



UNIVERSIDADE FEDERAL DE SÃO CARLOS  
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA  
DEPARTAMENTO DE FÍSICA

DAVI DE MELLO LUCERO

**O PROBLEMA FERMI-PASTA-ULAM-TSINGOU** : Equipartição de  
energia vista através de simulações computacionais

SÃO CARLOS/SP  
2021

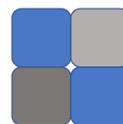
DAVI DE MELLO LUCERO

**O PROBLEMA FERMI-PASTA-ULAM-TSINGOU** : Equipartição de energia vista através de simulações computacionais

Trabalho de Conclusão de Curso, apresentado como requisito parcial para a obtenção de grau de Bacharel em Física, pela Universidade Federal de São Carlos.

Orientador: Prof . Dr. Pedro Augusto F. P. Moreira  
CCET/DF/UFSCAR

SÃO CARLOS/SP  
2021



## **Ata de Apresentação da Monografia da Disciplina Trabalho de Conclusão de Curso - Bacharelado para o curso Física - Integral da Universidade Federal de São Carlos**

Aos vinte e cinco dias do mês de novembro do ano de 2021, às 15 horas, virtualmente, reuniu-se a Banca Avaliadora da apresentação da Monografia na forma de regulamento da disciplina Trabalho de Conclusão de Curso - Bacharelado, composta pelos avaliadores: Prof. Dr. Vivaldo Leiria Campo Junior, Prof. Dr. Javier Fernando Ramos Caro e Prof. Dr. Pedro Augusto Franco Pinheiro Moreira, para proceder a avaliação da Monografia da disciplina Trabalho de Conclusão de Curso - Bacharelado, intitulada: "O problema Fermi-Pasta-Ulam-Tsingou: Equipartição de energia vista através de simulações computacionais." de autoria do aluno: Davi de Melo Lucero.

A sessão pública foi instalada pelo presidente da Banca, o qual, após a apresentação do aluno, passou a palavra aos demais avaliadores da Banca. Terminada a arguição, a Banca preencheu a ficha de avaliação da apresentação do Trabalho de Conclusão de Curso e atribuiu ao aluno as notas abaixo:

Avaliador 1 (Orientador): Prof. Dr. Pedro A. F. P. Moreira. Nota: 9,0 (nove).

Avaliador 2: Prof. Dr. Vivaldo Leiria Campo Junior. Nota: 8,0 (oito).

Avaliador 3: Prof. Dr. Javier Fernando Ramos Caro. Nota: 9,0 (nove)

De acordo com o Regulamento do Trabalho de Conclusão de Curso - Bacharelado, o aluno foi aprovado com a nota 8,7 (oito virgula sete).

Sem sugestão de correções ( )      Com sugestão de correções (X)

Nada mais havendo a tratar, foi encerrada a sessão e para constar eu, Pedro Augusto Franco Pinheiro Moreira, presidente da Banca Examinadora, lavrei esta Ata, que vai assinada por mim e pelos demais membros.

São Carlos, 25 de novembro de 2021.

*Pedro Augusto F. Moreira*

*Vivaldo Leiria Campo Junior*

*“Ali onde outros fracassaram, não fracassarei.”*  
*(Júlio Verne)*

## RESUMO

A equipartição de energia é um dos resultados mais fundamentais da Mecânica Estatística, surgindo nos mais diversos sistemas físicos. Com o intuito de explorar este conceito, dois experimentos computacionais do paradoxo Fermi-Pasta-Ulam-Tsingou foram implementados neste trabalho. Nos experimentos, há uma situação na qual acontece a equipartição de energia e, em outro, em que ela não se realiza. A segunda situação levou ao conhecido paradoxo de Fermi-Pasta-Ulam-Tsingou, com grandes implicações para o desenvolvimento da Física Computacional, Mecânica Estatística, entre outros campos. Os experimentos são realizados através de uma implementação numérica elaborada por meio de um computador pessoal e outra através de uma ferramenta computacional disponível na internet, bastante utilizada pela comunidade científica. As implementações são acompanhadas de uma análise matemática, utilizando-se modos normais.

**Palavras-chave:** Equipartição de energia. Paradoxo de Fermi-Pasta-Ulam-Tsingou. Simulação numérica. Dinâmica molecular. Física computacional.

## ABSTRACT

The equipartition of energy is one of the most fundamental results of Statistical Mechanics, appearing in a broad range of physical systems. In order to explore this concept, two computational experiments of the Fermi-Pasta-Ulam-Tsingou paradox were implemented. In one of the experiments the equipartition of energy happens and the other one not. The latter experiment leads to the known Fermi-Pasta-Ulam-Tsingou Paradox, with great implications to development of Computational Physics, Statistical Mechanics amongst other fields. The experiments are carried out through a home-made numerical implementation and a computational research tool available on internet and usually adopted by the scientific community. There is a mathematical analysis, using normal modes, to support the computational experiments.

**Keywords:** Equipartition of energy. Fermi-Pasta-Ulam-Tsingou paradox. Numerical simulation. Molecular dynamics. Computational physics.

## LISTA DE FIGURAS

Figura 1 – Esquema do problema FPUT para forças de interação linear. As partículas nos extremos estão fixas às paredes. . . . .	21
Figura 2 – Gráfico das energias armazenadas nos cinco primeiros modos normais (para uma energia inicial alta). . . . .	30
Figura 3 – Gráfico da energia média final associada a cada modo normal . . . . .	31
Figura 4 – Gráfico das energias armazenadas nos cinco primeiros modos normais (para uma energia inicial baixa). . . . .	31
Figura 5 – Gráfico das energias armazenadas nos dois primeiros modos normais (para um tempo de iteração alto). . . . .	32
Figura 6 – Gráfico dos picos do primeiro modo normal em função do tempo, gerando uma função “envelope”, representada pela linha azul. A quantidade de energia alocada no primeiro modo normal em relação à energia total é sinalizada pela linha vermelha com pontos. . . . .	33
Figura 7 – Representação das simulações de dinâmica molecular com dois passos temporais diferentes, mostrando as energias armazenadas nos cinco primeiros modos normais (para uma recorrência de primeira ordem). . . . .	35
Figura 8 – Representação das simulações de dinâmica molecular com dois passos temporais diferentes, mostrando as energias armazenadas nos dois primeiros modos normais (para uma recorrência de segunda ordem). . . . .	36

## LISTA DE TABELAS

Tabela 1 – Tabela das frequências para a solução numérica, em Hz. . . . .	32
Tabela 2 – Tabela de frequências, em Hz, para dois passos temporais diferentes e valores relativos percentuais $\Delta f_t$ comparados aos da solução numérica, sendo o subscrito $t$ indicativo do passo temporal 0,1 ou 0,5. . . . .	34

## LISTA DE SIGLAS

FPUT Fermi-Pasta-Ulam-Tsingou

LAMMPS *Large-scale Atomic/Molecular Massively Parallel Simulator*

## LISTA DE SÍMBOLOS

$\vec{F}$	Força total
$m$	Massa
$\vec{a}$	Vetor aceleração
$\vec{p}$	Vetor momento
$t$	Tempo
$\vec{r}$	Vetor posição
$\vec{v}$	Vetor velocidade
$W$	Trabalho
$K$	Energia cinética
$V$	Energia potencial
$q$	Coordenada generalizada
$\mathcal{L}$	Lagrangiana
$\mathcal{H}$	Hamiltoniana
$\delta t$	Passo temporal ( <i>Timestep</i> )
$x_i$	Deslocamento de uma partícula de seu centro de equilíbrio
$\alpha$	Coefficiente não-linear quadrático
$\beta$	Coefficiente não-linear cúbico
$E$	Energia total
$\omega$	Frequência de um modo normal
$\omega_0$	Frequência fundamental
$K_b$	Constante de Boltzman
$T$	Temperatura

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	OBJETIVOS	13
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
1.2	ESTRUTURA DO TRABALHO	13
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>15</b>
2.1	MECÂNICA NEWTONIANA	15
2.1.1	Teorema da conservação de energia	16
2.2	MECÂNICA LAGRANGIANA	16
2.3	MÉTODO DAS DIFERENÇAS FINITAS	19
2.3.1	Algoritmo de Verlet	20
2.4	DESCRIÇÃO FORMAL DO PROBLEMA	21
2.5	MODOS NORMAIS	22
2.6	TEOREMA DA EQUIPARTIÇÃO DE ENERGIA	25
2.7	IMPLEMENTAÇÃO EM <i>Python</i>	27
2.8	IMPLEMENTAÇÃO NO LAMMPS	28
<b>3</b>	<b>RESULTADOS</b>	<b>30</b>
3.1	PACOTE COMPUTACIONAL	34
<b>4</b>	<b>CONCLUSÃO</b>	<b>37</b>
	<b>REFERÊNCIAS</b>	<b>38</b>

<b>Apêndices</b>	<b>40</b>
<b>APÊNDICE A Códigos em Python . . . . .</b>	<b>41</b>
<b>APÊNDICE B Códigos para o LAMMPS . . . . .</b>	<b>51</b>

## 1 INTRODUÇÃO

O paradoxo de FPU (Enrico Fermi, John R. Pasta, Stanislaw M. Ulam) resulta de uma análise computacional de um problema hipotético, em que a suposição não foi confirmada pelos resultados que foram obtidos. A análise do problema gerou um paradoxo que começaria a ser respondido somente 10 anos depois do relatório divulgando o resultado, o que auxiliou no desenvolvimento das teorias de sólitons e do caos. O intuito original era estudar como um sistema evolui para seu equilíbrio térmico. Para isso, decidiram simular uma corrente de partículas que interagem entre seus vizinhos apenas, tendo os extremos ligados à paredes fixas. As interações seriam forças elásticas, ou seja, dependentes do deslocamento relativo a seus centros de equilíbrio. As forças teriam um termo linear como a Força de Hooke e mais um termo não-linear, podendo este ser quadrático ou cúbico (FERMI; PASTA; ULAM, 1940; GALLAVOTTI, 2007).

Quando o caso foi estudado pela primeira vez em Los Alamos, Estados Unidos, em 1955, além dos três participantes que relataram o caso, dando origem ao nome do paradoxo, tivemos a participação de mais uma pessoa. Mary Tsingou contribuiu no estudo numérico, implementando o código que seria utilizado. Seu nome não é mencionado no relatório da época, mas, nos padrões atuais para coautoria, ela seria mencionada no artigo que poderia ser escrito. Sendo assim, atualmente, o paradoxo é denominado pela sigla FPUT (Fermi-Pasta-Ulam-Tsingou). Os resultados do relatório nunca vieram a ser publicados em um formato de artigo devido a morte prematura de Fermi, que gerou dúvidas de como ele, sendo um dos autores, assinaria o artigo (DAUXOIS, 2008).

A premissa inicial do paradoxo de FPUT consiste no Teorema da Equipartição de Energia. Assim, era esperado que a energia total do sistema fosse distribuída igualmente entre suas várias partículas. No caso em questão, a distribuição de energia entre as partículas pode ser descrita através dos modos normais de vibração das mesmas. Pretendia-se observar a distribuição uniforme de energia entre os diversos modos normais de vibração com o passar do tempo. Isso significaria que o sistema alcançou um equilíbrio térmico e seria uma exemplificação computacional do Teorema da Equipartição de Energia. Caso as forças entre as partículas fossem estritamente lineares, isso não ocorreria, pois a energia alocada em cada modo não conseguiria acessar outros modos. Imaginava-se que uma componente não-linear na força tornaria acessível qualquer modo de vibração, porém não foi o observado.

A princípio, foi observada a tendência do sistema de distribuir a energia. O primeiro modo de vibração, antes estimulado, perdeu energia ao longo do tempo, a qual começou a se alocar nos modos de energia mais baixos. Entretanto, por um descuido, deixaram a simulação decorrer por um tempo maior do que era planejado. Ao retornar ao laboratório

para corrigir tal erro, se depararam com um resultado inesperado. A energia, que supostamente deveria estar igualmente partilhada entre os modos de vibração, estava quase completamente alocada no primeiro modo de vibração. De fato, somente 3% da energia não estava presente no primeiro modo. Devido a esta observação, deixaram a simulação correr por ainda mais tempo. Notaram, então, que existia um ciclo, no qual a energia saía do primeiro modo de vibração, começava a se distribuir nos modos mais baixos, para, por fim, voltar quase que inteiramente para o primeiro modo de vibração. Contudo, em 2015, relatou-se que o sistema FPUT poderia termalizar - atingir equipartição de energia - pelo menos entre modos normais livres (interação entre três fônons) (ONORATO et al., 2015).

