1 = V1x_1 * h

```



```
1739     V1y_1 = V1y_1 * h
1740     V2x_1 = V2x_1 * h
1741     V2y_1 = V2y_1 * h
1742
1743     x1_k = x1 + x1_1 * 0.5
1744     y1_k = y1 + y1_1 * 0.5
1745     x2_k = x2 + x2_1 * 0.5
1746     y2_k = y2 + y2_1 * 0.5
1747     V1x_k = V1x + V1x_1 * 0.5
1748     V1y_k = V1y + V1y_1 * 0.5
1749     V2x_k = V2x + V2x_1 * 0.5
1750     V2y_k = V2y + V2y_1 * 0.5
1751
1752     x1_2 = V1x_k * h
1753     y1_2 = V1y_k * h
1754     x2_2 = V2x_k * h
1755     y2_2 = V2y_k * h
1756
1757     V1x_2, V1y_2, V2x_2, V2y_2 = campo_w_1(
1758         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
1759     )
1760     V1x_2 = V1x_2 * h
1761     V1y_2 = V1y_2 * h
1762     V2x_2 = V2x_2 * h
1763     V2y_2 = V2y_2 * h
1764
1765     x1_k = x1 + x1_2 * 0.5
1766     y1_k = y1 + y1_2 * 0.5
1767     x2_k = x2 + x2_2 * 0.5
1768     y2_k = y2 + y2_2 * 0.5
1769     V1x_k = V1x + V1x_2 * 0.5
1770     V1y_k = V1y + V1y_2 * 0.5
1771     V2x_k = V2x + V2x_2 * 0.5
1772     V2y_k = V2y + V2y_2 * 0.5
1773
1774     x1_3 = V1x_k * h
1775     y1_3 = V1y_k * h
1776     x2_3 = V2x_k * h
1777     y2_3 = V2y_k * h
1778
1779     V1x_3, V1y_3, V2x_3, V2y_3 = campo_w_1(
1780         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
1781     )
1782     V1x_3 = V1x_3 * h
1783     V1y_3 = V1y_3 * h
1784     V2x_3 = V2x_3 * h
1785     V2y_3 = V2y_3 * h
1786
1787     x1_k = x1 + x1_3
1788     y1_k = y1 + y1_3
1789     x2_k = x2 + x2_3
1790     y2_k = y2 + y2_3
1791     V1x_k = V1x + V1x_3
1792     V1y_k = V1y + V1y_3
1793     V2x_k = V2x + V2x_3
1794     V2y_k = V2y + V2y_3
1795
1796     x1_4 = V1x_k * h
1797     y1_4 = V1y_k * h
1798     x2_4 = V2x_k * h
```

```

1799     y2_4 = V2y_k * h
1800
1801     V1x_4, V1y_4, V2x_4, V2y_4 = campo_w_1(
1802         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
1803     )
1804     V1x_4 = V1x_4 * h
1805     V1y_4 = V1y_4 * h
1806     V2x_4 = V2x_4 * h
1807     V2y_4 = V2y_4 * h
1808
1809     x1 = x1 + (x1_1 + 2 * (x1_2 + x1_3) + x1_4) / 6
1810     y1 = y1 + (y1_1 + 2 * (y1_2 + y1_3) + y1_4) / 6
1811     x2 = x2 + (x2_1 + 2 * (x2_2 + x2_3) + x2_4) / 6
1812     y2 = y2 + (y2_1 + 2 * (y2_2 + y2_3) + y2_4) / 6
1813     V1x = V1x + (V1x_1 + 2 * (V1x_2 + V1x_3) + V1x_4) / 6
1814     V1y = V1y + (V1y_1 + 2 * (V1y_2 + V1y_3) + V1y_4) / 6
1815     V2x = V2x + (V2x_1 + 2 * (V2x_2 + V2x_3) + V2x_4) / 6
1816     V2y = V2y + (V2y_1 + 2 * (V2y_2 + V2y_3) + V2y_4) / 6
1817
1818     return h, x1, y1, x2, y2, V1x, V1y, V2x, V2y
1819
1820
1821 # Aplicação do método Runge-Kutta com retardo - Aproximação de ordem 1
1822 def rk4_method_tr_1(
1823     path, n, m, f, t, dt, S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
1824 ):
1825     with open(path, "w") as outfile:
1826         outfile.write(f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n")
1827
1828         for _ in tqdm(range(1, m + 1, 1)):
1829             for _ in range(1, n + 1, 1):
1830                 t += dt
1831                 dt_atual, x1, y1, x2, y2, V1x, V1y, V2x, V2y = rk4_tr_1(
1832                     S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, dt
1833                 )
1834                 dt = dt_atual
1835
1836                 outfile.write(
1837                     f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n"
1838                 )
1839
1840                 if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (
1841                     np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)) < 2
1842                 ):
1843                     break
1844
1845                 vx = V2x - V1x
1846                 vy = V2y - V1y
1847                 rr = np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
1848                 ex = (x2 - x1) / (rr)
1849                 ey = (y2 - y1) / (rr)
1850                 DT = (-vx * ey + vy * ex) / (rr)
1851                 dt = (2.0 * np.pi) / (DT * n) / f
1852                 if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (rr < 2):
1853                     break
1854
1855
1856 # Campo para calcular no tempo de retardo - Aproximação de ordem 2
1857 def campo_w_2(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
1858

```

```

1859     tr = -np.sqrt((x2 - x1) ** 2.0 + (y2 - y1) ** 2)
1860
1861     _, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr = rk4_tr_1(
1862         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, tr
1863     )
1864
1865     n = 100
1866     h = np.abs(tr) / n
1867     tr = tr + h
1868
1869     (
1870         -,
1871         x1_tr_h,
1872         y1_tr_h,
1873         x2_tr_h,
1874         y2_tr_h,
1875         V1x_tr_h,
1876         V1y_tr_h,
1877         V2x_tr_h,
1878         V2y_tr_h,
1879     ) = rk4_tr_1(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, tr)
1880
1881     A1x_tr = (V1x_tr_h - V1x_tr) / h
1882     A1y_tr = (V1y_tr_h - V1y_tr) / h
1883     A2x_tr = (V2x_tr_h - V2x_tr) / h
1884     A2y_tr = (V2y_tr_h - V2y_tr) / h
1885
1886     r12 = np.sqrt((-x1 + x2_tr) ** 2 + (-y1 + y2_tr) ** 2)
1887     r21 = np.sqrt((x1_tr - x2) ** 2 + (y1_tr - y2) ** 2)
1888
1889     e1X = (-x1_tr + x2) / r21
1890     e1Y = (-y1_tr + y2) / r21
1891
1892     e2X = (-x1 + x2_tr) / r12
1893     e2Y = (-y1 + y2_tr) / r12
1894
1895     A1x = (
1896         -(V1x ** 2) - V1y ** 2 + 1) ** 2
1897         * (
1898             -(V1x ** 2) + 1)
1899         * (
1900             -S
1901             * (V2y_tr + e2Y)
1902             * (V1y + e2Y)
1903             * A2x_tr
1904             / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1905             + S
1906             * (V2x_tr + e2X)
1907             * (V1y + e2Y)
1908             * A2y_tr
1909             / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1910             + S
1911             * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
1912             * (V2x_tr * e2Y * V1y - V2y_tr * e2X * V1y + V2x_tr + e2X)
1913             / (r12 ** 2 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1914         )
1915     - V1x
1916     * V1y
1917     * (
1918         S

```

```

1919         * (V2y_tr + e2Y)
1920         * (V1x + e2X)
1921         * A2x_tr
1922         / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1923         - S
1924         * (V2x_tr + e2X)
1925         * (V1x + e2X)
1926         * A2y_tr
1927         / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1928         - S
1929         * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
1930         * (V2x_tr * e2Y * V1x - V2y_tr * e2X * V1x - V2y_tr - e2Y)
1931         / (r12 ** 2 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1932     )
1933 )
1934 / eta
1935 )
1936 A1y = (
1937     -(V1x ** 2) - V1y ** 2 + 1) ** 2
1938 * (
1939     -V1x
1940     * V1y
1941     * (
1942         -S
1943         * (V2y_tr + e2Y)
1944         * (V1y + e2Y)
1945         * A2x_tr
1946         / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1947         + S
1948         * (V2x_tr + e2X)
1949         * (V1y + e2Y)
1950         * A2y_tr
1951         / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1952         + S
1953         * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
1954         * (V2x_tr * e2Y * V1y - V2y_tr * e2X * V1y + V2x_tr + e2X)
1955         / (r12 ** 2 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1956     )
1957 + (-(V1y ** 2) + 1)
1958 * (
1959     S
1960     * (V2y_tr + e2Y)
1961     * (V1x + e2X)
1962     * A2x_tr
1963     / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1964     - S
1965     * (V2x_tr + e2X)
1966     * (V1x + e2X)
1967     * A2y_tr
1968     / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1969     - S
1970     * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
1971     * (V2x_tr * e2Y * V1x - V2y_tr * e2X * V1x - V2y_tr - e2Y)
1972     / (r12 ** 2 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
1973 )
1974 )
1975 / eta
1976 )
1977 A2x = (-(V2x ** 2) - V2y ** 2 + 1) ** 2 * (
1978     -(V2x ** 2) + 1)

```