Neste trabalho, portanto, serão implementados dois experimentos computacionais, de forma a demonstrar que o paradoxo de FPUT pode ser simulado por meio dos métodos de dinâmica molecular atuais.

## 1.1 OBJETIVOS

### 1.1.1 Objetivo geral

Demonstrar que o paradoxo de FPUT pode ser simulado através de métodos atuais de dinâmica molecular, servindo de base para estudos mais profundos acerca do tema.

### 1.1.2 Objetivos específicos

- Estabelecer uma clara fundamentação teórica acerca da física do sistema em estudo;
- Calcular resultados fundamentais que servirão como uma base fundamental de comparação, utilizando um algoritmo em *Python*;
- Realizar a implementação para o estudo de caso em questão, utilizando ferramentas de dinâmica molecular;
- Comparar os resultados obtidos pelas duas metodologias e, por fim;
- Verificar a aplicabilidade da dinâmica molecular para o paradoxo FPUT.

## 1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em 4 seções. Nesta primeira seção, foi apresentada um relato acerca da história do paradoxo, mostrando a importância do trabalho de FPUT, assim como uma breve descrição acerca do sistema que o gerou. Além disso, nesta seção, foram apresentados os objetivos do presente trabalho. A fundamentação teórica é desenvolvida na seção 2, explorando conceitos de mecânica clássica e mecânica estatística e estabelecendo formalmente o paradoxo FPUT. Na seção 3, são apresentados os resultados obtidos, tanto por uma solução exclusivamente derivada de bibliotecas em *Python*, como por uma solução resultante de um programa de dinâmica molecular, com suas devidas

discussões. Por fim, a seção 4 contém as devidas conclusões e perspectivas para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 MECÂNICA NEWTONIANA

Desde o início dos estudos em Física, o campo da Mecânica foi amplamente explorado. Este campo engloba o estudo do movimento dos corpos materiais, tendo uma de suas grandes vertentes a Mecânica Clássica, abrangendo pesquisas e teorias até os tempos atuais. Dentre essas teorias, uma das mais conhecidas e influentes da Física, é apresentada pela primeira vez no livro *Philosophiae Naturalis Principia Mathematica*, escrito por Isaac Newton em 1687. A formulação original das leis foi alvo de críticas e é apresentado a seguir, como releitura, almejando evitar inconsistências lógicas e reforçar os seus significados físicos (PIRES, 2011).

**1ª Lei:** Dada uma partícula infinitamente distante de qualquer corpo, portanto uma partícula isolada, podemos definir um sistema de coordenadas, chamado **referencial inercial**, no qual a partícula descreve um movimento com velocidade vetorial constante. Vale elencar que:

- Uma partícula é usualmente definida como um corpo cujas dimensões mensuráveis são desprezíveis;
- É necessário definir, *a priori*, métodos de mensuração de tempo e espaço, para, conseqüentemente, se obter a velocidade e aceleração de uma partícula;
- É assumido na mecânica newtoniana um espaço euclidiano e uma relatividade galileana.

**2ª Lei:** O movimento de uma dada partícula, dado um referencial inercial, é governado pela **força total**  $\vec{F}$  a qual essa partícula está sujeita, sendo esta definida por

$$\vec{F} = m\vec{a} = \frac{d\vec{p}}{dt}, \quad (1)$$

$$\frac{d^2\vec{r}}{dt^2} = \frac{d\vec{v}}{dt} = \vec{a}, \quad (2)$$

em que  $m$  é a massa da partícula,  $\vec{a}$  sua aceleração,  $\vec{p}$  o seu momento,  $\vec{v}$  sua velocidade e  $\vec{r}$  sua posição. O vetor posição une a origem do referencial inercial à partícula. Se escolhermos uma partícula  $U$  no universo como referência, podemos atribuir a **massa** de qualquer outra partícula  $P$  ao observar a interação, expressa na terceira lei, de  $U$  com  $P$ .

**3ª Lei:** Dada a interação de duas partículas, as forças experimentadas pelas partículas tem magnitude iguais e direções opostas, sendo esta interação expressa por

$$\vec{F}_{12} = -\vec{F}_{21}. \quad (3)$$

### 2.1.1 Teorema da conservação de energia

Consideremos o trabalho  $W$  realizado por uma força externa em uma partícula ao longo de uma trajetória, definido pela integral de linha

$$W_{12} = \int_1^2 \vec{F} \cdot d\vec{s}, \quad (4)$$

$$= \int_1^2 m \frac{d\vec{v}}{dt} \cdot \vec{v} dt,$$

$$= \frac{m}{2} \int_1^2 \frac{d}{dt} v^2 dt,$$

$$W_{12} = \frac{mv_2^2}{2} - \frac{mv_1^2}{2}. \quad (5)$$

O termo  $mv^2/2$  define o que é chamado de *energia cinética*  $K$ . Portanto, o trabalho de uma força externa sob uma partícula pode ser definido pela variação de energia cinética da mesma.

As forças externas podem ser colocadas em duas grandes categorias, as **forças conservativas**, como a força gravitacional, e as **forças não conservativas**, como a força de atrito. Se uma força é conservativa, então o trabalho realizado pela força ao longo de qualquer trajetória que una os pontos 1 e 2 independe do caminho realizado. A integral 4 se resume a

$$W_{12} = \oint \vec{F} \cdot d\vec{s} = 0. \quad (6)$$

É demonstrável que, uma condição suficiente para a integral ser igual a zero, é

$$\vec{F} = -\nabla V(\vec{r}), \quad (7)$$

sendo  $V$  chamado de **energia potencial** (uma função que depende exclusivamente da posição). Substituindo a equação 7 na equação 4, chegamos em

$$W_{12} = V_1 - V_2, \quad (8)$$

e, utilizando a equação 5, temos como resultado

$$K_2 + V_2 = K_1 + V_1. \quad (9)$$

A equação acima determina o teorema da conservação de energia. Tal teorema afirma que, para uma partícula sujeita apenas a ação de forças conservativas, a energia total da partícula,  $K + V$ , é conservada.

## 2.2 MECÂNICA LAGRANGIANA

A partir da equação 1, é de se esperar que qualquer problema de mecânica seja resolvido ao se solucionar as  $N$  equações diferenciais de um dado sistema<sup>1</sup>. Entretanto,

---

<sup>1</sup>Para um sistema com  $N$  partículas.

apesar da formulação newtoniana ter a capacidade de descrever inúmeros sistemas, por vezes se torna inconveniente ao descrever sistemas em que existem **vínculos**. Vínculos, por sua vez, são limitações aos movimentos do sistema que existem *a priori* e que introduzem novas dificuldades à resolução da dinâmica de um sistema, sendo uma delas o acoplamento das equações diferenciais.

Se os vínculos podem ser expressos por

$$f(\vec{r}_1, \dots, \vec{r}_N, t) = 0, \quad (10)$$

então estes são ditos holônomos e permitem a introdução de coordenadas generalizadas. Um sistema de  $N$  partículas admite  $3N$  coordenadas independentes, sendo estes seus graus de liberdade. Se no sistema existem  $k$  vínculos holônomos, então existem, ao todo,  $3N - k$  graus de liberdade que podem ser expressos por novas variáveis independentes. Sendo assim, as variáveis  $\vec{r}_1, \dots, \vec{r}_N$  podem ser expressas por esse novo conjunto de variáveis independentes, pela transformação

$$\begin{aligned} \vec{r}_1 &= \vec{r}_1(q_1, q_2, \dots, q_{3N-k}, t) \\ &\vdots \\ \vec{r}_N &= \vec{r}_N(q_1, q_2, \dots, q_{3N-k}, t). \end{aligned} \quad (11)$$

A partir das coordenadas generalizadas  $q_N$  e suas derivadas temporais  $\dot{q}_N$ , podemos definir a Lagrangiana  $\mathcal{L}(\vec{q}, \dot{\vec{q}})$  em termos da energia cinética e potencial como

$$\mathcal{L} = K - V \quad (12)$$

e chegar na equação de movimento lagrangiana

$$\frac{d}{dt} \left( \frac{\partial \mathcal{L}}{\partial \dot{q}_k} \right) - \left( \frac{\partial \mathcal{L}}{\partial q_k} \right) = 0. \quad (13)$$

Considerando um sistema de coordenadas cartesianas e definindo  $K$  e  $V$  como<sup>2</sup>

$$\begin{aligned} K &= \sum_{i=1}^N \frac{\vec{p}_i^2}{2m_i}, \\ V &= \sum_i V_1(\vec{r}_i) + \sum_i \sum_{j>i} V_2(\vec{r}_i, \vec{r}_j) + \sum_i \sum_{j>i} \sum_{k>j} V_3(\vec{r}_i, \vec{r}_j, \vec{r}_k) + \dots, \end{aligned} \quad (14)$$

a equação 13 se torna

$$m_i \ddot{\vec{r}}_i = \vec{f}_i, \quad (15)$$

---

<sup>2</sup>Supõe-se que a energia potencial possa ser dividida em funções que dependam apenas das posições das partículas em si e das distâncias com seus possíveis pares, trios e assim por diante.

em que

$$\vec{f}_i = \nabla_{r_i} \mathcal{L} = -\nabla_{r_i} V = - \left( \frac{\partial V}{\partial x_i} \hat{x}_i + \frac{\partial V}{\partial y_i} \hat{y}_i + \frac{\partial V}{\partial z_i} \hat{z}_i \right) \quad (16)$$

é a força em dada partícula.

O momento generalizado  $p_k$  conjugado a  $q_k$  é definido por

$$p_k = \frac{\partial \mathcal{L}}{\partial \dot{q}_k}. \quad (17)$$

O Hamiltoniano é estritamente definido pela equação

$$\mathcal{H}(\vec{p}, \vec{q}) = \sum_k \dot{q}_k p_k - \mathcal{L}(\vec{q}, \dot{\vec{q}}), \quad (18)$$

onde é assumido que podemos escrever  $\dot{q}_k$  como uma função do momento  $p$ . Para os propósitos do paradoxo FPUT, que envolve um potencial  $V$  independente do tempo e das velocidades, o Hamiltoniano se reduz a energia total do sistema.

As equações de movimento na forma Hamiltoniana são dados como

$$\begin{aligned} \dot{q}_k &= \frac{\partial \mathcal{H}}{\partial p_k} \\ \dot{p}_k &= -\frac{\partial \mathcal{H}}{\partial q_k}. \end{aligned} \quad (19)$$

Para coordenadas cartesianas, o conjunto de equações 19 pode ser reescrito como

$$\begin{aligned} \dot{\vec{r}}_i &= \frac{\vec{p}_i}{m_i} \\ \dot{\vec{p}}_i &= -\nabla_{r_i} V = \vec{f}_i. \end{aligned} \quad (20)$$

Uma consequência das equações 13 e 17, é que um dado momento generalizado  $p_k$  pode ser conservado, ou seja,  $\dot{p}_k = 0$ . Para isso acontecer, é necessário que  $\mathcal{L}$ , e por consequência  $\mathcal{H}$ , seja independente da coordenada generalizada  $q_k$  correspondente. Vale ressaltar que, para qualquer arranjo de partículas, é possível escolher seis coordenadas generalizadas, em que suas mudanças correspondem a translações do centro de massa e rotações sob o centro de massa. Assim, as outras  $3N - 6$  coordenadas envolverão o movimento das partículas entre si. Se a energia potencial depende apenas da distância das partículas entre si e não temos nenhum campo externo aplicado, então  $V$ ,  $\mathcal{L}$  e  $\mathcal{H}$  são independentes destas seis coordenadas generalizadas. Assumindo que  $\mathcal{H}$  não dependa explicitamente do tempo, ou seja,  $\partial \mathcal{H} / \partial t = 0$ , a derivada total  $\dot{\mathcal{H}}$  pode ser escrita como

$$\frac{d\mathcal{H}}{dt} = \sum_k \left( \frac{\partial \mathcal{H}}{\partial q_k} \dot{q}_k + \frac{\partial \mathcal{H}}{\partial p_k} \dot{p}_k \right) = 0. \quad (21)$$

Logo, o Hamiltoniano é uma constante de movimento, sendo essa afirmação válida mesmo que tenha um potencial externo aplicado.

Outro ponto a ser notado é que, dada as equações de movimento, na forma lagrangiana ou na forma hamiltoniana, a derivada espacial do potencial é mencionada. Sendo assim, a resolução das equações (por meios computacionais) depende de  $V$  variar suavemente com a posição. Os casos em que ocorre a colisão de partículas (seja entre si ou com os limites do sistema) e que  $V$  varia de maneira descontínua, devem ser tratados explicitamente para lidar com as dinâmicas da partícula.

### 2.3 MÉTODO DAS DIFERENÇAS FINITAS

Um método padrão para a solução de equações diferenciais ordinárias tais quais a equação 15, é o método das diferenças finitas. Dada as posições, velocidades e informações dinâmicas relevantes num tempo  $t$ , são obtidas as mesmas informações num tempo posterior  $t + \delta t$ , com uma precisão arbitrária. As equações são resolvidas iterativamente, passo a passo.

A escolha do passo temporal, também conhecido como *timestep*,  $\delta t$  depende do método de solução mas, geralmente, é muito menor que o tempo da partícula deslocar o seu próprio comprimento<sup>3</sup>. No presente trabalho, dada a natureza adimensional das partículas, o passo temporal foi escolhido de forma que a simulação termine em um número razoável de iterações.

Na natureza, o tempo é uma variável contínua. Em computadores, a variável tempo é discreta - ocorre em passos. Para haver conservação da energia é necessário que haja uma continuidade temporal. Em um computador, aproximamos da realidade através da diminuição do passo temporal. Por um lado, passos temporais menores nos levam à conservação de energia melhores, mas também um número de iterações maiores, com processamentos mais longos, o que implica no aumento dos custos computacionais (TUCKERMAN, 2010).

Diferentes algoritmos se enquadram no método de diferenças finitas, entre eles o *predictor-corrector* e o *Runge-Kutta*, mas todos compartilham aspectos desejáveis para uma simulação de sucesso (LANDAU; PAEZ; BORDEIANU, 2015; BUTCHER, 2008), a saber:

- Rapidez e requisição de pouca memória;
- Viabilização do uso de um passo temporal  $\delta t$  grande;
- Possibilidade de replicar ao máximo a trajetória clássica;
- Satisfação das leis de conservação de energia e momento, além de ser reversível temporalmente;
- Metodologia simples e facilmente programável.

---

<sup>3</sup>A escolha do passo temporal baseando-se no tamanho da partícula é relevante quando o tamanho da partícula é levado em conta nas simulações.

A memória necessária para armazenar as posições, velocidades e assim por diante é relativamente pequena, se levarmos em consideração os computadores atuais, tornando o primeiro item pouco relevante para a escolha de algoritmos. É mais importante, portanto, que o algoritmo permita o uso de  $\delta t$  longos, dessa maneira, um dado período de simulação pode ser coberto por um número razoável de iterações. Em contrapartida, quanto maior o passo temporal, menos precisamente a solução obtida se aproximará da trajetória clássica. Ainda que nenhum algoritmo proveja uma solução exata para um tempo longo, é mais relevante que as trajetórias das partículas permaneçam na hiper-superfície apropriada de conservação de energia do espaço de fase. A conservação de energia é deteriorada à medida que o passo temporal aumenta, então todas as simulações envolvem uma troca entre economia e precisão, ou seja, um bom algoritmo permite o uso de um passo temporal<sup>4</sup> longo enquanto preserva uma conservação de energia adequada.

### 2.3.1 Algoritmo de Verlet

Pelos motivos expostos na seção anterior, a maior parte dos programas de dinâmica molecular utiliza uma variante do algoritmo inicialmente desenvolvido por Loup Verlet. Uma das maneiras de escrever tal algoritmo é na forma *velocity*-Verlet

$$\begin{aligned} v\left(t + \frac{1}{2}\delta t\right) &= v(t) + \frac{1}{2}\delta t a(t), \\ r(t + \delta t) &= r(t) + \delta t v\left(t + \frac{1}{2}\delta t\right), \\ v(t + \delta t) &= v\left(t + \frac{1}{2}\delta t\right) + \frac{1}{2}\delta t a(t + \delta t). \end{aligned} \quad (22)$$

No primeiro passo do algoritmo, as velocidades são avançadas para um valor intermediário, a partir da aceleração no tempo  $t$ . Com essas velocidades, as posições das partículas são propelas para o tempo  $t + \delta t$ . Neste momento, as forças agindo nas partículas são reavaliadas, permitindo o cálculo das acelerações nesse instante pela equação 15. Com isso, os valores finais das velocidades são calculados.

Esse método é numericamente estável, conveniente e simples. É exatamente reversível no tempo e, dado forças conservativas, garante a conservação de momento linear. Foi demonstrado que o método tem excelentes propriedades de conservação de energia, até mesmo para um passo temporal longo. No artigo “original”, as equações 22 foram escritas de maneiras levemente diferentes

$$\begin{aligned} r(t + \delta t) &= r(t) + \delta t v(t) + \frac{1}{2}\delta t^2 a(t) \\ v(t + \delta t) &= v(t) + \frac{1}{2}\delta t [a(t) + a(t + \delta t)]. \end{aligned} \quad (23)$$

---

<sup>4</sup>Para partículas leves, temperaturas altas ou potenciais que variam rapidamente, é necessário a utilização de um passo temporal mais curto.

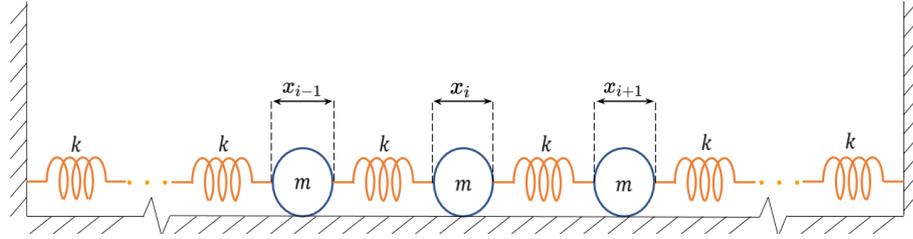


Figura 1 – Esquema do problema FPUT para forças de interação linear. As partículas nos extremos estão fixas às paredes.

Na prática, a velocidade ainda é incrementada em duas etapas, a não ser que os valores iniciais e finais da aceleração sejam desnecessariamente armazenados (VERLET, 1967; GEAR, 1971; SWOPE et al., 1982).

#### 2.4 DESCRIÇÃO FORMAL DO PROBLEMA

Até este momento nessa seção, foram abordados a teoria da Mecânica Clássica e algoritmos de solução numérica para equações diferenciais. Agora, apresentaremos o problema introduzido por FPUT formalmente no qual aplicaremos os conceitos introduzidos anteriormente para analisá-lo. Temos, portanto, um sistema unidimensional de  $N$  partículas com forças entre seus vizinhos, possuindo termos não-lineares. Esses termos podem ser considerados ora quadráticos, ora cúbicos. Sendo  $x_i$  o deslocamento da  $i$ -ésima partícula em relação a seu centro de equilíbrio, podemos representar o problema pela Figura 1 e, conseqüentemente, descrever a força de interação linear combinada com a interação quadrática como

$$F_i = m\ddot{x}_i = k_1 \{(x_{i+1} - x_i) + (x_{i-1} - x_i)\} + k_2 \{(x_{i+1} - x_i)^2 + (x_{i-1} - x_i)^2\}, \quad (24)$$

onde  $x_{i\pm 1} - x_i = \Delta x_i$  representa a distância entre duas partículas consecutivas e  $\ddot{x}_i$  é a aceleração da  $i$ -ésima partícula numa representação de Newton para a derivada segunda da posição em relação ao tempo. Neste problema, todas as partículas possuem a mesma massa  $m$ . Analogamente, podemos escrever o caso de interações lineares e cúbicas na forma

$$F_i = m\ddot{x}_i = k_1 \{(x_{i+1} - x_i) + (x_{i-1} - x_i)\} + k_2 \{(x_{i+1} - x_i)^3 + (x_{i-1} - x_i)^3\}. \quad (25)$$

Para obtermos as acelerações de cada partícula e conseguirmos a descrição cinética do movimento, podemos considerar a razão da constante elástica do termo linear pela massa como a unidade e definir coeficientes  $\alpha$  e  $\beta$  para os termos quadráticos e cúbicos. Esses novos coeficientes definem o quanto esses termos afetam a dinâmica do sistema.

Sendo assim, temos as seguintes equações:

$$\ddot{x}_i = (x_{i+1} - x_i) + (x_{i-1} - x_i) + \alpha \left\{ (x_{i+1} - x_i)^2 + (x_{i-1} - x_i)^2 \right\} \quad (26)$$

$$\ddot{x}_i = (x_{i+1} - x_i) + (x_{i-1} - x_i) + \beta \left\{ (x_{i+1} - x_i)^3 + (x_{i-1} - x_i)^3 \right\} \quad (27)$$

Então, considerando-se uma força conservativa, podemos escrevê-la como  $F_i = -\partial V/\partial x_i$ , em que  $V$  é a energia potencial. Para o caso quadrático, a energia potencial será

$$V = \sum_{i=0}^N \frac{1}{2} \left\{ [(x_{i+1} - x_i)^2 + (x_{i-1} - x_i)^2] + \frac{\alpha}{3} [(x_{i+1} - x_i)^3 + (x_{i-1} - x_i)^3] \right\} \quad (28)$$

e, para o caso cúbico,

$$V = \sum_{i=0}^N \frac{1}{2} \left\{ [(x_{i+1} - x_i)^2 + (x_{i-1} - x_i)^2] + \frac{\beta}{4} [(x_{i+1} - x_i)^4 + (x_{i-1} - x_i)^4] \right\}. \quad (29)$$

Supondo que  $x_0$  e  $x_{N+1}$  são as paredes do sistema, a energia total do sistema, dependendo de cada caso, pode ser escrita como

$$E = \sum_{i=0}^N \left\{ \frac{p_i^2}{2m} + \frac{1}{2}(x_{i+1} - x_i)^2 + \frac{\alpha}{3}(x_{i+1} - x_i)^3 \right\} \quad (30)$$

e

$$E = \sum_{i=0}^N \left\{ \frac{p_i^2}{2m} + \frac{1}{2}(x_{i+1} - x_i)^2 + \frac{\beta}{4}(x_{i+1} - x_i)^4 \right\}, \quad (31)$$

para os casos quadrático e cúbico, respectivamente.

Se possuímos um sistema que realiza pequenas oscilações acopladas, podemos aproximar uma solução particular em que todas as partículas realizam movimentos harmônicos simples com a mesma frequência. A solução mais geral possível seria a soma dessas soluções particulares, chamadas de modos normais de vibração. Portanto, à princípio, sabendo as frequências de vibração e o deslocamento dos modos normais, obteríamos a solução do problema.

## 2.5 MODOS NORMAIS

Para obtermos a solução do problema, é preciso revisar os conceitos de modos normais. Suponhamos um sistema idêntico ao do paradoxo FPUT, exceto que, agora,

iremos considerar apenas termos lineares na força (ARFKEN; WEBER, 1999; BOYCE; DIPRIMA, 2012). A equação de movimento de cada partícula se resumirá a

$$m\ddot{x}_i = k(x_{i+1} + x_{i-1} - 2x_i). \quad (32)$$

Vamos propor uma solução do tipo

$$x_i(t) = A_i e^{j\omega t}, \quad (33)$$

onde  $j$  é o número imaginário tal que  $j^2 = -1$ . Inserindo-o na equação 32, temos que

$$-\omega^2 A_i = \frac{k}{m}(A_{i+1} + A_{i-1} - 2A_i). \quad (34)$$

Estabeleceremos que  $\omega_0^2 = \frac{k}{m}$ . Desta forma, podemos reescrever a equação 34 como

$$\frac{2\omega_0^2 - \omega^2}{\omega_0^2} = \frac{A_{i+1} + A_{i-1}}{A_i}. \quad (35)$$

Para um dado valor  $\omega$ , temos que o lado esquerdo da equação deve ser constante, pois  $\omega_0$  é um valor arbitrário. Portanto, para estar condizente com o lado esquerdo da equação, o lado direito também deve ser constante. Vamos supor, agora, que  $0 \leq \omega \leq 2\omega_0$ . Isto nos permite escrever a constante do lado esquerdo da equação como  $2 \cos(\theta)$ . Assim,

$$\cos(\theta) = \frac{A_{i+1} + A_{i-1}}{2A_i}, \quad (36)$$

$$2 \cos(\theta) = \frac{2\omega_0^2 - \omega^2}{\omega_0^2}. \quad (37)$$

Suponhamos que qualquer  $A_i$  possa ser escrito como  $A_i = B \cos(i\theta) + C \sin(i\theta)$ . Desejamos considerar desta forma para provar, por indução, que tal suposição é válida. Podemos determinar o valor de  $B$  através de  $A_0 \equiv B$ .  $C$ , por sua vez, pode ser determinado a partir de  $A_1 = B \cos(\theta) + C \sin(\theta)$ . Precisamos, por fim, determinar  $A_{i+1}$ . Da equação 36, temos que

$$A_{i+1} = 2A_i \cos(\theta) - A_{i-1}, \quad (38)$$

$$\begin{aligned} A_{i+1} &= 2 \cos(\theta) \{B \cos(i\theta) + C \sin(i\theta)\} - \{B \cos((i-1)\theta) + C \sin((i-1)\theta)\}, \\ &= B \{2 \cos(\theta) \cos(i\theta) - \cos((i-1)\theta)\} + C \{2 \cos(\theta) \sin(i\theta) - \sin((i-1)\theta)\}, \\ &= B \{\cos(\theta) \cos(i\theta) - \sin(i\theta) \sin(\theta)\} + C \{\cos(\theta) \sin(i\theta) + \sin(\theta) \cos(i\theta)\}, \end{aligned} \quad (39)$$

o que nos leva a

$$A_{i+1} = B \cos((i+1)\theta) + C \sin((i+1)\theta). \quad (40)$$

Portanto, como  $A_i$  é válido para  $A_0, A_1, A_{i+1}$ , provamos, por indução, que ele é válido para qualquer  $A_i$  e obtemos a solução

$$x_i = \{B \cos(i\theta) + C \sin(i\theta)\} e^{j\omega t}, \quad (41)$$

desde que a suposição sobre a frequência seja respeitada. Para o estudo em questão, possuímos duas condições de contorno, relativas às paredes fixas do sistema. De  $x_0(t) = 0$ , temos que  $B = 0$ . De  $x_{N+1}(t) = 0$ , temos que  $\theta = \frac{m\pi}{N+1}$ , para uma solução geral não trivial, sendo  $m$  um número inteiro.