```

1979     * (
1980         -S
1981         * (V2y - e1Y)
1982         * (V1y_tr - e1Y)
1983         * A1x_tr
1984         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
1985         + S
1986         * (V2y - e1Y)
1987         * (V1x_tr - e1X)
1988         * A1y_tr
1989         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
1990         - S
1991         * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
1992         * (V1x_tr * e1Y * V2y - V1y_tr * e1X * V2y - V1x_tr + e1X)
1993         / (r21 ** 2 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
1994     )
1995     - V2x
1996     * V2y
1997     * (
1998         S
1999         * (V2x - e1X)
2000         * (V1y_tr - e1Y)
2001         * A1x_tr
2002         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2003         - S
2004         * (V2x - e1X)
2005         * (V1x_tr - e1X)
2006         * A1y_tr
2007         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2008         + S
2009         * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2010         * (V1x_tr * e1Y * V2x - V1y_tr * e1X * V2x + V1y_tr - e1Y)
2011         / (r21 ** 2 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2012     )
2013 )
2014 A2y = (-(V2x ** 2) - V2y ** 2 + 1) ** 2 * (
2015     -V2x
2016     * V2y
2017     * (
2018         -S
2019         * (V2y - e1Y)
2020         * (V1y_tr - e1Y)
2021         * A1x_tr
2022         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2023         + S
2024         * (V2y - e1Y)
2025         * (V1x_tr - e1X)
2026         * A1y_tr
2027         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2028         - S
2029         * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2030         * (V1x_tr * e1Y * V2y - V1y_tr * e1X * V2y - V1x_tr + e1X)
2031         / (r21 ** 2 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2032     )
2033     + (-(V2y ** 2) + 1)
2034     * (
2035         S
2036         * (V2x - e1X)
2037         * (V1y_tr - e1Y)
2038         * A1x_tr

```

```

2039         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2040         - S
2041         * (V2x - e1X)
2042         * (V1x_tr - e1X)
2043         * A1y_tr
2044         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2045         + S
2046         * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2047         * (V1x_tr * e1Y * V2x - V1y_tr * e1X * V2x + V1y_tr - e1Y)
2048         / (r21 ** 2 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2049     )
2050 )
2051
2052     return A1x, A1y, A2x, A2y
2053
2054
2055 # Runge-Kutta 4 ordem com retardo - Aproximação de ordem 2 do campo
2056 def rk4_tr_2(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, h):
2057
2058     x1_1 = V1x * h
2059     y1_1 = V1y * h
2060     x2_1 = V2x * h
2061     y2_1 = V2y * h
2062
2063     V1x_1, V1y_1, V2x_1, V2y_1 = campo_w_2(
2064         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
2065     )
2066     V1x_1 = V1x_1 * h
2067     V1y_1 = V1y_1 * h
2068     V2x_1 = V2x_1 * h
2069     V2y_1 = V2y_1 * h
2070
2071     x1_k = x1 + x1_1 * 0.5
2072     y1_k = y1 + y1_1 * 0.5
2073     x2_k = x2 + x2_1 * 0.5
2074     y2_k = y2 + y2_1 * 0.5
2075     V1x_k = V1x + V1x_1 * 0.5
2076     V1y_k = V1y + V1y_1 * 0.5
2077     V2x_k = V2x + V2x_1 * 0.5
2078     V2y_k = V2y + V2y_1 * 0.5
2079
2080     x1_2 = V1x_k * h
2081     y1_2 = V1y_k * h
2082     x2_2 = V2x_k * h
2083     y2_2 = V2y_k * h
2084
2085     V1x_2, V1y_2, V2x_2, V2y_2 = campo_w_2(
2086         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
2087     )
2088     V1x_2 = V1x_2 * h
2089     V1y_2 = V1y_2 * h
2090     V2x_2 = V2x_2 * h
2091     V2y_2 = V2y_2 * h
2092
2093     x1_k = x1 + x1_2 * 0.5
2094     y1_k = y1 + y1_2 * 0.5
2095     x2_k = x2 + x2_2 * 0.5
2096     y2_k = y2 + y2_2 * 0.5
2097     V1x_k = V1x + V1x_2 * 0.5
2098     V1y_k = V1y + V1y_2 * 0.5

```

```

2099     V2x_k = V2x + V2x_2 * 0.5
2100     V2y_k = V2y + V2y_2 * 0.5
2101
2102     x1_3 = V1x_k * h
2103     y1_3 = V1y_k * h
2104     x2_3 = V2x_k * h
2105     y2_3 = V2y_k * h
2106
2107     V1x_3, V1y_3, V2x_3, V2y_3 = campo_w_2(
2108         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
2109     )
2110     V1x_3 = V1x_3 * h
2111     V1y_3 = V1y_3 * h
2112     V2x_3 = V2x_3 * h
2113     V2y_3 = V2y_3 * h
2114
2115     x1_k = x1 + x1_3
2116     y1_k = y1 + y1_3
2117     x2_k = x2 + x2_3
2118     y2_k = y2 + y2_3
2119     V1x_k = V1x + V1x_3
2120     V1y_k = V1y + V1y_3
2121     V2x_k = V2x + V2x_3
2122     V2y_k = V2y + V2y_3
2123
2124     x1_4 = V1x_k * h
2125     y1_4 = V1y_k * h
2126     x2_4 = V2x_k * h
2127     y2_4 = V2y_k * h
2128
2129     V1x_4, V1y_4, V2x_4, V2y_4 = campo_w_2(
2130         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
2131     )
2132     V1x_4 = V1x_4 * h
2133     V1y_4 = V1y_4 * h
2134     V2x_4 = V2x_4 * h
2135     V2y_4 = V2y_4 * h
2136
2137     x1 = x1 + (x1_1 + 2 * (x1_2 + x1_3) + x1_4) / 6
2138     y1 = y1 + (y1_1 + 2 * (y1_2 + y1_3) + y1_4) / 6
2139     x2 = x2 + (x2_1 + 2 * (x2_2 + x2_3) + x2_4) / 6
2140     y2 = y2 + (y2_1 + 2 * (y2_2 + y2_3) + y2_4) / 6
2141     V1x = V1x + (V1x_1 + 2 * (V1x_2 + V1x_3) + V1x_4) / 6
2142     V1y = V1y + (V1y_1 + 2 * (V1y_2 + V1y_3) + V1y_4) / 6
2143     V2x = V2x + (V2x_1 + 2 * (V2x_2 + V2x_3) + V2x_4) / 6
2144     V2y = V2y + (V2y_1 + 2 * (V2y_2 + V2y_3) + V2y_4) / 6
2145
2146     return h, x1, y1, x2, y2, V1x, V1y, V2x, V2y
2147
2148
2149 # Aplicação do método Runge-Kutta com retardo - Aproximação de ordem 2
2150 def rk4_method_tr_2(
2151     path, n, m, f, t, dt, S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
2152 ):
2153     with open(path, "w") as outfile:
2154         outfile.write(f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n")
2155
2156         for _ in tqdm(range(1, m + 1, 1)):
2157             for _ in range(1, n + 1, 1):
2158                 t += dt

```

```

2159         dt_atual, x1, y1, x2, y2, V1x, V1y, V2x, V2y = rk4_tr_2(
2160             S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, dt
2161         )
2162         dt = dt_atual
2163
2164         outfile.write(
2165             f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n"
2166         )
2167
2168         if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (
2169             np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)) < 2
2170         ):
2171             break
2172
2173         vx = V2x - V1x
2174         vy = V2y - V1y
2175         rr = np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
2176         ex = (x2 - x1) / (rr)
2177         ey = (y2 - y1) / (rr)
2178         DT = (-vx * ey + vy * ex) / (rr)
2179         dt = (2.0 * np.pi) / (DT * n) / f
2180         if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (rr < 2):
2181             break
2182
2183
2184 # Campo para calcular no tempo de retardo - Aproximação de ordem 3
2185 def campo_w_3(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
2186
2187     tr = -np.sqrt((x2 - x1) ** 2.0 + (y2 - y1) ** 2)
2188
2189     _, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr = rk4_tr_2(
2190         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, tr
2191     )
2192
2193     n = 100
2194     h = np.abs(tr) / n
2195     tr = tr + h
2196
2197     (
2198         -,
2199         x1_tr_h,
2200         y1_tr_h,
2201         x2_tr_h,
2202         y2_tr_h,
2203         V1x_tr_h,
2204         V1y_tr_h,
2205         V2x_tr_h,
2206         V2y_tr_h,
2207     ) = rk4_tr_2(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, tr)
2208
2209     A1x_tr = (V1x_tr_h - V1x_tr) / h
2210     A1y_tr = (V1y_tr_h - V1y_tr) / h
2211     A2x_tr = (V2x_tr_h - V2x_tr) / h
2212     A2y_tr = (V2y_tr_h - V2y_tr) / h
2213
2214     r12 = np.sqrt((-x1 + x2_tr) ** 2 + (-y1 + y2_tr) ** 2)
2215     r21 = np.sqrt((x1_tr - x2) ** 2 + (y1_tr - y2) ** 2)
2216
2217     e1X = (-x1_tr + x2) / r21
2218     e1Y = (-y1_tr + y2) / r21

```