Desta maneira, resta apenas o termo senoidal para descrever o movimento da partícula: para cada modo normal, podemos associar uma curva senoidal, que limitará os deslocamentos máximos de cada partícula em relação ao seu centro de equilíbrio, assim como fixará a direção do deslocamento relativo das mesmas. Encontrados os  $\theta$  possíveis, podemos descobrir as frequências associadas a cada modo. Da equação 37, podemos escrever

$$\begin{aligned} \omega^2 &= 2\omega_0^2(1 - \cos(\theta)), \\ &= 2\omega_0^2 \left\{ 1 - \cos \left( \frac{m\pi}{N+1} \right) \right\}, \\ &= 2\omega_0^2 \left\{ 2 \sin^2 \left( \frac{m\pi}{2(N+1)} \right) \right\}, \end{aligned} \quad (42)$$

e

$$\omega = 2\omega_0 \left\{ \sin \left( \frac{m\pi}{2(N+1)} \right) \right\}. \quad (43)$$

O movimento da partícula é dado como a soma de todas as soluções particulares (modos normais). Como estamos levando em consideração um número discreto de partículas, conseqüentemente teremos uma quantidade discreta de modos. Logo, a solução mais geral, em que tomamos a parte real da exponencial, será

$$x_i(t) = \sum_{m=1}^N C_m \sin \left( \frac{im\pi}{N+1} \right) \cos(\omega_m t), \quad (44)$$

e

$$\omega_m = 2\omega_0 \sin \left( \frac{m\pi}{2(N+1)} \right). \quad (45)$$

A energia de cada modo se divide entre potencial e cinética. Portanto, a energia total, que é a soma da energia alocada em cada modo (descartando termos de ordem maior), será dada por

$$E_{total} = \sum_{i=1}^N \frac{\omega_i^2 A_i^2 + \dot{A}_i^2}{2}. \quad (46)$$

Note que  $A_i$  se refere à amplitude da oscilação de um dado modo normal e  $\dot{A}_i$  se refere à amplitude da velocidade deste mesmo modo. A questão principal, portanto, será como obter essas amplitudes para todo o tempo estabelecido. Se, para cada partícula, sabemos a sua posição, podemos aplicar uma transformada de Fourier no conjunto de posições das partículas e obter, assim, a amplitude de oscilação de todos os modos normais.

De maneira análoga, se aplicarmos uma transformada de Fourier no conjunto de velocidades de todas as partículas, obteremos a amplitude de velocidade de todos os modos normais. Portanto, se criarmos um programa que solucione as equações de movimento e retorne as posições e velocidades para qualquer tempo, podemos obter as amplitudes de oscilação da posição e da velocidade para cada frequência normal e, assim, calcular tanto a energia cinética alocada em cada modo normal, quanto a energia total.

## 2.6 TEOREMA DA EQUIPARTIÇÃO DE ENERGIA

Suponhamos que um sistema esteja isolado, ou seja, que a energia interna dele seja constante ao longo do tempo. Para um determinado tempo, é esperado que este sistema atinja equilíbrio térmico, isto é, sozinho, o sistema está permitido a assumir determinadas características intrínsecas. Estas características podem ser, no caso de um gás, a própria pressão ou temperatura. Já para o sistema FPUT, uma característica relevante é a contribuição média de cada termo do Hamiltoniano para a energia do sistema.

A mecânica estatística clássica fornece um resultado muito útil quando avaliamos justamente essa contribuição média para a energia interna do sistema, baseada em duas premissas. A energia mencionada na equação 46 pode ser função das coordenadas generalizadas e dos momentos correspondentes

$$E = E(q_1, \dots, q_N, p_1, \dots, p_N). \quad (47)$$

Dependendo de cada situação, a energia pode ser aditiva, a qual é separável em funções de uma variável apenas, como

$$E = \varepsilon_i(p_i) + E'(q_1, \dots, p_N), \quad (48)$$

onde  $\varepsilon_i$  é parte da energia dependente apenas da variável  $p_i$  (podendo ser uma coordenada ou momento) e o restante da energia total  $E'$  não depende desta variável. A energia também pode se separar em funções quadráticas, seguindo a forma

$$\varepsilon_i(p_i) = a_i p_i^2, \quad (49)$$

onde  $a_i$  é uma constante. Essas duas características são utilizadas pela mecânica estatística para derivar o Teorema da Equipartição de Energia. Vale notar que o sistema FPUT segue essas duas premissas: ele se divide em  $2N$  funções quadráticas de uma variável apenas, sendo, assim, passível da aplicação do teorema.

Com o nosso sistema em equilíbrio térmico, este assume uma temperatura bem definida, a qual se relaciona com a energia pela constante de Boltzman,  $K_b$ . Esse sistema também permite que seja utilizado o *ensemble* canônico para calcular o valor médio esperado conforme

$$\langle \varepsilon_i(p_i) \rangle = \frac{\int_{-\infty}^{\infty} \varepsilon_i(p_i) e^{-\beta E} dq_1 \dots dp_f}{\int_{-\infty}^{\infty} e^{-\beta E} dq_1 \dots dp_f}, \quad (50)$$

em que  $\beta$  é relacionado à temperatura por  $\beta = (K_b T)^{-1}$ . Ao utilizar as características da energia, podemos continuar desenvolvendo a equação 50 como

$$\langle \varepsilon_i(p_i) \rangle = \frac{\int_{-\infty}^{\infty} \varepsilon_i(p_i) e^{-\beta \varepsilon_i(p_i)} dp_i \int_{-\infty}^{\infty} e^{-\beta E'} dq_1 \dots dp_f}{\int_{-\infty}^{\infty} e^{-\beta \varepsilon_i(p_i)} dp_i \int_{-\infty}^{\infty} e^{-\beta E'} dq_1 \dots dp_f}, \quad (51)$$

$$\langle \varepsilon_i(p_i) \rangle = a_i \frac{\int_{-\infty}^{\infty} p_i^2 e^{-\beta a_i p_i^2} dp_i}{\int_{-\infty}^{\infty} e^{-\beta a_i p_i^2} dp_i}, \quad (52)$$

e

$$\langle \varepsilon_i(p_i) \rangle = \frac{1}{2} K_b T. \quad (53)$$

Portanto, cada termo quadrático do Hamiltoniano contribui com um valor médio de  $\frac{1}{2} K_b T$  para a energia média do sistema, quando o mesmo se encontrar em equilíbrio térmico. Vale notar que termos não quadráticos para osciladores não-harmônicos, desde que aditivos, estão sujeitos à valores médios similares aos correspondentes aos termos quadráticos. A energia interna média depende da potência que a posição e o momento conjugados são elevados na função da energia. Alguns casos não quadráticos são abordados pedagogicamente por Prasanth e co-autores (PRASANTH; NISHANTH; UDAYANANDAN, 2019; REIF, 2009).

Como demonstrado, se o sistema está em equilíbrio térmico, as coordenadas e momentos generalizados  $q_i$  e  $p_i$  deveriam contribuir na mesma proporção para a energia

total do sistema (se ambos forem termos quadráticos). Visando a solução com apenas termos lineares na força, da Seção 2.5, em equilíbrio térmico, as amplitudes e suas derivadas temporais,  $A_i$  e  $\dot{A}_i$ , deveriam contribuir na mesma quantidade para a energia total do sistema, por ambos serem termos quadráticos. Em outras palavras: a energia associada a cada modo normal de vibração deveria ser a mesma, em média.

Buscando esta expectativa da equipartição de energia de um sistema em equilíbrio térmico, analisaremos o problema FPUT, em sua totalidade, através de simulações numéricas. Exemplificaremos situações em que acontecem a equipartição de energia (como previsto pelo teorema apresentado) e uma situação específica que levou ao conhecido paradoxo FPUT, em que a equipartição de energia não acontece.

## 2.7 IMPLEMENTAÇÃO EM *Python*

Para o estudo do paradoxo FPUT, primeiramente foi escrito um código em *Python*, a fim de se obter os principais resultados. A escolha desta linguagem foi por ser facilmente programável, altamente documentada e possuir diversas bibliotecas disponíveis *online*. Para a solução das equações de movimento, consideremos a equação 24, em que as forças têm uma componente quadrada e que tenhamos  $N$  partículas. Podemos criar um vetor de tamanho  $2N$ , no qual os primeiros  $N$  termos representam o deslocamento relativo e os últimos  $N$  termos representam as velocidades. Assim, o sistema de  $N$  equações diferenciais de segunda ordem se torna um sistema de  $2N$  equações diferenciais de primeira ordem. Este sistema pode ser descrito da seguinte maneira, com  $i$  variando de 1 a  $N - 2$

$$\dot{x}_0 = x_N,$$

$$\dot{x}_i = x_{N+i},$$

$$\dot{x}_{N-1} = x_{2N-1},$$

$$\dot{x}_N = x_1 - 2x_0 + \alpha \{(x_1 - x_0)^2 + x_0^2\},$$

$$\dot{x}_{N+i} = x_{i+1} + x_{i-1} - 2x_i + \alpha \{(x_{i+1} - x_i)^2 + (x_{i-1} - x_i)^2\},$$

$$\dot{x}_{2N-1} = x_{N-2} - 2x_{N-1} + \alpha \{x_{N-1}^2 + (x_{N-2} - x_{N-1})^2\}. \quad (54)$$

O código A.1 (disponível no apêndice) foi utilizado para a resolução deste sistema de equações diferenciais acopladas e está devidamente comentado. As integrações numéricas

são feitas pela função *odeint* da biblioteca *scipy*, adequados para resolver sistemas na forma  $dy/dt = f(t,y)$  através do método de Adams (*predictor-corrector*) (VIRTANEN et al., 2020).

A hipótese inicial de FPUT era que, em condições de equilíbrio térmico (ou seja, dado um tempo de simulação suficientemente alto), o teorema de equipartição de energia ocorreria. No programa implementado, se tivermos uma interação não-linear forte ou uma quantidade inicial de energia consideravelmente alta, observaremos que a equipartição de energia acontecerá. Ao manipular as condições iniciais do sistema, portanto a energia total, conseguiremos replicar tanto um sistema em que ocorre a equipartição de energia entre os modos normais, quanto um sistema em que isso não ocorre, o qual gerou o paradoxo FPUT.

## 2.8 IMPLEMENTAÇÃO NO LAMMPS

A solução numérica da seção anterior é relevante e factível apenas para um número pequeno de partículas e passos de integrações, sendo aplicável quando não há grande preocupação com eficiência computacional. Pesquisadores, atualmente, necessitam lidar com número de partículas da ordem de milhares, bilhões e até cerca de  $10^{10}$  passos de integrações. Na Física, essas partículas geralmente representam átomos, núcleos e elétrons. Implementações computacionais que descrevem a interação entre esses entes e seus movimentos são chamados, em Física Computacional, de Dinâmica Molecular. Para a aprendizagem de uma ferramenta computacional do estado da arte da área, foi utilizado um pacote computacional bastante adotado pela comunidade científica (LANDAU; PAEZ; BORDEIANU, 2015).

Para se obter as posições e velocidades de cada partícula, podemos usar a abordagem de dinâmica molecular implementada no LAMMPS (*Large-scale Atomic/Molecular Massively Parallel Simulator*). Esse tipo de abordagem é bastante utilizada atualmente nos campos da Física, Química, Biologia e Engenharias, pois permite uma fácil visualização dos movimentos atômicos e, computacionalmente, há vários pacotes de dinâmica molecular disponíveis (PLIMPTON, 1995).

A dinâmica molecular, por sua vez, se baseia na possibilidade de descrever a trajetória de um sistema de partículas através da solução das Leis de Newton por meio de computadores (ou, de maneira equivalente, um sistema Hamiltoniano para mecânica clássica). No caso do LAMMPS, as integrais são resolvidas pelo algoritmo *velocity-Verlet* (NEWMAN, 2013; TUCKERMAN, 2010; GOLDSTEIN; POOLE; SAFKO, 2002).

Apesar da solução por dinâmica molecular se basear na solução da equação 24, como feito na seção anterior, não será necessário reescrever o conjunto de equação diferenciais, sendo necessário descrever apenas o sistema através do tipo de ligação entre partículas. Para o caso em questão, estamos considerando molas de constante igual que unem as

partículas, portanto temos apenas uma espécie de ligação. Em um outro tipo de situação, com escala mais aprofundada e características mais complexas, essas ligações descreveriam, por exemplo, ligações químicas usando potenciais adequados, os quais podemos encontrar publicados em revistas científicas (PINTO et al., 2014).

De maneira análoga à seção anterior, é necessário informar as condições iniciais do sistema. Entretanto, para se obter a solução correta através do LAMMPS, necessitamos adicionar duas partículas extras no sistema, para servirem como paredes. O LAMMPS é processado através de comandos fornecidos por um arquivo de entrada, chamado *script*. Uma descrição detalhada de cada comando, com seus argumentos possíveis, pode ser encontrada na documentação do LAMMPS, em inglês.

No caso, o *script* utilizado no LAMMPS é o código B.1 (detalhado no apêndice) que, ao ser processado, resulta em um arquivo com as posições e velocidades das partículas em cada iteração da simulação. Para o funcionamento do *script*, é necessário dois arquivos auxiliares. O primeiro está relacionado às condições iniciais do sistema, fornecendo a posição inicial de cada partícula e descrevendo como as partículas interagem entre si. Na biblioteca do LAMMPS temos alguns potenciais disponíveis para utilização, como por exemplo o de Lennard-Jones. Ainda assim, podemos fornecer um potencial específico, sendo o segundo arquivo necessário justamente para isto.

A necessidade da criação do arquivo relacionado às forças de interação surgiu do fato de nenhum modelo contido no LAMMPS assemelhar-se às forças de interação exigidas pelo nosso estudo de caso. O imprescindível deste arquivo é a descrição da magnitude das forças e a energia de ligação em função da distância da ligação. Os dois arquivos mencionados anteriormente estão escritos a partir de códigos em *Python* de forma a serem facilmente ajustáveis.

Na posse deste arquivo, com a posição e velocidade de todas as partículas para cada iteração do simulador, foi escrito um código em *Python* para podermos analisar os resultados e compará-los à solução obtida na seção anterior. Essencialmente, é realizada uma transformada rápida de Fourier permitindo a comparação de resultados.

### 3 RESULTADOS

Para a obtenção dos resultados foram utilizadas 32 partículas nas simulações do sistema FPUT. Ademais, foi simulado um sistema com interações quadráticas (obedecendo a equação 26), em que o coeficiente  $\alpha$  tinha um valor de 0,25. Vale notar que, como a razão de  $k/m$  foi assumida como a unidade, seus valores não importam para as simulações derivadas da implementação em *Python*. Entretanto, para a implementação no *LAMMPS* esses valores são relevantes. No *LAMMPS* foi assumida os valores  $k = 1 \text{ N/m}$  e  $m = 1 \text{ Kg}$  para a especificação da força de ligação entre as partículas.

Para ilustrar a equipartição de energia, foi executado o código A.1 ajustando-se as condições iniciais. Todas as partículas foram iniciadas com velocidade nula. Além disso, seus deslocamentos estavam apenas no primeiro modo normal, cuja amplitude máxima era de  $10 \text{ m}$ . Estas condições têm o efeito prático de tornar a energia do sistema relativamente alta. Com estes resultados, dois gráficos foram gerados.

Na Figura 2, temos a representação da quantidade de energia alocada nos primeiros cinco modos de vibração pelo tempo de evolução. Já a Figura 3 mostra a energia média final de cada modo normal. É importante ressaltar que, para se obter a energia média final, foram utilizados os últimos dez valores obtidos de cada modo normal. Desta forma, notamos que a energia é distribuída ao longo do tempo entre os diversos modos normais.

Este resultado não foi o encontrado por Fermi e co-autores, pois os mesmos estabele-

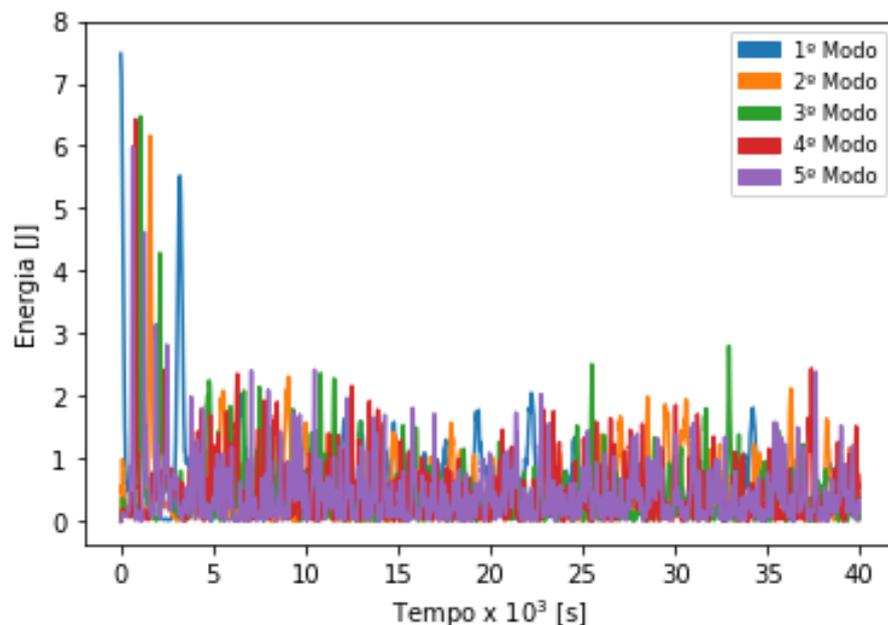


Figura 2 – Gráfico das energias armazenadas nos cinco primeiros modos normais (para uma energia inicial alta).

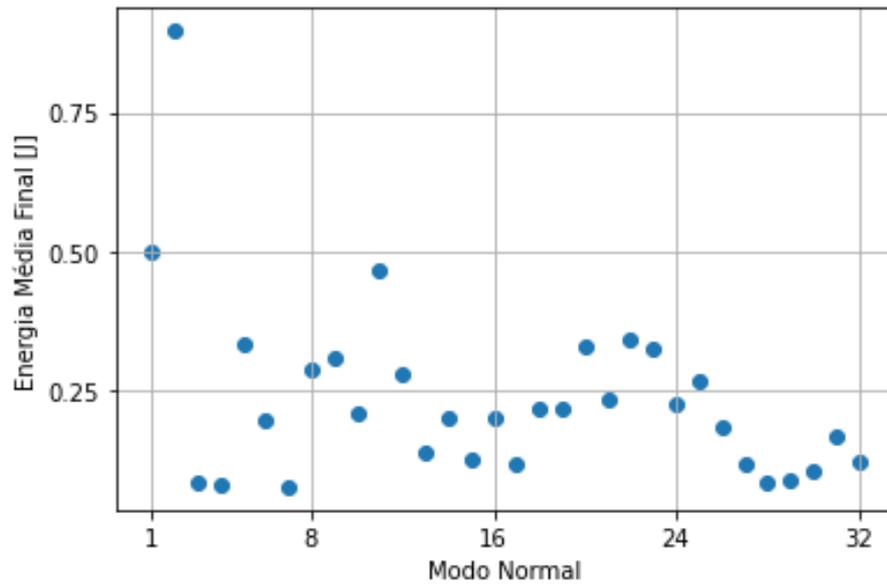


Figura 3 – Gráfico da energia média final associada a cada modo normal

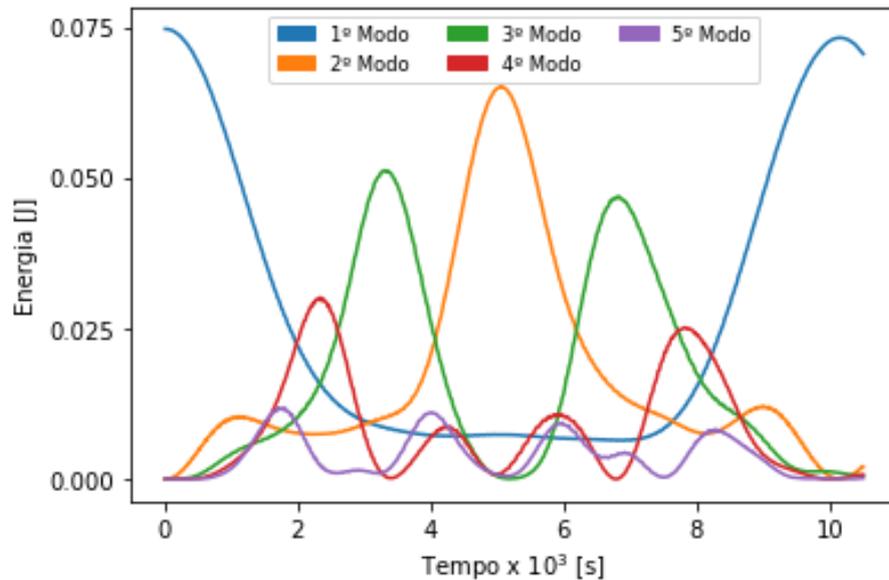


Figura 4 – Gráfico das energias armazenadas nos cinco primeiros modos normais (para uma energia inicial baixa).

ceram uma energia total inicial menor. Para realizarmos uma simulação similar à feita por FPUT, foi executado o código anteriormente mencionado alterando a amplitude máxima do primeiro modo normal para  $1\text{ m}$ , sendo os resultados demonstrados graficamente nas Figuras 4 e 5.

Nota-se, na Figura 4, que a energia está, à princípio, totalmente no primeiro modo de vibração, devido às condições iniciais das partículas. Ao acompanhar a energia desse modo, percebe-se que a mesma se distribui ao longo do tempo para outros modos normais de valor baixo. Contudo, antes que esta energia possa ser alocada em modos mais altos, a

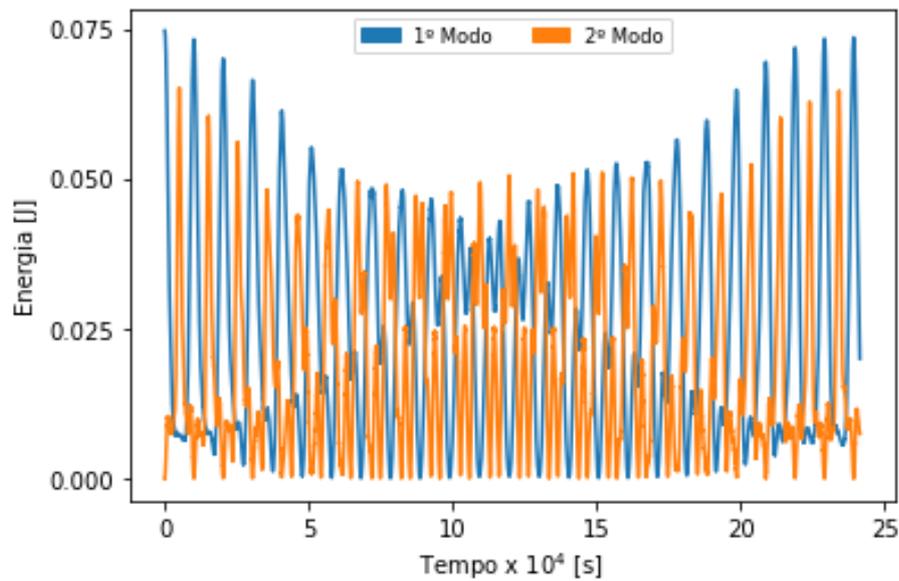


Figura 5 – Gráfico das energias armazenadas nos dois primeiros modos normais (para um tempo de iteração alto).

Tabela 1 – Tabela das frequências para a solução numérica, em Hz.