```

2219
2220     e2X = (-x1 + x2_tr) / r12
2221     e2Y = (-y1 + y2_tr) / r12
2222
2223     A1x = (
2224         (-(V1x ** 2) - V1y ** 2 + 1) ** 2
2225         * (
2226             (-(V1x ** 2) + 1)
2227             * (
2228                 -S
2229                 * (V2y_tr + e2Y)
2230                 * (V1y + e2Y)
2231                 * A2x_tr
2232                 / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2233                 + S
2234                 * (V2x_tr + e2X)
2235                 * (V1y + e2Y)
2236                 * A2y_tr
2237                 / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2238                 + S
2239                 * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
2240                 * (V2x_tr * e2Y * V1y - V2y_tr * e2X * V1y + V2x_tr + e2X)
2241                 / (r12 ** 2 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2242             )
2243             - V1x
2244             * V1y
2245             * (
2246                 S
2247                 * (V2y_tr + e2Y)
2248                 * (V1x + e2X)
2249                 * A2x_tr
2250                 / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2251                 - S
2252                 * (V2x_tr + e2X)
2253                 * (V1x + e2X)
2254                 * A2y_tr
2255                 / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2256                 - S
2257                 * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
2258                 * (V2x_tr * e2Y * V1x - V2y_tr * e2X * V1x - V2y_tr - e2Y)
2259                 / (r12 ** 2 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2260             )
2261         )
2262         / eta
2263     )
2264     A1y = (
2265         (-(V1x ** 2) - V1y ** 2 + 1) ** 2
2266         * (
2267             -V1x
2268             * V1y
2269             * (
2270                 -S
2271                 * (V2y_tr + e2Y)
2272                 * (V1y + e2Y)
2273                 * A2x_tr
2274                 / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2275                 + S
2276                 * (V2x_tr + e2X)
2277                 * (V1y + e2Y)
2278                 * A2y_tr

```

```

2279         / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2280         + S
2281         * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
2282         * (V2x_tr * e2Y * V1y - V2y_tr * e2X * V1y + V2x_tr + e2X)
2283         / (r12 ** 2 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2284     )
2285     + (-(V1y ** 2) + 1)
2286     * (
2287         S
2288         * (V2y_tr + e2Y)
2289         * (V1x + e2X)
2290         * A2x_tr
2291         / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2292         - S
2293         * (V2x_tr + e2X)
2294         * (V1x + e2X)
2295         * A2y_tr
2296         / (r12 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2297         - S
2298         * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
2299         * (V2x_tr * e2Y * V1x - V2y_tr * e2X * V1x - V2y_tr - e2Y)
2300         / (r12 ** 2 * (V2x_tr * e2X + V2y_tr * e2Y + 1) ** 3)
2301     )
2302 )
2303 / eta
2304 )
2305 A2x = (-(V2x ** 2) - V2y ** 2 + 1) ** 2 * (
2306     (-(V2x ** 2) + 1)
2307     * (
2308         -S
2309         * (V2y - e1Y)
2310         * (V1y_tr - e1Y)
2311         * A1x_tr
2312         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2313         + S
2314         * (V2y - e1Y)
2315         * (V1x_tr - e1X)
2316         * A1y_tr
2317         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2318         - S
2319         * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2320         * (V1x_tr * e1Y * V2y - V1y_tr * e1X * V2y - V1x_tr + e1X)
2321         / (r21 ** 2 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2322     )
2323 - V2x
2324 * V2y
2325 * (
2326     S
2327     * (V2x - e1X)
2328     * (V1y_tr - e1Y)
2329     * A1x_tr
2330     / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2331     - S
2332     * (V2x - e1X)
2333     * (V1x_tr - e1X)
2334     * A1y_tr
2335     / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2336     + S
2337     * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2338     * (V1x_tr * e1Y * V2x - V1y_tr * e1X * V2x + V1y_tr - e1Y)

```

```

2339         / (r21 ** 2 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2340     )
2341 )
2342 A2y = (-(V2x ** 2) - V2y ** 2 + 1) ** 2 * (
2343     -V2x
2344     * V2y
2345     * (
2346         -S
2347         * (V2y - e1Y)
2348         * (V1y_tr - e1Y)
2349         * A1x_tr
2350         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2351         + S
2352         * (V2y - e1Y)
2353         * (V1x_tr - e1X)
2354         * A1y_tr
2355         / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2356         - S
2357         * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2358         * (V1x_tr * e1Y * V2y - V1y_tr * e1X * V2y - V1x_tr + e1X)
2359         / (r21 ** 2 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2360     )
2361 + (-(V2y ** 2) + 1)
2362 * (
2363     S
2364     * (V2x - e1X)
2365     * (V1y_tr - e1Y)
2366     * A1x_tr
2367     / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2368     - S
2369     * (V2x - e1X)
2370     * (V1x_tr - e1X)
2371     * A1y_tr
2372     / (r21 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2373     + S
2374     * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2375     * (V1x_tr * e1Y * V2x - V1y_tr * e1X * V2x + V1y_tr - e1Y)
2376     / (r21 ** 2 * (-V1x_tr * e1X - V1y_tr * e1Y + 1) ** 3)
2377 )
2378 )
2379
2380 return A1x, A1y, A2x, A2y
2381
2382
2383 # Runge-Kutta 4 ordem com retardo - Aproximação de ordem 3 do campo
2384 def rk4_tr_3(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, h):
2385
2386     x1_1 = V1x * h
2387     y1_1 = V1y * h
2388     x2_1 = V2x * h
2389     y2_1 = V2y * h
2390
2391     V1x_1, V1y_1, V2x_1, V2y_1 = campo_w_3(
2392         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
2393     )
2394     V1x_1 = V1x_1 * h
2395     V1y_1 = V1y_1 * h
2396     V2x_1 = V2x_1 * h
2397     V2y_1 = V2y_1 * h
2398

```

```
2399     x1_k = x1 + x1_1 * 0.5
2400     y1_k = y1 + y1_1 * 0.5
2401     x2_k = x2 + x2_1 * 0.5
2402     y2_k = y2 + y2_1 * 0.5
2403     V1x_k = V1x + V1x_1 * 0.5
2404     V1y_k = V1y + V1y_1 * 0.5
2405     V2x_k = V2x + V2x_1 * 0.5
2406     V2y_k = V2y + V2y_1 * 0.5
2407
2408     x1_2 = V1x_k * h
2409     y1_2 = V1y_k * h
2410     x2_2 = V2x_k * h
2411     y2_2 = V2y_k * h
2412
2413     V1x_2, V1y_2, V2x_2, V2y_2 = campo_w_3(
2414         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
2415     )
2416     V1x_2 = V1x_2 * h
2417     V1y_2 = V1y_2 * h
2418     V2x_2 = V2x_2 * h
2419     V2y_2 = V2y_2 * h
2420
2421     x1_k = x1 + x1_2 * 0.5
2422     y1_k = y1 + y1_2 * 0.5
2423     x2_k = x2 + x2_2 * 0.5
2424     y2_k = y2 + y2_2 * 0.5
2425     V1x_k = V1x + V1x_2 * 0.5
2426     V1y_k = V1y + V1y_2 * 0.5
2427     V2x_k = V2x + V2x_2 * 0.5
2428     V2y_k = V2y + V2y_2 * 0.5
2429
2430     x1_3 = V1x_k * h
2431     y1_3 = V1y_k * h
2432     x2_3 = V2x_k * h
2433     y2_3 = V2y_k * h
2434
2435     V1x_3, V1y_3, V2x_3, V2y_3 = campo_w_3(
2436         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
2437     )
2438     V1x_3 = V1x_3 * h
2439     V1y_3 = V1y_3 * h
2440     V2x_3 = V2x_3 * h
2441     V2y_3 = V2y_3 * h
2442
2443     x1_k = x1 + x1_3
2444     y1_k = y1 + y1_3
2445     x2_k = x2 + x2_3
2446     y2_k = y2 + y2_3
2447     V1x_k = V1x + V1x_3
2448     V1y_k = V1y + V1y_3
2449     V2x_k = V2x + V2x_3
2450     V2y_k = V2y + V2y_3
2451
2452     x1_4 = V1x_k * h
2453     y1_4 = V1y_k * h
2454     x2_4 = V2x_k * h
2455     y2_4 = V2y_k * h
2456
2457     V1x_4, V1y_4, V2x_4, V2y_4 = campo_w_3(
2458         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
```