1º modo normal (Hz)	1ª ordem (Hz)	2ª ordem (Hz)
$9,52 \times 10^{-2}$	$9,62 \times 10^{-5}$	$4,18 \times 10^{-6}$

mesma retorna quase inteiramente para o primeiro modo normal.

Esta recorrência é exibida na segunda coluna da Tabela 1 e é denominada de Frequência de Recorrência de Primeira Ordem. A primeira coluna da Tabela 1 contém a frequência natural do modo normal (no caso, para o primeiro modo) e é calculada diretamente da equação 42. Aqui, ela nos ajuda a fornecer uma dimensão das frequências envolvidas no problema.

A Figura 5 representa apenas os modos 1 e 2, de modo a facilitar a compreensão dos dados. É notável, portanto, o fluxo do ciclo de energia partindo do primeiro modo normal e depois se distribuindo para os demais, representados pelo segundo modo normal. Desejamos enfatizar que a quantidade de energia retornada para o modo 1 varia com o tempo. Ou seja, além da recorrência de primeira ordem, temos uma recorrência de segunda ordem com uma frequência menor, como mostrado na Tabela 1.

Na recorrência de segunda ordem, podemos notar que os picos do primeiro modo normal diminuem, perdendo até 40 % da sua energia inicial. Depois, retornam basicamente às condições iniciais, com apenas 3 % de perda. Para melhor visualização dessa oscilação, temos o gráfico (representado pela Figura 6), com os valores dos picos do primeiro modo normal em função do tempo. Neste, esta oscilação é evidenciada e podemos, inclusive, determinar sua frequência.

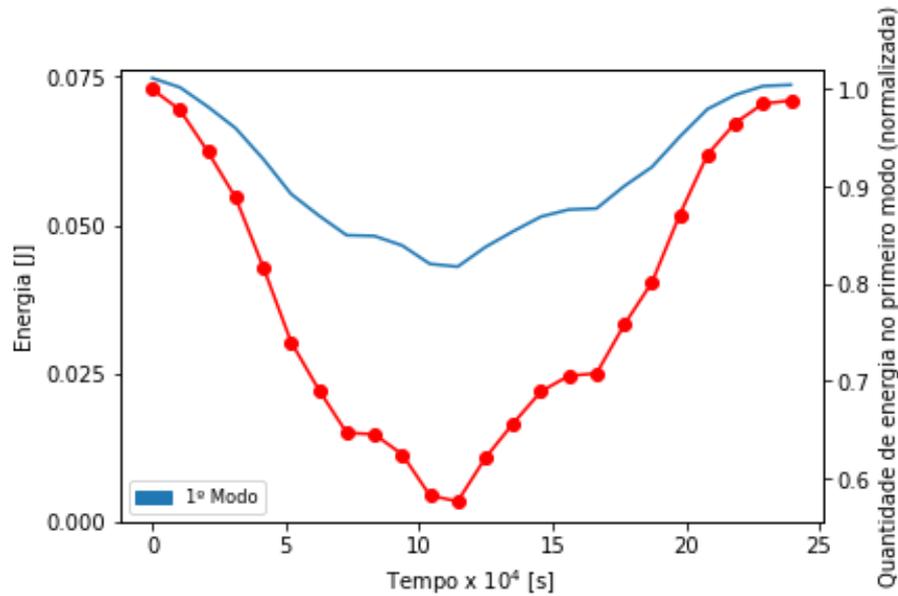


Figura 6 – Gráfico dos picos do primeiro modo normal em função do tempo, gerando uma função “envelope”, representada pela linha azul. A quantidade de energia alocada no primeiro modo normal em relação à energia total é sinalizada pela linha vermelha com pontos.

Na Figura 6, a linha vermelha com pontos mostra o valor relativo do pico em relação ao valor da energia total, de forma a facilitar a percepção da quantidade percentual da perda de energia para os demais modos normais.

Ao longo dos anos, com o estudo do paradoxo de FPUT, foi questionada a existência de recorrências de níveis mais altos. Com este questionamento surge, no entanto, a dificuldade de se notar numericamente as alterações conforme aumenta-se a ordem de recorrência. Além disso, para as alterações serem notadas, a simulação deve ocorrer por mais tempo, o que está além da capacidade de um computador pessoal. Apesar de já ter sido comprovada a existência de recorrências de terceira e quarta ordem, não foi possível demonstrar nesta simulação, pois é demasiadamente curta (PACE; CAMPBELL, 2019).

Com isto, temos que dois exemplos foram analisados: o primeiro, no qual foi fornecido uma grande quantidade de energia para o primeiro modo normal e o segundo, em que a energia fornecida não foi tão alta quanto a utilizada neste trabalho. No primeiro caso, vemos uma dispersão da energia por todos os modos normais, enquanto que, no segundo caso, a energia não se equiparte.

Podemos explicar este comportamento através de um teorema enunciado por Kolmogorov (1954), o qual deu surgimento à teoria KAM (*Kolmogorov-Arnol'd-Moser*). Esta teoria diz respeito ao fato que, se possuímos um sistema oscilatório conservativo não-linear, em que existe uma interação não-linear fraca, essa interação servirá apenas para mudar levemente as frequências de oscilação e introduzir pequenos harmônicos não-lineares. Porém, se possuímos uma interação não-linear forte (ou um sistema com energia alta), é

Tabela 2 – Tabela de frequências, em Hz, para dois passos temporais diferentes e valores relativos percentuais  $\Delta f_t$  comparados aos da solução numérica, sendo o subscrito  $t$  indicativo do passo temporal 0,1 ou 0,5.

Timestep	1 <sup>a</sup> ordem (Hz)	$\Delta f_{0,1}$	2 <sup>a</sup> ordem (Hz)	$\Delta f_{0,5}$
0.1	$9.55 \times 10^{-5}$	99	$4.34 \times 10^{-6}$	96
0.5	$8.25 \times 10^{-5}$	86	$3.75 \times 10^{-6}$	89

esperado que ele atinja equilíbrio térmico, no caso abordado neste trabalho (WALKER; FORD, 1969).

### 3.1 PACOTE COMPUTACIONAL

Obtidas as posições e velocidades das partículas pelo LAMMPS, foi executado o código A.4, para análise dos resultados obtidos. Com estes resultados, foram elaborados os gráficos apresentados na Figura 7a e 7b. Podemos notar que os gráficos das figuras 7 e 8 são muito similares aos apresentados na solução obtida estritamente pelo *Python*. Porém, há uma distinção entre eles: o número de iterações realizadas. No caso de dinâmica molecular, este número é em função do passo temporal.

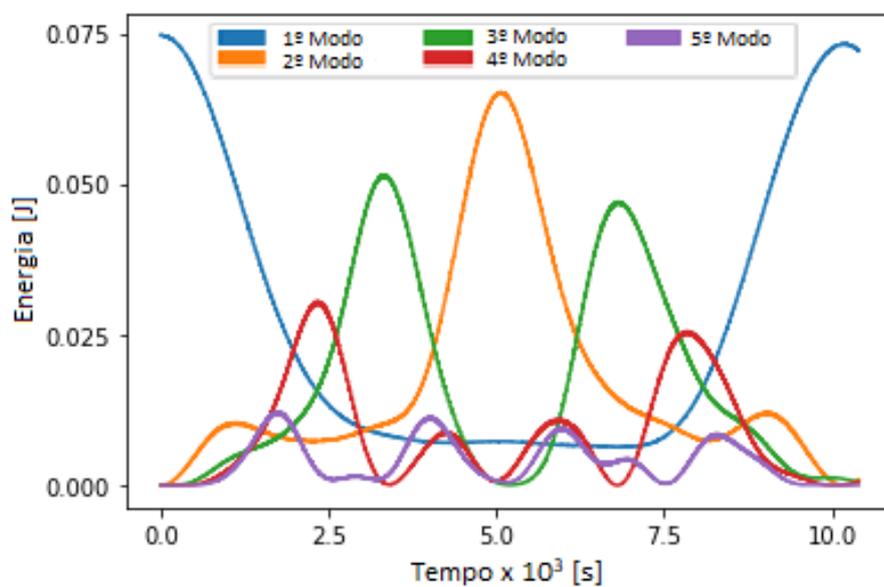
Quanto maior o passo de integração em uma simulação de dinâmica molecular, mais impreciso se torna a conservação de energia. Isso pode ser percebido pela maior participação dos modos normais (além do primeiro) na alocação de energia, assim como nas frequências de recorrência que foram calculadas.

Tomando as frequências obtidas pela solução numérica como referência, os dados para cada frequência, assim como suas concordâncias associadas, são apresentados na Tabela 2. Portanto, longos passos de integração levam a uma pior conservação da energia (TUCKERMAN, 2010).

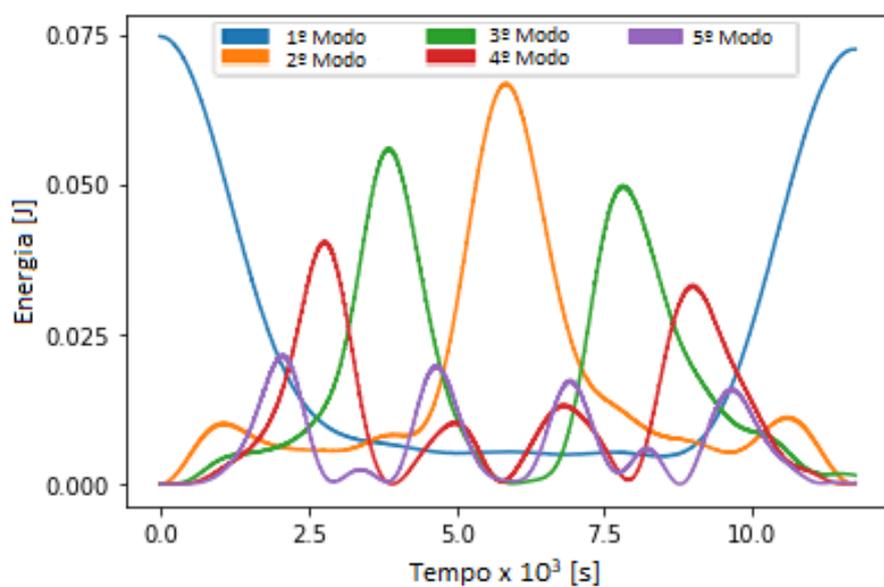
A partir das figuras 7 e 8, podemos mensurar as frequências  $f_{DM}$  de 1<sup>a</sup> e 2<sup>a</sup> ordem para os passos temporais de 0,1 e 0,5. Os resultados dessas determinações são mostrados na Tabela 2, junto com os valores de concordância,  $\Delta f$ , entre a solução por dinâmica molecular  $f_{DM}$  e a solução numérica  $f_{SN}$

$$\Delta f = 100 \times \left( 1 - \frac{|f_{DM} - f_{SN}|}{f_{SN}} \right). \quad (55)$$

Notamos que há uma redução relativa das frequências para a solução numérica e, quanto maior o passo temporal, maior ainda é a redução da frequência. Os valores de 2<sup>a</sup> ordem são mais afetados, pois necessitam de processamentos mais longos para se obter precisão similar aos de 1<sup>a</sup> ordem. Assim, a conservação de energia para processamentos mais longos é diretamente afetada pela grandeza do passo temporal, como discutido anteriormente.

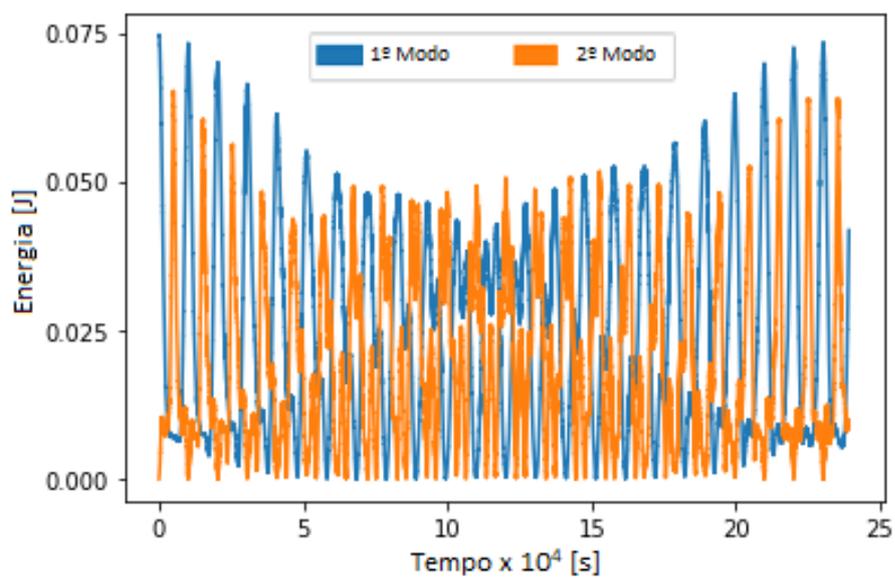


(a) Passo temporal de 0,1.