```

2459     )
2460     V1x_4 = V1x_4 * h
2461     V1y_4 = V1y_4 * h
2462     V2x_4 = V2x_4 * h
2463     V2y_4 = V2y_4 * h
2464
2465     x1 = x1 + (x1_1 + 2 * (x1_2 + x1_3) + x1_4) / 6
2466     y1 = y1 + (y1_1 + 2 * (y1_2 + y1_3) + y1_4) / 6
2467     x2 = x2 + (x2_1 + 2 * (x2_2 + x2_3) + x2_4) / 6
2468     y2 = y2 + (y2_1 + 2 * (y2_2 + y2_3) + y2_4) / 6
2469     V1x = V1x + (V1x_1 + 2 * (V1x_2 + V1x_3) + V1x_4) / 6
2470     V1y = V1y + (V1y_1 + 2 * (V1y_2 + V1y_3) + V1y_4) / 6
2471     V2x = V2x + (V2x_1 + 2 * (V2x_2 + V2x_3) + V2x_4) / 6
2472     V2y = V2y + (V2y_1 + 2 * (V2y_2 + V2y_3) + V2y_4) / 6
2473
2474     return h, x1, y1, x2, y2, V1x, V1y, V2x, V2y
2475
2476
2477 # Aplicação do método Runge-Kutta com retardo - Aproximação de ordem 3
2478 def rk4_method_tr_3(
2479     path, n, m, f, t, dt, S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
2480 ):
2481     with open(path, "w") as outfile:
2482         outfile.write(f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n")
2483
2484         for _ in tqdm(range(1, m + 1, 1)):
2485             for _ in range(1, n + 1, 1):
2486                 t += dt
2487                 dt_atual, x1, y1, x2, y2, V1x, V1y, V2x, V2y = rk4_tr_3(
2488                     S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, dt
2489                 )
2490                 dt = dt_atual
2491
2492                 outfile.write(
2493                     f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n"
2494                 )
2495
2496                 if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (
2497                     np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)) < 2
2498                 ):
2499                     break
2500
2501                 vx = V2x - V1x
2502                 vy = V2y - V1y
2503                 rr = np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
2504                 ex = (x2 - x1) / (rr)
2505                 ey = (y2 - y1) / (rr)
2506                 DT = (-vx * ey + vy * ex) / (rr)
2507                 dt = (2.0 * np.pi) / (DT * n) / f
2508                 if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (rr < 2):
2509                     break
2510
2511
2512 # ### Equações com os tempos retardados e avançados ###
2513
2514 # Aceleração da partícula 1 no eixo x para o tempo retardado e avançado
2515 def f_alx_tr_av(
2516     S,
2517     eta,
2518     x1,

```

```

2519     y1,
2520     x2_tr,
2521     y2_tr,
2522     x2_ta,
2523     y2_ta,
2524     V1x,
2525     V1y,
2526     V2x_tr,
2527     V2y_tr,
2528     V2x_ta,
2529     V2y_ta,
2530     A2x_tr,
2531     A2y_tr,
2532     A2x_ta,
2533     A2y_ta,
2534 ):
2535
2536     r12_tr = np.sqrt((-x1 + x2_tr) ** 2 + (-y1 + y2_tr) ** 2)
2537     e2X_tr = (-x1 + x2_tr) / r12_tr
2538     e2Y_tr = (-y1 + y2_tr) / r12_tr
2539
2540     r12_ta = np.sqrt((-x1 + x2_ta) ** 2 + (-y1 + y2_ta) ** 2)
2541     e2X_ta = (-x1 + x2_ta) / r12_ta
2542     e2Y_ta = (-y1 + y2_ta) / r12_ta
2543
2544     ff1x = (
2545         -(V1x ** 2) - V1y ** 2 + 1) ** 2
2546         * (
2547             -(V1x ** 2) + 1)
2548             * (
2549                 -(1 / 2)
2550                 * S
2551                 * (V2y_tr + e2Y_tr)
2552                 * (V1y + e2Y_tr)
2553                 * A2x_tr
2554                 / (r12_tr * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2555                 + (1 / 2)
2556                 * S
2557                 * (V2x_tr + e2X_tr)
2558                 * (V1y + e2Y_tr)
2559                 * A2y_tr
2560                 / (r12_tr * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2561                 + (1 / 2)
2562                 * S
2563                 * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
2564                 * (
2565                     V2x_tr * e2Y_tr * V1y
2566                     - V2y_tr * e2X_tr * V1y
2567                     + V2x_tr
2568                     + e2X_tr
2569                 )
2570                 / (r12_tr ** 2 * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2571                 - (1 / 2)
2572                 * S
2573                 * (-V2y_ta + e2Y_ta)
2574                 * (V1y + e2Y_ta)
2575                 * A2x_ta
2576                 / (r12_ta * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2577                 + (1 / 2)
2578                 * S

```

```

2579     * (-V2x_ta + e2X_ta)
2580     * (V1y + e2Y_ta)
2581     * A2y_ta
2582     / (r12_ta * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2583     + (1 / 2)
2584     * S
2585     * (V2x_ta ** 2 + V2y_ta ** 2 - 1)
2586     * (
2587         -V2x_ta * e2Y_ta * V1y
2588         + V2y_ta * e2X_ta * V1y
2589         - V2x_ta
2590         + e2X_ta
2591     )
2592     / (r12_ta ** 2 * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2593 )
2594 - V1x
2595 * V1y
2596 * (
2597     (1 / 2)
2598     * S
2599     * (V2y_tr + e2Y_tr)
2600     * (V1x + e2X_tr)
2601     * A2x_tr
2602     / (r12_tr * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2603     - (1 / 2)
2604     * S
2605     * (V2x_tr + e2X_tr)
2606     * (V1x + e2X_tr)
2607     * A2y_tr
2608     / (r12_tr * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2609     - (1 / 2)
2610     * S
2611     * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
2612     * (
2613         V2x_tr * e2Y_tr * V1x
2614         - V2y_tr * e2X_tr * V1x
2615         - V2y_tr
2616         - e2Y_tr
2617     )
2618     / (r12_tr ** 2 * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2619     + (1 / 2)
2620     * S
2621     * (-V2y_ta + e2Y_ta)
2622     * (V1x + e2X_ta)
2623     * A2x_ta
2624     / (r12_ta * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2625     - (1 / 2)
2626     * S
2627     * (-V2x_ta + e2X_ta)
2628     * (V1x + e2X_ta)
2629     * A2y_ta
2630     / (r12_ta * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2631     - (1 / 2)
2632     * S
2633     * (V2x_ta ** 2 + V2y_ta ** 2 - 1)
2634     * (
2635         -V2x_ta * e2Y_ta * V1x
2636         + V2y_ta * e2X_ta * V1x
2637         + V2y_ta
2638         - e2Y_ta

```

```

2639         )
2640         / (r12_ta ** 2 * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2641     )
2642 )
2643 / eta
2644 )
2645 return ff1x
2646
2647
2648 # Aceleração da partícula 1 no eixo y para o tempo retardado e avançado
2649 def f_a1y_tr_av(
2650     S,
2651     eta,
2652     x1,
2653     y1,
2654     x2_tr,
2655     y2_tr,
2656     x2_ta,
2657     y2_ta,
2658     V1x,
2659     V1y,
2660     V2x_tr,
2661     V2y_tr,
2662     V2x_ta,
2663     V2y_ta,
2664     A2x_tr,
2665     A2y_tr,
2666     A2x_ta,
2667     A2y_ta,
2668 ):
2669
2670     r12_tr = np.sqrt((-x1 + x2_tr) ** 2 + (-y1 + y2_tr) ** 2)
2671     e2X_tr = (-x1 + x2_tr) / r12_tr
2672     e2Y_tr = (-y1 + y2_tr) / r12_tr
2673
2674     r12_ta = np.sqrt((-x1 + x2_ta) ** 2 + (-y1 + y2_ta) ** 2)
2675     e2X_ta = (-x1 + x2_ta) / r12_ta
2676     e2Y_ta = (-y1 + y2_ta) / r12_ta
2677     ff1y = (
2678         -(V1x ** 2) - V1y ** 2 + 1) ** 2
2679         * (
2680             -V1x
2681             * V1y
2682             * (
2683                 -(1 / 2)
2684                 * S
2685                 * (V2y_tr + e2Y_tr)
2686                 * (V1y + e2Y_tr)
2687                 * A2x_tr
2688                 / (r12_tr * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2689                 + (1 / 2)
2690                 * S
2691                 * (V2x_tr + e2X_tr)
2692                 * (V1y + e2Y_tr)
2693                 * A2y_tr
2694                 / (r12_tr * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2695                 + (1 / 2)
2696                 * S
2697                 * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
2698                 * (

```



```

2699         V2x_tr * e2Y_tr * V1y
2700         - V2y_tr * e2X_tr * V1y
2701         + V2x_tr
2702         + e2X_tr
2703     )
2704     / (r12_tr ** 2 * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2705     - (1 / 2)
2706     * S
2707     * (-V2y_ta + e2Y_ta)
2708     * (V1y + e2Y_ta)
2709     * A2x_ta
2710     / (r12_ta * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2711     + (1 / 2)
2712     * S
2713     * (-V2x_ta + e2X_ta)
2714     * (V1y + e2Y_ta)
2715     * A2y_ta
2716     / (r12_ta * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2717     + (1 / 2)
2718     * S
2719     * (V2x_ta ** 2 + V2y_ta ** 2 - 1)
2720     * (
2721         -V2x_ta * e2Y_ta * V1y
2722         + V2y_ta * e2X_ta * V1y
2723         - V2x_ta
2724         + e2X_ta
2725     )
2726     / (r12_ta ** 2 * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2727 )
2728 + (-(V1y ** 2) + 1)
2729 * (
2730     (1 / 2)
2731     * S
2732     * (V2y_tr + e2Y_tr)
2733     * (V1x + e2X_tr)
2734     * A2x_tr
2735     / (r12_tr * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2736     - (1 / 2)
2737     * S
2738     * (V2x_tr + e2X_tr)
2739     * (V1x + e2X_tr)
2740     * A2y_tr
2741     / (r12_tr * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2742     - (1 / 2)
2743     * S
2744     * (V2x_tr ** 2 + V2y_tr ** 2 - 1)
2745     * (
2746         V2x_tr * e2Y_tr * V1x
2747         - V2y_tr * e2X_tr * V1x
2748         - V2y_tr
2749         - e2Y_tr
2750     )
2751     / (r12_tr ** 2 * (V2x_tr * e2X_tr + V2y_tr * e2Y_tr + 1) ** 3)
2752     + (1 / 2)
2753     * S
2754     * (-V2y_ta + e2Y_ta)
2755     * (V1x + e2X_ta)
2756     * A2x_ta
2757     / (r12_ta * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2758     - (1 / 2)

```

```

2759         * S
2760         * (-V2x_ta + e2X_ta)
2761         * (V1x + e2X_ta)
2762         * A2y_ta
2763         / (r12_ta * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2764         - (1 / 2)
2765         * S
2766         * (V2x_ta ** 2 + V2y_ta ** 2 - 1)
2767         * (
2768             -V2x_ta * e2Y_ta * V1x
2769             + V2y_ta * e2X_ta * V1x
2770             + V2y_ta
2771             - e2Y_ta
2772         )
2773         / (r12_ta ** 2 * (-V2x_ta * e2X_ta - V2y_ta * e2Y_ta + 1) ** 3)
2774     )
2775 )
2776 / eta
2777 )
2778
2779 return ff1y
2780
2781
2782 # Aceleração da partícula 2 no eixo x para o tempo retardado e avançado
2783 def f_a2x_tr_av(
2784     S,
2785     x1_tr,
2786     y1_tr,
2787     x1_ta,
2788     y1_ta,
2789     x2,
2790     y2,
2791     V1x_tr,
2792     V1y_tr,
2793     V1x_ta,
2794     V1y_ta,
2795     V2x,
2796     V2y,
2797     A1x_tr,
2798     A1y_tr,
2799     A1x_ta,
2800     A1y_ta,
2801 ):
2802
2803     r21_tr = np.sqrt((x1_tr - x2) ** 2 + (y1_tr - y2) ** 2)
2804     e1X_tr = (-x1_tr + x2) / r21_tr
2805     e1Y_tr = (-y1_tr + y2) / r21_tr
2806
2807     r21_ta = np.sqrt((x1_ta - x2) ** 2 + (y1_ta - y2) ** 2)
2808     e1X_ta = (-x1_ta + x2) / r21_ta
2809     e1Y_ta = (-y1_ta + y2) / r21_ta
2810
2811     ff2x = (- (V2x ** 2) - V2y ** 2 + 1) ** 2 * (
2812         (- (V2x ** 2) + 1)
2813         * (
2814             - (1 / 2)
2815             * S
2816             * (V2y - e1Y_tr)
2817             * (V1y_tr - e1Y_tr)
2818             * A1x_tr

```

```

2819 / (r21_tr * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2820 + (1 / 2)
2821 * S
2822 * (V2y - e1Y_tr)
2823 * (V1x_tr - e1X_tr)
2824 * A1y_tr
2825 / (r21_tr * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2826 - (1 / 2)
2827 * S
2828 * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2829 * (V1x_tr * e1Y_tr * V2y - V1y_tr * e1X_tr * V2y - V1x_tr + e1X_tr)
2830 / (r21_tr ** 2 * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2831 - (1 / 2)
2832 * S
2833 * (V2y - e1Y_ta)
2834 * (-V1y_ta - e1Y_ta)
2835 * A1x_ta
2836 / (r21_ta * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2837 + (1 / 2)
2838 * S
2839 * (V2y - e1Y_ta)
2840 * (-V1x_ta - e1X_ta)
2841 * A1y_ta
2842 / (r21_ta * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2843 - (1 / 2)
2844 * S
2845 * (V1x_ta ** 2 + V1y_ta ** 2 - 1)
2846 * (
2847     -V1x_ta * e1Y_ta * V2y
2848     + V1y_ta * e1X_ta * V2y
2849     + V1x_ta
2850     + e1X_ta
2851 )
2852 / (r21_ta ** 2 * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2853 )
2854 - V2x
2855 * V2y
2856 * (
2857     (1 / 2)
2858     * S
2859     * (V2x - e1X_tr)
2860     * (V1y_tr - e1Y_tr)
2861     * A1x_tr
2862     / (r21_tr * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2863     - (1 / 2)
2864     * S
2865     * (V2x - e1X_tr)
2866     * (V1x_tr - e1X_tr)
2867     * A1y_tr
2868     / (r21_tr * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2869     + (1 / 2)
2870     * S
2871     * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2872     * (V1x_tr * e1Y_tr * V2x - V1y_tr * e1X_tr * V2x + V1y_tr - e1Y_tr)
2873     / (r21_tr ** 2 * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2874     + (1 / 2)
2875     * S
2876     * (V2x - e1X_ta)
2877     * (-V1y_ta - e1Y_ta)
2878     * A1x_ta

```

```

2879         / (r21_ta * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2880         - (1 / 2)
2881         * S
2882         * (V2x - e1X_ta)
2883         * (-V1x_ta - e1X_ta)
2884         * A1y_ta
2885         / (r21_ta * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2886         + (1 / 2)
2887         * S
2888         * (V1x_ta ** 2 + V1y_ta ** 2 - 1)
2889         * (
2890             -V1x_ta * e1Y_ta * V2x
2891             + V1y_ta * e1X_ta * V2x
2892             - V1y_ta
2893             - e1Y_ta
2894         )
2895         / (r21_ta ** 2 * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2896     )
2897 )
2898
2899     return ff2x
2900
2901
2902 # Aceleração da partícula 2 no eixo y para o tempo retardado e avançado
2903 def f_a2y_tr_av(
2904     S,
2905     x1_tr,
2906     y1_tr,
2907     x1_ta,
2908     y1_ta,
2909     x2,
2910     y2,
2911     V1x_tr,
2912     V1y_tr,
2913     V1x_ta,
2914     V1y_ta,
2915     V2x,
2916     V2y,
2917     A1x_tr,
2918     A1y_tr,
2919     A1x_ta,
2920     A1y_ta,
2921 ):
2922
2923     r21_tr = np.sqrt((x1_tr - x2) ** 2 + (y1_tr - y2) ** 2)
2924     e1X_tr = (-x1_tr + x2) / r21_tr
2925     e1Y_tr = (-y1_tr + y2) / r21_tr
2926
2927     r21_ta = np.sqrt((x1_ta - x2) ** 2 + (y1_ta - y2) ** 2)
2928     e1X_ta = (-x1_ta + x2) / r21_ta
2929     e1Y_ta = (-y1_ta + y2) / r21_ta
2930     ff2y = ((-V2x ** 2) - V2y ** 2 + 1) ** 2 * (
2931         -V2x
2932         * V2y
2933         * (
2934             -(1 / 2)
2935             * S
2936             * (V2y - e1Y_tr)
2937             * (V1y_tr - e1Y_tr)
2938             * A1x_tr

```

```

2939 / (r21_tr * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2940 + (1 / 2)
2941 * S
2942 * (V2y - e1Y_tr)
2943 * (V1x_tr - e1X_tr)
2944 * A1y_tr
2945 / (r21_tr * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2946 - (1 / 2)
2947 * S
2948 * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2949 * (V1x_tr * e1Y_tr * V2y - V1y_tr * e1X_tr * V2y - V1x_tr + e1X_tr)
2950 / (r21_tr ** 2 * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2951 - (1 / 2)
2952 * S
2953 * (V2y - e1Y_ta)
2954 * (-V1y_ta - e1Y_ta)
2955 * A1x_ta
2956 / (r21_ta * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2957 + (1 / 2)
2958 * S
2959 * (V2y - e1Y_ta)
2960 * (-V1x_ta - e1X_ta)
2961 * A1y_ta
2962 / (r21_ta * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2963 - (1 / 2)
2964 * S
2965 * (V1x_ta ** 2 + V1y_ta ** 2 - 1)
2966 * (
2967     -V1x_ta * e1Y_ta * V2y
2968     + V1y_ta * e1X_ta * V2y
2969     + V1x_ta
2970     + e1X_ta
2971 )
2972 / (r21_ta ** 2 * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
2973 )
2974 + (-(V2y ** 2) + 1)
2975 * (
2976     (1 / 2)
2977     * S
2978     * (V2x - e1X_tr)
2979     * (V1y_tr - e1Y_tr)
2980     * A1x_tr
2981     / (r21_tr * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2982     - (1 / 2)
2983     * S
2984     * (V2x - e1X_tr)
2985     * (V1x_tr - e1X_tr)
2986     * A1y_tr
2987     / (r21_tr * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2988     + (1 / 2)
2989     * S
2990     * (V1x_tr ** 2 + V1y_tr ** 2 - 1)
2991     * (V1x_tr * e1Y_tr * V2x - V1y_tr * e1X_tr * V2x + V1y_tr - e1Y_tr)
2992     / (r21_tr ** 2 * (-V1x_tr * e1X_tr - V1y_tr * e1Y_tr + 1) ** 3)
2993     + (1 / 2)
2994     * S
2995     * (V2x - e1X_ta)
2996     * (-V1y_ta - e1Y_ta)
2997     * A1x_ta
2998     / (r21_ta * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)

```