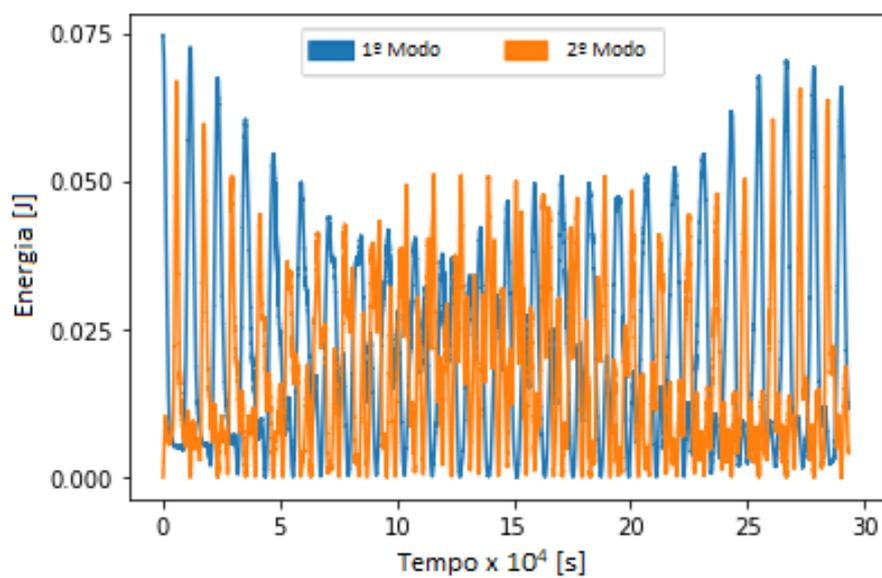


(b) Passo temporal de 0,5.

Figura 7 – Representação das simulações de dinâmica molecular com dois passos temporais diferentes, mostrando as energias armazenadas nos cinco primeiros modos normais (para uma recorrência de primeira ordem).



(a) Passo temporal de 0,1.



(b) Passo temporal de 0,5.

Figura 8 – Representação das simulações de dinâmica molecular com dois passos temporais diferentes, mostrando as energias armazenadas nos dois primeiros modos normais (para uma recorrência de segunda ordem).

## 4 CONCLUSÃO

Ao se estudar problemas em Física, muitas vezes nos deparamos com casos que assumem premissas relativamente simples e que possivelmente geram soluções analíticas. Entretanto, ao incrementar pequenas complexidades às premissas, essas soluções comumente se tornam muito mais complexas de serem obtidas analiticamente, ocasionalmente impossíveis. São nestes casos que os métodos computacionais (e, por consequência, o ramo de física computacional) demonstram o seu valor.

Nos fundamentos teóricos foram desenvolvidos os conceitos de mecânica clássica necessários para a compreensão do paradoxo FPUT. A princípio, foram revisados os conceitos de mecânica newtoniana, que governa a dinâmica do sistema. Em seguida, foi realizada uma descrição detalhada do sistema FPUT, assim como um desenvolvimento das equações de movimento do sistema. Consequentemente, foram abordados dois conceitos importantes para a análise do paradoxo: o de modos normais e o teorema da equipartição de energia.

Sobre a implementação dos algoritmos, foi abordado em detalhes o desenvolvimento em *Python*, com os códigos disponíveis no apêndice, assim como toda as informações para viabilizar a utilização do LAMMPS. Foi explanada, também, as condições iniciais e, por fim, demonstrado os valores esperados de acordo com o teorema da equipartição de energia.

Para fins de comparação e validação da solução através de métodos de dinâmica molecular, foram obtidas duas frequências características do paradoxo FPUT: as relativas às recorrências de primeira e segunda ordem. A partir da análise destes resultados, constatou-se que a solução obtida a partir dos resultados do LAMMPS são adequadas.

Desta forma, é possível embasar a utilização de outras ferramentas do LAMMPS para trabalhos futuros que envolvam física computacional, até mesmo pesquisas que envolvam preceitos da mecânica quântica, uma vez que a aplicação do LAMMPS para dinâmica molecular foi validada com este trabalho.

O presente trabalho permitiu o aprendizado de conceitos de Física Computacional assim como métodos utilizados em pesquisa, por meio da programação em *Python*. Além disso, viabilizou o aprofundamento de conceitos de dinâmica molecular. Vale ressaltar que, apesar do trabalho não ser cientificamente inédito, o mesmo pôde ser adaptado e publicado como um artigo na área de ensino, ilustrando conceitos de Física e Física Computacional para alunos de ensino superior (LUCERO; MOREIRA, 2021).

## REFERÊNCIAS

- ARFKEN, G. B.; WEBER, H. J. **Mathematical methods for physicists**. [S.l.]: American Association of Physics Teachers, 1999. Citado na página 23.
- BOYCE, W. E.; DIPRIMA, R. C. **Elementary differential equations and boundary value problems**. [S.l.]: John Wiley & Sons, Inc., 2012. Citado na página 23.
- BUTCHER, J. **Numerical methods for ordinary differential equations**. [S.l.]: Wiley, 2008. Citado na página 19.
- DAUXOIS, T. Fermi, pasta, ulam and a mysterious lady. **Phys. Today**, v. 61, n. arXiv: 0801.1590, p. 55–57, 2008. Citado na página 12.
- FERMI, E.; PASTA, J.; ULAM, S. **Los Alamos internal report,(1955)**. [S.l.], 1940. Citado na página 12.
- GALLAVOTTI, G. **The Fermi-Pasta-Ulam problem: a status report**. [S.l.]: Springer, 2007. v. 728. Citado na página 12.
- GEAR, C. W. **Numerical initial value problems in ordinary differential equations**. [S.l.]: Prentice-Hall, 1971. Citado na página 21.
- GOLDSTEIN, H.; POOLE, C.; SAFKO, J. **Classical mechanics**. [S.l.]: American Association of Physics Teachers, 2002. Citado na página 28.
- LANDAU, R. H.; PAEZ, M. J.; BORDEIANU, C. C. **Computational physics: problem solving with Python**. [S.l.]: John Wiley & Sons, 2015. Citado 2 vezes nas páginas 19 e 28.
- LUCERO, D. d. M.; MOREIRA, P. A. F. P. O problema fermi-pasta-ulam-tsingou: equipartição de energia vista através de simulações computacionais. **Revista Brasileira de Ensino de Física**, Scielo, v. 43, 2021. Citado na página 37.
- NEWMAN, M. **Computational physics**. [S.l.]: CreateSpace Independent Publ., 2013. Citado na página 28.
- ONORATO, M.; VOZELLA, L.; PROMENT, D.; LVOV, Y. V. Route to thermalization in the  $\alpha$ -fermi–pasta–ulam system. **Proceedings of the National Academy of Sciences**, v. 112, n. 14, p. 4208–4213, 2015. Citado na página 13.
- PACE, S. D.; CAMPBELL, D. K. Behavior and breakdown of higher-order fermi-pasta-ulam-tsingou recurrences. **Chaos: An Interdisciplinary Journal of Nonlinear Science**, v. 29, n. 2, p. 023132, 2019. Citado na página 33.
- PINTO, L. M. C.; QUAINO, P.; ARCE, M. D.; SANTOS, E.; SCHMICKLER, W. Electrochemical adsorption of oh on pt(111) in alkaline solutions: combining dft and molecular dynamics. **ChemPhysChem**, v. 15, n. 10, p. 2003–2009, 2014. Citado na página 29.
- PIRES, A. S. **Evolução das idéias da física**. [S.l.]: Livraria da Física, 2011. Citado na página 15.
- PLIMPTON, S. Fast parallel algorithms for short-range molecular dynamics. **Journal of Computational Physics**, Elsevier, v. 117, n. 1, p. 1–19, 1995. Citado na página 28.

PRASANTH, P.; NISHANTH, P.; UDAYANANDAN, K. Equipartition theorem for non-linear oscillators. **Physics Education**, v. 35, n. 4, p. 58–64, 2019. Citado na página 26.

REIF, F. **Fundamentals of statistical and thermal physics**. [S.l.]: Waveland Press, 2009. (McGraw-Hill series in fundamentals of physics). Citado na página 26.

SWOPE, W. C.; ANDERSEN, H. C.; BERENS, P. H.; WILSON, K. R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. **The Journal of Chemical Physics**, v. 76, n. 1, p. 637–649, 1982. Citado na página 21.

TUCKERMAN, M. **Statistical mechanics: theory and molecular simulation**. [S.l.]: Oxford university press, 2010. Citado 3 vezes nas páginas 19, 28 e 34.

VERLET, L. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. **Phys. Rev.**, American Physical Society, v. 159, p. 98–103, 1967. Citado na página 21.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J. et al. Scipy 1.0: fundamental algorithms for scientific computing in python. **Nature Methods**, Nature Publishing Group, v. 17, n. 3, p. 261–272, 2020. Citado na página 28.

WALKER, G. H.; FORD, J. Amplitude instability and ergodic behavior for conservative nonlinear oscillator systems. **Phys. Rev.**, American Physical Society, v. 188, p. 416–432, 1969. Citado na página 34.

## **Apêndices**

## APÊNDICE A – Códigos em Python

Código A.1 – Solução Utilizando Python

```
# Importando bibliotecas e funções necessárias
import math as m
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from scipy.integrate import odeint

# Definindo parâmetros do sistema
N=32          # Número de partículas. Tem que ser uma potência de
              dois.
alpha=0.25   # Coeficiente da interação não-linear.
R_1_O=10500  # Tempo de recorrência de primeira ordem
              aproximadamente.
DT=20        # Define o tamanho do timestep.

# Definindo função para solucionar o sistema FPUT.
# Retorna a energia de cada modo normal e uma lista com o tempo
de cada iteração.
def integrar(a=1, mult=1):
    # 'a' é o deslocamento máximo das partículas. Ligado a
    energia total.
    # 'mult' é quantidade de vezes que 'R_1_O' é multiplicado.
    Define a quantidade de iterações do sistema.

    tempoMax=R_1_O*mult      # Tempo máximo de iteração.
    maxIter=tempoMax//DT     # Quantidade de iterações.
    tempo= list(range (0,tempoMax+1, DT))

    condIniciais=[]         # Condições iniciais do sistema.
    omegak2=[]             # Frequência ao quadrado de cada
                          modo normal.
    # Excitação inicial
    for x in range(1, N+1):
        condIniciais.insert(x-1,a*m.sin(m.pi*x/(N+1)))
        condIniciais.insert(x+N-1,0)
```

```

omegak2.insert(x-1, 4*(m.sin(m.pi*x/2/(N+1)))**2)

# Integração temporal
D=[0]*2*N
def dydt(y,t):          # Função que define as equações
    diferenciáveis do sistema.
    D[0]=y[N]
    D[1:N-1] = y[N+1: 2*N-1]
    D[N-1]=y[2*N-1]
    D[N]=y[1]-2*y[0]+alpha*((y[1]-y[0])**2-y[0]**2)
    D[N+1:2*N-1] = y[2:N]+y[0:N-2]-2*y[1:N-1]+alpha*((y[2:N]
        ]-y[1:N-1])**2-(y[1:N-1]-y[0:N-2])**2)
    D[2*N-1]=y[N-2]-2*y[N-1]+alpha*(y[N-1]**2-(y[N-1]-y[N
        -2])**2)
    return D
solucao=odeint(dydt, condIniciais, tempo)
tempo.pop(0)

# Ajustando a solução e realizando uma transformada rápida
de Fourier.
YX,YV,x,v,Energia = [], [], [], [], []
for i in range(0,maxIter):
    YX.insert(i,[])
    YV.insert(i,[])
    x.insert(i,[])
    v.insert(i,[])
    Energia.insert(i,[])
for i in range(1,maxIter+1):
    YX[i-1].insert(0,0)
    YV[i-1].insert(0,0)
    for j in range(0,N):
        YX[i-1].insert(j+1,solucao[i-1][j])
        YV[i-1].insert(j+1,solucao[i-1][j+N])
for i in range(0,maxIter):
    for j in range(0,N+1):
        x[i].insert(j,YX[i][j])
        v[i].insert(j,YV[i][j])
    x[i].insert(N+1,0)
    v[i].insert(N+1,0)