```

2999         - (1 / 2)
3000         * S
3001         * (V2x - e1X_ta)
3002         * (-V1x_ta - e1X_ta)
3003         * A1y_ta
3004         / (r21_ta * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
3005         + (1 / 2)
3006         * S
3007         * (V1x_ta ** 2 + V1y_ta ** 2 - 1)
3008         * (
3009             -V1x_ta * e1Y_ta * V2x
3010             + V1y_ta * e1X_ta * V2x
3011             - V1y_ta
3012             - e1Y_ta
3013         )
3014         / (r21_ta ** 2 * (V1x_ta * e1X_ta + V1y_ta * e1Y_ta + 1) ** 3)
3015     )
3016 )
3017
3018     return ff2y
3019
3020
3021 # ## Equações instantâneas para o campo simétrico (tr -> t, ta -> t) ##
3022
3023 # Aceleração da partícula 1 no eixo x campo simétrico
3024 def d_V1x_inst_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3025
3026     r = np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
3027     eX = (x2 - x1) / (r)
3028     eY = (y2 - y1) / (r)
3029
3030     a1x = (
3031         -(eX * S) / (eta * r * r)
3032         + (3 * V1x * V1x * eX * eY * eY * (-eta * r + S))
3033         / (r * r * eta * (-r * r * eta + 1))
3034         - (3 * (-eta * r + S) * eY * eY * eX * V1y * V1y)
3035         / (r * r * eta * (-r * r * eta + 1))
3036         - (3 * (-eta * r + S) * (2 * eX * eX - 1) * eY * V1x * V1y)
3037         / (r * r * eta * (-r * r * eta + 1))
3038         - (
3039             (
3040                 eta * r * (2 * r * (-eX * eX + 2 * eY * eY) - 3 * S * eY * eY)
3041                 + 3 * eX * eX
3042                 - 1
3043             )
3044             * eX
3045             * S
3046             * V2x
3047             * V2x
3048         )
3049         / (r * r * eta * (-r * r * eta + 1))
3050         - (
3051             eX
3052             * (
3053                 eta * (3 * S * r * eY * eY - r * r * (6 * eY * eY - 1))
3054                 + 3 * eY * eY
3055                 - 1
3056             )
3057             * S
3058             * V2y

```

```

3059         * V2y
3060     )
3061     / (r * r * eta * (-r * r * eta + 1))
3062     - (
3063         3
3064         * S
3065         * eY
3066         * (
3067             eta * r * (r * (-4 * eX * eX + 1) + S * (2 * eX * eX - 1))
3068             + 2 * eX * eX
3069         )
3070         * V2x
3071         * V2y
3072     )
3073     / (r * r * eta * (-r * r * eta + 1))
3074     - (eY * eY * eX * V1x * V2x) / (eta * r * r * r)
3075     + ((eY * eY * eta * r * r - S * r + eX * eX) * eY * V1x * V2y)
3076     / (eta * r * r * r * (-r * r * eta + 1))
3077     + (
3078         (-((eX * eX + 1) * S - r) * r * r * eta + S * eX * eX)
3079         * S
3080         * eY
3081         * V1y
3082         * V2x
3083     )
3084     / (eta * r * r * r * (-r * r * eta + 1))
3085     + ((-S * eX * eX + r) * eX * S * V1y * V2y) / (eta * r * r * r)
3086 )
3087
3088 return a1x
3089
3090
3091 # Aceleração da partícula 1 no eixo y campo simétrico
3092 def d_V1y_inst_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3093
3094     r = np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
3095     eX = (x2 - x1) / (r)
3096     eY = (y2 - y1) / (r)
3097
3098     a1y = (
3099         -(eY * S) / (r * r * eta)
3100         - ((3 * (-eta * r + S)) * eX * eX * eY * V1x * V1x)
3101         / (eta * r * r * (-eta * r * r + 1))
3102         + (3 * eX * eX * (-eta * r + S) * eY * V1y * V1y)
3103         / (r * r * eta * (-eta * r * r + 1))
3104         - ((3 * (2 * eY * eY - 1)) * (-eta * r + S) * eX * V1x * V1y)
3105         / (r * r * eta * (-eta * r * r + 1))
3106     - (
3107         (
3108             eta * (3 * S * r * eX * eX - (6 * eX * eX - 1) * r * r)
3109             + 3 * eX * eX
3110             - 1
3111         )
3112         * S
3113         * eY
3114         * V2x
3115         * V2x
3116     )
3117     / (r * r * eta * (-eta * r * r + 1))
3118     - (

```

```

3119         eY
3120         * S
3121         * (
3122             eta * r * (2 * r * (2 * eX * eX - eY * eY) - 3 * S * eX * eX)
3123             + 3 * eY * eY
3124             - 1
3125         )
3126         * V2y
3127         * V2y
3128     )
3129 / (r * r * eta * (-eta * r * r + 1))
3130 - (
3131     3
3132     * eX
3133     * S
3134     * (
3135         eta * r * (r * (-4 * eY * eY + 1) + S * (2 * eY * eY - 1))
3136         + 2 * eY * eY
3137     )
3138     * V2x
3139     * V2y
3140 )
3141 / (r * r * eta * (-eta * r * r + 1))
3142 + ((-S * eY * eY + r) * eY * S * V1x * V2x) / (eta * r * r * r)
3143 + (
3144     S
3145     * eX
3146     * (-((eY * eY + 1) * S - r) * r * r * eta + S * eY * eY)
3147     * V1x
3148     * V2y
3149 )
3150 / (eta * r * r * r * (-eta * r * r + 1))
3151 + (eX * (eX * eX * eta * r * r - S * r + eY * eY) * V1y * V2x)
3152 / (eta * r * r * r * (-eta * r * r + 1))
3153 - (eX * eX * eY * V1y * V2y) / (eta * r * r * r)
3154 )
3155
3156 return a1y
3157
3158
3159 # Aceleração da partícula 2 no eixo x campo simétrico
3160 def d_V2x_inst_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3161
3162     r = np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
3163     eX = (x2 - x1) / (r)
3164     eY = (y2 - y1) / (r)
3165
3166     a2x = (
3167         (eX * S) / (r * r)
3168         + (
3169             eX
3170             * S
3171             * (
3172                 3 * eX * eX
3173                 - 1
3174                 - 3 * S * r * eY * eY
3175                 - (2 * (3 * eX * eX - 2)) * eta * r * r
3176             )
3177             * V1x
3178             * V1x

```



```

3179     )
3180     / (r * r * (-eta * r * r + 1))
3181     + (
3182         eX
3183         * S
3184         * (
3185             3 * S * r * eY * eY
3186             + 3 * eY * eY
3187             - 1
3188             - (6 * eY * eY - 1) * eta * r * r
3189         )
3190         * V1y
3191         * V1y
3192     )
3193     / (r * r * (-eta * r * r + 1))
3194     + (
3195         3
3196         * S
3197         * eY
3198         * (
3199             eta * r * r * (-4 * eX * eX + 1)
3200             + S * r * (2 * eX * eX - 1)
3201             + 2 * eX * eX
3202         )
3203         * V1x
3204         * V1y
3205     )
3206     / (r * r * (-eta * r * r + 1))
3207     - (3 * eY * eY * (S - r) * eX * V2x * V2x)
3208     / (r * r * (-eta * r * r + 1))
3209     + (3 * (S - r) * eY * eY * eX * V2y * V2y)
3210     / (r * r * (-eta * r * r + 1))
3211     + (3 * (S - r) * eY * (2 * eX * eX - 1) * V2x * V2y)
3212     / (r * r * (-eta * r * r + 1))
3213     + (eY * eY * eX * V1x * V2x) / (eta * r * r * r)
3214     - (
3215         S
3216         * (
3217             eta * eta * r * r * r
3218             - S * r * r * (eX * eX + 1) * eta
3219             + S * eX * eX
3220         )
3221         * eY
3222         * V1x
3223         * V2y
3224     )
3225     / (eta * r * r * r * (-eta * r * r + 1))
3226     - ((eX * eX + eta * (eY * eY * r * r - S * r)) * eY * V1y * V2x)
3227     / (eta * r * r * r * (-eta * r * r + 1))
3228     - ((-S * eX * eX + eta * r) * eX * S * V1y * V2y) / (eta * r * r * r)
3229 )
3230
3231 return a2x
3232
3233
3234 # Aceleração da partícula 2 no eixo y campo simétrico
3235 def d_V2y_inst_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3236
3237     r = np.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2)
3238     eX = (x2 - x1) / (r)

```

```

3239     eY = (y2 - y1) / (r)
3240
3241     a2y = (
3242         (eY * S) / (r * r)
3243         + (
3244             (
3245                 3 * eX * eX
3246                 - 1
3247                 + 3 * S * r * eX * eX
3248                 - eta * r * r * (6 * eX * eX - 1)
3249             )
3250             * S
3251             * eY
3252             * V1x
3253             * V1x
3254         )
3255         / (r * r * (-eta * r * r + 1))
3256         + (
3257             (
3258                 3 * eY * eY
3259                 - 1
3260                 - 3 * S * r * eX * eX
3261                 - 2 * eta * r * r * (-2 * eX * eX + eY * eY)
3262             )
3263             * eY
3264             * S
3265             * V1y
3266             * V1y
3267         )
3268         / (r * r * (-eta * r * r + 1))
3269         + (
3270             3
3271             * eX
3272             * (
3273                 eta * r * r * (-4 * eY * eY + 1)
3274                 + S * r * (2 * eY * eY - 1)
3275                 + 2 * eY * eY
3276             )
3277             * S
3278             * V1x
3279             * V1y
3280         )
3281         / (r * r * (-eta * r * r + 1))
3282         + (3 * (S - r) * eX * eX * eY * V2x * V2x)
3283         / (r * r * (-eta * r * r + 1))
3284         - (3 * (S - r) * eX * eX * eY * V2y * V2y)
3285         / (r * r * (-eta * r * r + 1))
3286         + (3 * eX * (S - r) * (2 * eY * eY - 1) * V2x * V2y)
3287         / (r * r * (-eta * r * r + 1))
3288         - ((-S * eY * eY + eta * r) * eY * S * V1x * V2x) / (eta * r * r * r)
3289         - (eX * (eta * r * (eX * eX * r - S) + eY * eY) * V1x * V2y)
3290         / (eta * r * r * r * (-eta * r * r + 1))
3291         - (
3292             S
3293             * (
3294                 eta * eta * r * r * r
3295                 - S * r * r * (eY * eY + 1) * eta
3296                 + S * eY * eY
3297             )
3298             * eX

```

```

3299         * V1y
3300         * V2x
3301     )
3302     / (eta * r * r * r * (-eta * r * r + 1))
3303     + eX * eX * eY * V1y * V2y / (eta * r * r * r)
3304 )
3305
3306     return a2y
3307
3308
3309 # Equações instantâneas lineares para o campo simétrico (tr -> t, ta -> t)
3310
3311 # Aceleração da partícula 1 no eixo x linear campo simétrico
3312 def d_V1x_inst_linear_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3313
3314     a1x = (
3315         -S
3316         * (x2 - x1)
3317         / (((x2 - x1) ** 2 + (y2 - y1) ** 2) ** (3.0 / 2.0) * eta)
3318     )
3319
3320     return a1x
3321
3322
3323 # Aceleração da partícula 2 no eixo x linear campo simétrico
3324 def d_V1y_inst_linear_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3325
3326     a1y = (
3327         -S
3328         * (y2 - y1)
3329         / (((x2 - x1) ** 2 + (y2 - y1) ** 2) ** (3.0 / 2.0) * eta)
3330     )
3331
3332     return a1y
3333
3334
3335 # Aceleração da partícula 2 no eixo x linear campo simétrico
3336 def d_V2x_inst_linear_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3337
3338     a2x = S * (x2 - x1) / ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** (3.0 / 2.0)
3339
3340     return a2x
3341
3342 # Aceleração da partícula 2 no eixo y linear campo simétrico
3343 def d_V2y_inst_linear_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3344
3345     a2y = S * (y2 - y1) / ((x2 - x1) ** 2 + (y2 - y1) ** 2) ** (3.0 / 2.0)
3346
3347     return a2y
3348
3349
3350 # ### Modelos de Runge-Kutta para os campos vetorias com campo simétrico ###
3351
3352 # Runge-Kutta 4 ordem para o campo simétrico instantâneo - Aproximação de ordem
3353 # zero do campo
3354 def rk4_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, step):
3355
3356     x1_1 = V1x * step
3357     y1_1 = V1y * step
3358     x2_1 = V2x * step

```

```

3358     y2_1 = V2y * step
3359
3360     V1x_1 = d_V1x_inst_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y) * step
3361     V1y_1 = d_V1y_inst_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y) * step
3362     V2x_1 = d_V2x_inst_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y) * step
3363     V2y_1 = d_V2y_inst_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y) * step
3364
3365     x1_k = x1 + x1_1 * 0.5
3366     y1_k = y1 + y1_1 * 0.5
3367     x2_k = x2 + x2_1 * 0.5
3368     y2_k = y2 + y2_1 * 0.5
3369     V1x_k = V1x + V1x_1 * 0.5
3370     V1y_k = V1y + V1y_1 * 0.5
3371     V2x_k = V2x + V2x_1 * 0.5
3372     V2y_k = V2y + V2y_1 * 0.5
3373
3374     x1_2 = V1x_k * step
3375     y1_2 = V1y_k * step
3376     x2_2 = V2x_k * step
3377     y2_2 = V2y_k * step
3378
3379     V1x_2 = (
3380         d_V1x_inst_sim(
3381             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3382         )
3383         * step
3384     )
3385     V1y_2 = (
3386         d_V1y_inst_sim(
3387             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3388         )
3389         * step
3390     )
3391     V2x_2 = (
3392         d_V2x_inst_sim(
3393             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3394         )
3395         * step
3396     )
3397     V2y_2 = (
3398         d_V2y_inst_sim(
3399             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3400         )
3401         * step
3402     )
3403
3404     x1_k = x1 + x1_2 * 0.5
3405     y1_k = y1 + y1_2 * 0.5
3406     x2_k = x2 + x2_2 * 0.5
3407     y2_k = y2 + y2_2 * 0.5
3408
3409     V1x_k = V1x + V1x_2 * 0.5
3410     V1y_k = V1y + V1y_2 * 0.5
3411     V2x_k = V2x + V2x_2 * 0.5
3412     V2y_k = V2y + V2y_2 * 0.5
3413
3414     x1_3 = V1x_k * step
3415     y1_3 = V1y_k * step
3416     x2_3 = V2x_k * step
3417     y2_3 = V2y_k * step

```

```
3418
3419     V1x_3 = (
3420         d_V1x_inst_sim(
3421             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3422         )
3423         * step
3424     )
3425     V1y_3 = (
3426         d_V1y_inst_sim(
3427             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3428         )
3429         * step
3430     )
3431     V2x_3 = (
3432         d_V2x_inst_sim(
3433             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3434         )
3435         * step
3436     )
3437     V2y_3 = (
3438         d_V2y_inst_sim(
3439             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3440         )
3441         * step
3442     )
3443
3444     x1_k = x1 + x1_3
3445     y1_k = y1 + y1_3
3446     x2_k = x2 + x2_3
3447     y2_k = y2 + y2_3
3448
3449     V1x_k = V1x + V1x_3
3450     V1y_k = V1y + V1y_3
3451     V2x_k = V2x + V2x_3
3452     V2y_k = V2y + V2y_3
3453
3454     x1_4 = V1x_k * step
3455     y1_4 = V1y_k * step
3456     x2_4 = V2x_k * step
3457     y2_4 = V2y_k * step
3458
3459     V1x_4 = (
3460         d_V1x_inst_sim(
3461             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3462         )
3463         * step
3464     )
3465     V1y_4 = (
3466         d_V1y_inst_sim(
3467             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3468         )
3469         * step
3470     )
3471     V2x_4 = (
3472         d_V2x_inst_sim(
3473             S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3474         )
3475         * step
3476     )
3477     V2y_4 = (
```

```

3478     d_V2y_inst_sim(
3479         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3480     )
3481     * step
3482 )
3483
3484 x1 = x1 + (x1_1 + 2 * (x1_2 + x1_3) + x1_4) / 6
3485 y1 = y1 + (y1_1 + 2 * (y1_2 + y1_3) + y1_4) / 6
3486 x2 = x2 + (x2_1 + 2 * (x2_2 + x2_3) + x2_4) / 6
3487 y2 = y2 + (y2_1 + 2 * (y2_2 + y2_3) + y2_4) / 6
3488
3489 V1x = V1x + (V1x_1 + 2 * (V1x_2 + V1x_3) + V1x_4) / 6
3490 V1y = V1y + (V1y_1 + 2 * (V1y_2 + V1y_3) + V1y_4) / 6
3491 V2x = V2x + (V2x_1 + 2 * (V2x_2 + V2x_3) + V2x_4) / 6
3492 V2y = V2y + (V2y_1 + 2 * (V2y_2 + V2y_3) + V2y_4) / 6
3493
3494 return step, x1, y1, x2, y2, V1x, V1y, V2x, V2y
3495
3496
3497 # Aplicação do método Runge Kutta para o campo simétrico instantâneo - Aproxima
    ção de ordem zero
3498 def rk4_method_inst_sim(
3499     path, n, m, f, t, dt, S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, count=10
3500 ):
3501     contador = 0
3502     with open(path, "w") as outfile:
3503         outfile.write(f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n")
3504         for _ in tqdm(range(1, m + 1, 1)):
3505             contador += 1
3506             for _ in range(1, n + 1, 1):
3507                 t += dt
3508                 dt_atual, x1, y1, x2, y2, V1x, V1y, V2x, V2y = rk4_sim(
3509                     S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, dt
3510                 )
3511                 dt = dt_atual
3512                 if contador == count:
3513                     outfile.write(
3514                         f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n"
3515                     )
3516                     contador = 0
3517                 if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (
3518                     np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)) < 2
3519                 ):
3520                     break
3521
3522                 vx = V2x - V1x
3523                 vy = V2y - V1y
3524                 rr = np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
3525                 ex = (x2 - x1) / (rr)
3526                 ey = (y2 - y1) / (rr)
3527                 DT = (-vx * ey + vy * ex) / (rr)
3528                 dt = (2.0 * np.pi) / (DT * n) / f
3529                 if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (rr < 2):
3530                     break
3531
3532
3533 # Campo para calcular no tempo de retardo e avançado - Aproximação de ordem 1
3534 def campo_w_sim(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y):
3535
3536     tr = -np.sqrt((x2 - x1) ** 2.0 + (y2 - y1) ** 2)

```