```

```

    for j in range (N,0,-1):
        x[i].insert(66-j,-YX[i][j])
        v[i].insert(66-j,-YV[i][j])

sXF=((np.fft.fft(x)).imag)/m.sqrt(2*(N+1))
sVF=((np.fft.fft(v)).imag)/m.sqrt(2*(N+1))

# Calculando a energia de cada modo normal ao longo do tempo
.
for i in range(0,len(Energia)):
    for j in range(N):
        Energia[i].insert(j,(omegak2[j]*(sXF[i][j+1])**2+(
            sVF[i][j+1])**2)/2)
    return np.array(Energia),tempo

# Obter as cores padrão do plot e definir a legenda dos modos
normais
cmap=plt.get_cmap('tab10')
modo1=mpatches.Patch(color=cmap(0),label='1º Modo')
modo2=mpatches.Patch(color=cmap(1),label='2º Modo')
modo3=mpatches.Patch(color=cmap(2),label='3º Modo')
modo4=mpatches.Patch(color=cmap(3),label='4º Modo')
modo5=mpatches.Patch(color=cmap(4),label='5º Modo')

# Obter resultados para uma energia inicial alta
# Desenhando a evolução temporal da energia dos cinco primeiros
modos normais
Energ,t=integrar(10,4)
E=Energ[:, :5]
tempo=np.array(t)/1000
plt.figure(0)
fig,ax=plt.subplots()
plt.xlabel('Tempo x $10^3$ [s]')
plt.ylabel('Energia [J]')
plt.plot(tempo[:1999],E[:1999])
plt.yticks(np.arange(0,9,1))
plt.legend(handles=[modo1,modo2,modo3,modo4,modo5],loc='best',
    ncol=1,fontsize='small')
plt.show()

```

```

# Desenhando a energia média final dos N modos normais
media=[]
for i in range(N):
    media.append(np.mean(Energ[-10:,i]))
plt.figure(1)
fig, ax=plt.subplots()
plt.xlabel('Modo Normal')
plt.ylabel('Energia Média Final [J]')
plt.grid(b=True)
plt.xticks([1,8,16,24,32])
plt.yticks(np.arange(0, 1.25, 0.25))
plt.scatter(range(1,N+1),media)
plt.show()

# Obter resultados para uma energia inicial baixa
# Desenhando a evolução temporal da energia dos cinco primeiros
    modos normais
Energ,t=integrar(1,1)
E=Energ[:, :5]
tempo=np.array(t)/1000
plt.figure(2)
fig, ax=plt.subplots()
plt.xlabel('Tempo x $10^3$ [s]')
plt.ylabel('Energia [J]')
plt.plot(tempo,E)
plt.yticks(np.arange(0, 0.1, 0.025))
plt.legend(handles=[modo1, modo2, modo3, modo4, modo5], loc='upper
    center', ncol=3, fontsize='small')
plt.show()

# Obter resultados para recorrência de segunda ordem
# Desenhando a evolução temporal da energia dos dois primeiros
    modos normais
Energ,t=integrar(1,23)
E=Energ[:, :2]
tempo=np.array(t)/10000
plt.figure(3)
fig, ax=plt.subplots()

```

```

plt.xlabel('Tempo x $10^4$ [s]')
plt.ylabel('Energia [J]')
plt.plot(tempo,E)
plt.yticks(np.arange(0, 0.1, 0.025))
plt.legend(handles=[modo1,modo2],loc='upper center',ncol=2,
           fontsize='small')
plt.show()

# Desenhando função envelope do primeiro modo normal.
ENE,porcentagem ,maxi = [], [], [0,0]
for i in range(0, len(Energ), 510):
    maxi[0]=0
    inte = []
    a=0
    for j in range(510):
        if (i+j)<len(Energ):
            inte.append(Energ[i+j][0])
            if (Energ[i+j][0] > maxi[0]):
                maxi[0]=Energ[i+j][0]
                maxi[1]=i+j
    for k in range(N):
        a+=Energ[maxi[1]][k]
    porcentagem.append(Energ[maxi[1]][0]/a)
    ENE.append(max(inte))
tempo=np.linspace(0, maxi[1]*DT/10000, len(ENE))
plt.figure(2)
fig, ax=plt.subplots()
plt.xlabel('Tempo x $10^4$ [s]')
plt.ylabel('Energia [J]')
plt.plot(tempo,ENE)
plt.yticks(np.arange(0, 0.1, 0.025))
ax2=ax.twinx()
ax2.plot(tempo, porcentagem, color='red', marker='o')
ax2.set_ylabel("Quantidade de energia no primeiro modo (
normalizada)")
plt.legend(handles=[modo1], loc='best', fontsize='small')
plt.show()

```

Código A.2 – Condições Iniciais

# -\*- coding: utf-8 -\*-

```

# Número de partículas simuladas com duas partículas a mais para
servirem de parede
N=34
b=[]
# Amplitude de deslocamento
a=1.0
# Atribuindo os deslocamentos relativos da x-ésima partícula em
relação ao seu centro de equilíbrio
import math
for x in range(1, N-1):
    b.append(a*math.sin(math.pi*x/(N-1)))

with open('fput.in', 'w+') as f:
    f.write('atoms position and bonds\n')
    f.write(str(N)+' atoms\n')
    f.write(str(N-1)+' bonds\n')
    f.write('1 atom types\n')
    f.write('1 bond types\n')
    f.write('0 '+str(N*5)+' xlo xhi\n')
    f.write('-0.5 0.5 ylo yhi\n')
    f.write('-0.5 0.5 zlo zhi\n\n')
    f.write('Atoms\n\n')
    f.write(str(0+1)+' 1 1 ' +str(0*5)+' .0 '+' 0.0 0.0\n')
    for i in range(1, N-1):
        f.write(str(i+1)+' 1 1 ' +str(i*5+b[i-1]))+' .0 '+' 0.0
            0.0\n')
    f.write(str(N)+' 1 1 ' +str((N-1)*5)+' .0 '+' 0.0 0.0\n')
    f.write('\nBonds\n\n')
    for i in range(1, N):
        f.write(str(i) + ' 1 '+str(i)+' '+str(i+1)+'\n')

```

### Código A.3 – Tabela para Força de Interação

```

# -*- coding: utf-8 -*-
alpha=0.25
beta=8
# Valor da rigidez da mola, numericamente igual a massa
k=1
# Distância básica dos átomos em que a mola está em repouso
l=5
# Incremento para o tamanho da ligação

```

```

dx=0.01
# Máximo deslocamento da mola do seu centro de equilíbrio
ux=2.5
# Quantidades de 'pontos' na tabela
N=int(2*ux/dx+1)

with open('bonds.in', 'w+') as f:
    f.write('#Termo linear e termo quadrático\n\n')
    f.write('QUAD\n')
    f.write('N '+str(N) +'\n\n')
    k2=k*alpha
    for i in range(N):
        x=i*dx+ux
        f.write(str(i+1) + ' ' + str(x) + ' ' + str(k*x**2/2+k2*x
            **2*l-k2*x**3/3-k*l*x-k2*l**2*x) + ' ' + str(k*(1-x)+k2
            *(1-x)**2) +'\n')
    f.write('\nCUB\n')
    f.write('N '+str(N) +'\n\n')
    k2=k*beta
    for i in range(N):
        x=i*dx+ux
        f.write(str(i+1) + ' ' + str(x) + ' ' + str(-x*(k*l+k2*l**3)
            +x**2*(k/2+3*k2/2*l**2)+x**3*k2*l-x**4*k2/4) + ' ' +
            str(k*(1-x)+k2*(1-x)**3) +'\n')

```

#### Código A.4 – Análise de Resultados do LAMMPS

```

#Para o funcionamento correto deste código é necessário como
    entrada o arquivo gerado pela simulação do LAMMPS
#Número de partículas
N=32
#Quantidade total de iterações feitas no LAMMPS através do
    comando run
IT=2392000
#Iterações para um dado timestep em que se obtém uma recorrência
    de primeira ordem
R_1_O=104000
#Timestep utilizado no script do LAMMPS
ts=0.1

#Para uma simulação com o timestep de 0.5, as variáveis 'IT' e'

```

*R\_1\_0' tiveram como valor 586575 e 23463*

```

import numpy as np
import math
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches

#Frequências de oscilação dos N modos normais
omegak2=[]
for x in range(1, N+1):
    omegak2.append(4*(math.sin(math.pi*x/(2*(N+1))))**2)
#

#Matrizes de posição e velocidade para as partículas
POS=np.empty([IT,2*N+2])
VEL=np.empty([IT,2*N+2])
#

#Contadores
it=0
k=0
#
with open('res.comp') as f:
    #Loop para obter a posição e a velocidade das partículas a
    partir da primeira iteração do LAMMPS
    for line in f:
        k=k+1
        # Condições para eliminar as linhas inúteis
        # O arquivo .comp tem linhas que se repetem, para situar
        qual iteração é, entre outras informações
        if ((k-1)%(N+1)!=0 and (k-2)%(N+1)!=0 and (k-3)%(N+1)
            !=0 and (k-4)%(N+1)!=0 and (k-5)%(N+1)!=0
            and (k-6)%(N+1)!=0 and (k-7)%(N+1)!=0 and (k-8)%(N
                +11)!=0 and (k-9)%(N+11)!=0
            #Condições para eliminar as informações das partículas
            extremas que são na verdade as paredes do sistema
            and (k-10)%(N+11)!=0 and (k-(N+11))%(N+11)!=0
            #Condição para ignorar a iteração zero
            and k>(N+11)):

```

```

        line=line.split()
        c=(k%(N+11)-10)
        POS[it ,c]=float ( line [0] ) -5*(c)
        VEL[it ,c]=float ( line [1] )
        if (c==N):
            it=it+1
#Vale notar que no arquivo dump são escrito apenas a posição e
    velocidade no eixo x para não ocupar muito espaço em disco

#Ajuste das matrizes para a transformada de Fourier
    POS[:,N+2:2*N+2]=-POS[:,N:0:-1]
    VEL[:,N+2:2*N+2]=-VEL[:,N:0:-1]
#

# A posição e a velocidade têm que estar organizadas em uma "
matriz", ou lista de listas
# Para a transformada de Fourier temos que ter um zero, a função
, outro zero, e o negativo da função ao contrário.

# Transformada de Fourier
    sXF=((np.fft.fft(POS)).imag)/math.sqrt(2*(N+1))
    sVF=((np.fft.fft(VEL)).imag)/math.sqrt(2*(N+1))

    #Conseguindo a energia dos primeiros cinco modos normais
    Energ=np.empty([IT,5])
    Energ[:,0:5]=(omegak2[0:5]*sXF[:,1:6]**2+sVF[:,1:6]**2)/2

# Conseguir as cores padrão de desenho e fazer as legendas
    cmap=plt.get_cmap('tab10')
    modo1=mpatches.Patch(color=cmap(0), label='modo 1')
    modo2=mpatches.Patch(color=cmap(1), label='modo 2')
    modo3=mpatches.Patch(color=cmap(2), label='modo 3')
    modo4=mpatches.Patch(color=cmap(3), label='modo 4')
    modo5=mpatches.Patch(color=cmap(4), label='modo 5')
#

# Sequência para fazer um gráfico até se obter a primeira
recorrência
    tempo=np.array(list(range(1,R_1_O+1)))*ts/1000

```

```

plt.figure(0)
fig, ax=plt.subplots()
plt.xlabel('Tempo x $10^3$')
plt.ylabel('Energia')
plt.plot(tempo, Energ[0:R_1_O,:])
plt.yticks(np.arange(0, 0.08, 0.025))
plt.xticks(np.arange(0, max(tempo), 2.5))
plt.legend(handles=[modo1, modo2, modo3, modo4, modo5], loc='best',
           ncol=3, fontsize='small')

# Sequência para se fazer um gráfico das energias para toda a
# simulação
tempo=np.array(list(range(1, IT+1)))*ts/10000
plt.figure(1)
fig, ax=plt.subplots()
plt.xlabel('Tempo x $10^4$')
plt.ylabel('Energia')
plt.plot(tempo, Energ[0:IT, 0:2])
plt.yticks(np.arange(0, 0.08, 0.025))
plt.legend(handles=[modo1, modo2], loc='best', ncol=2, fontsize='
medium')

```

**APÊNDICE B – Códigos para o LAMMPS**

## Código B.1 – Script para o LAMMPS

```
#

units si
dimension 2
atom_style bond
atom_modify sort 0 0
timestep 0.1
#timestep padrão na ordem de e-8

read_data data.inicial

write_dump all atom initial.lammpstrj

bond_style table linear 501
bond_coeff 1 bonds.table QUAD
#Caso quisesse testar o caso cúbico, apenas substituir o termo '
  QUAD' na linha acima por 'CUB'.

group AntiBoundary id 2:33

mass 1 1

neighbor 0.3 bin
neigh_modify every 20 delay 0 check no

fix 1 AntiBoundary nve

dump 1 all custom 1 res.comp x vx

run 2392000
```