```

3537     _, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr = rk4_sim(
3538         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, tr
3539     )
3540
3541     A1x_tr = d_V1x_inst_linear_sim(
3542         S, eta, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr
3543     )
3544     A1y_tr = d_V1y_inst_linear_sim(
3545         S, eta, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr
3546     )
3547     A2x_tr = d_V2x_inst_linear_sim(
3548         S, eta, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr
3549     )
3550     A2y_tr = d_V2y_inst_linear_sim(
3551         S, eta, x1_tr, y1_tr, x2_tr, y2_tr, V1x_tr, V1y_tr, V2x_tr, V2y_tr
3552     )
3553
3554     ta = np.sqrt((x2 - x1) ** 2.0 + (y2 - y1) ** 2)
3555
3556     _, x1_ta, y1_ta, x2_ta, y2_ta, V1x_ta, V1y_ta, V2x_ta, V2y_ta = rk4_sim(
3557         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, ta
3558     )
3559
3560     A1x_ta = d_V1x_inst_linear_sim(
3561         S, eta, x1_ta, y1_ta, x2_ta, y2_ta, V1x_ta, V1y_ta, V2x_ta, V2y_ta
3562     )
3563     A1y_ta = d_V1y_inst_linear_sim(
3564         S, eta, x1_ta, y1_ta, x2_ta, y2_ta, V1x_ta, V1y_ta, V2x_ta, V2y_ta
3565     )
3566     A2x_ta = d_V2x_inst_linear_sim(
3567         S, eta, x1_ta, y1_ta, x2_ta, y2_ta, V1x_ta, V1y_ta, V2x_ta, V2y_ta
3568     )
3569     A2y_ta = d_V2y_inst_linear_sim(
3570         S, eta, x1_ta, y1_ta, x2_ta, y2_ta, V1x_ta, V1y_ta, V2x_ta, V2y_ta
3571     )
3572
3573     A1x = f_a1x_tr_av(
3574         S,
3575         eta,
3576         x1,
3577         y1,
3578         x2_tr,
3579         y2_tr,
3580         x2_ta,
3581         y2_ta,
3582         V1x,
3583         V1y,
3584         V2x_tr,
3585         V2y_tr,
3586         V2x_ta,
3587         V2y_ta,
3588         A2x_tr,
3589         A2y_tr,
3590         A2x_ta,
3591         A2y_ta,
3592     )
3593     A1y = f_a1y_tr_av(
3594         S,
3595         eta,
3596         x1,

```

```
3597     y1,
3598     x2_tr,
3599     y2_tr,
3600     x2_ta,
3601     y2_ta,
3602     V1x,
3603     V1y,
3604     V2x_tr,
3605     V2y_tr,
3606     V2x_ta,
3607     V2y_ta,
3608     A2x_tr,
3609     A2y_tr,
3610     A2x_ta,
3611     A2y_ta,
3612 )
3613 A2x = f_a2x_tr_av(
3614     S,
3615     x1_tr,
3616     y1_tr,
3617     x1_ta,
3618     y1_ta,
3619     x2,
3620     y2,
3621     V1x_tr,
3622     V1y_tr,
3623     V1x_ta,
3624     V1y_ta,
3625     V2x,
3626     V2y,
3627     A1x_tr,
3628     A1y_tr,
3629     A1x_ta,
3630     A1y_ta,
3631 )
3632 A2y = f_a2y_tr_av(
3633     S,
3634     x1_tr,
3635     y1_tr,
3636     x1_ta,
3637     y1_ta,
3638     x2,
3639     y2,
3640     V1x_tr,
3641     V1y_tr,
3642     V1x_ta,
3643     V1y_ta,
3644     V2x,
3645     V2y,
3646     A1x_tr,
3647     A1y_tr,
3648     A1x_ta,
3649     A1y_ta,
3650 )
3651
3652     return A1x, A1y, A2x, A2y
3653
3654
3655 # Runge-Kutta 4 ordem com o tempo de retardo e avançado - Aproximação de ordem
3656     1 do campo
```



```

3656 def rk4_tr_av(S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, h):
3657
3658     x1_1 = V1x * h
3659     y1_1 = V1y * h
3660     x2_1 = V2x * h
3661     y2_1 = V2y * h
3662
3663     V1x_1, V1y_1, V2x_1, V2y_1 = campo_w_sim(
3664         S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
3665     )
3666     V1x_1 = V1x_1 * h
3667     V1y_1 = V1y_1 * h
3668     V2x_1 = V2x_1 * h
3669     V2y_1 = V2y_1 * h
3670
3671     x1_k = x1 + x1_1 * 0.5
3672     y1_k = y1 + y1_1 * 0.5
3673     x2_k = x2 + x2_1 * 0.5
3674     y2_k = y2 + y2_1 * 0.5
3675     V1x_k = V1x + V1x_1 * 0.5
3676     V1y_k = V1y + V1y_1 * 0.5
3677     V2x_k = V2x + V2x_1 * 0.5
3678     V2y_k = V2y + V2y_1 * 0.5
3679
3680     x1_2 = V1x_k * h
3681     y1_2 = V1y_k * h
3682     x2_2 = V2x_k * h
3683     y2_2 = V2y_k * h
3684
3685     V1x_2, V1y_2, V2x_2, V2y_2 = campo_w_sim(
3686         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3687     )
3688     V1x_2 = V1x_2 * h
3689     V1y_2 = V1y_2 * h
3690     V2x_2 = V2x_2 * h
3691     V2y_2 = V2y_2 * h
3692
3693     x1_k = x1 + x1_2 * 0.5
3694     y1_k = y1 + y1_2 * 0.5
3695     x2_k = x2 + x2_2 * 0.5
3696     y2_k = y2 + y2_2 * 0.5
3697     V1x_k = V1x + V1x_2 * 0.5
3698     V1y_k = V1y + V1y_2 * 0.5
3699     V2x_k = V2x + V2x_2 * 0.5
3700     V2y_k = V2y + V2y_2 * 0.5
3701
3702     x1_3 = V1x_k * h
3703     y1_3 = V1y_k * h
3704     x2_3 = V2x_k * h
3705     y2_3 = V2y_k * h
3706
3707     V1x_3, V1y_3, V2x_3, V2y_3 = campo_w_sim(
3708         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3709     )
3710     V1x_3 = V1x_3 * h
3711     V1y_3 = V1y_3 * h
3712     V2x_3 = V2x_3 * h
3713     V2y_3 = V2y_3 * h
3714
3715     x1_k = x1 + x1_3

```

```

3716     y1_k = y1 + y1_3
3717     x2_k = x2 + x2_3
3718     y2_k = y2 + y2_3
3719     V1x_k = V1x + V1x_3
3720     V1y_k = V1y + V1y_3
3721     V2x_k = V2x + V2x_3
3722     V2y_k = V2y + V2y_3
3723
3724     x1_4 = V1x_k * h
3725     y1_4 = V1y_k * h
3726     x2_4 = V2x_k * h
3727     y2_4 = V2y_k * h
3728
3729     V1x_4, V1y_4, V2x_4, V2y_4 = campo_w_sim(
3730         S, eta, x1_k, y1_k, x2_k, y2_k, V1x_k, V1y_k, V2x_k, V2y_k
3731     )
3732     V1x_4 = V1x_4 * h
3733     V1y_4 = V1y_4 * h
3734     V2x_4 = V2x_4 * h
3735     V2y_4 = V2y_4 * h
3736
3737     x1 = x1 + (x1_1 + 2 * (x1_2 + x1_3) + x1_4) / 6
3738     y1 = y1 + (y1_1 + 2 * (y1_2 + y1_3) + y1_4) / 6
3739     x2 = x2 + (x2_1 + 2 * (x2_2 + x2_3) + x2_4) / 6
3740     y2 = y2 + (y2_1 + 2 * (y2_2 + y2_3) + y2_4) / 6
3741     V1x = V1x + (V1x_1 + 2 * (V1x_2 + V1x_3) + V1x_4) / 6
3742     V1y = V1y + (V1y_1 + 2 * (V1y_2 + V1y_3) + V1y_4) / 6
3743     V2x = V2x + (V2x_1 + 2 * (V2x_2 + V2x_3) + V2x_4) / 6
3744     V2y = V2y + (V2y_1 + 2 * (V2y_2 + V2y_3) + V2y_4) / 6
3745
3746     return h, x1, y1, x2, y2, V1x, V1y, V2x, V2y
3747
3748
3749 # Aplicação do método Runge-Kutta com tempo de retardo e avançado Aproximação
    de ordem 1
3750 def rk4_method_tr_va(
3751     path, n, m, f, t, dt, S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y
3752 ):
3753
3754     with open(path, "w") as outfile:
3755         outfile.write(f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n")
3756         for _ in tqdm(range(1, m + 1, 1)):
3757
3758             for _ in range(1, n + 1, 1):
3759                 t += dt
3760                 dt_atual, x1, y1, x2, y2, V1x, V1y, V2x, V2y = rk4_tr_av(
3761                     S, eta, x1, y1, x2, y2, V1x, V1y, V2x, V2y, dt
3762                 )
3763                 dt = dt_atual
3764
3765                 outfile.write(
3766                     f"{t} {x1} {y1} {x2} {y2} {V1x} {V1y} {V2x} {V2y} \n"
3767                 )
3768
3769                 if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (
3770                     np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)) < 2
3771                 ):
3772                     break
3773
3774                 vx = V2x - V1x

```

```
3775     vy = V2y - V1y
3776     rr = np.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1))
3777     ex = (x2 - x1) / (rr)
3778     ey = (y2 - y1) / (rr)
3779     DT = (-vx * ey + vy * ex) / (rr)
3780     dt = (2.0 * np.pi) / (DT * n) / f
3781     if (np.sqrt(V2x ** 2 + V2y ** 2) > 0.8) or (rr < 2):
3782         break
```

Código E.2: func.py