RELATÓRIO DAS ATIVIDADES DA ALUNA DE INICIAÇÃO CIENTÍFICA - PIBITI/UNIR

*Projeto aprovado no Edital Nº 001/PIBITI/DPESQ/PROPESQ/UNIR/2022*

**1. IDENTIFICAÇÃO DO PROJETO**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do(a) Orientador(a) | | | Marco Polo Moreno de Souza |
| Nome do(a) Bolsista | | Calina Grazielli Dias Barros | |
| Título do Projeto | Pêndulos acoplados por uma haste – Desenvolvimento de um aplicativo de simulação computacional | | |

**2. RESUMO** (com palavras-chave – de 3 a 5)

**3. INTRODUÇÃO** (revisão bibliográfica, justificativa e objetivos) pêndulo acoplado com mola

**4. MATERIAL E MÉTODOS**

No semestre letivo 2022/1, no curso de física, foi ofertado a disciplina de Fundamentos da Programação que por indicação do orientador seria importante participar para relembrar conceitos de programação e aprender novas ferramentas como o *software Visual Studio Code* e o site <<https://www.w3schools.com/>>.

Para a realização, da Etapa 1: Estudo da dinâmica de um pêndulo simples, proposta no plano de trabalho foi articulado com o orientador encontros semanais para aprofundamento das noções básicas de um pêndulo simples. A princípios os encontros eram realizados nas segundas-feiras, no qual o orientador determinava os pontos focais de estudo e também explanava sobre o tema, tirando dúvidas e apresentando materiais de suporte para o estudo em casa.

Como parte do processo de ensino e aprendizagem o orientador solicitou que fosse feito a comparação entre a simulação do pêndulo disponível no site <<https://simufisica.com/>> e o método analítico de Euler.

O orientador propôs a participação na sessão de banner do evento, organizado pelo *Campus* Ji-Paraná, intitulado I Simpósio de Cursos da Universidade Federal de Rondônia Campus Ji-Paraná, nesse evento ocorreu simultaneamente a XIV Semana da Física (ver seção 6).

Para a realização da Etapa 2: Estudo da linguagem *Javascript*, iniciou-se o processo de programação no *software Visual Studio Code* que é livre e gratuito. Nos encontros com o orientador era explicado a lógica envolvida na programação bem como a teoria do método numérico do Runge-Kutta.

O método Runge-Kutta é um importante métodos interativo para solução de Equações Diferenciais Ordinárias (EDO). Para o projeto foi usado o método Runge-Kutta de 4ª ordem (CITAÇÃO), as equações usadas podem ser observadas na Tabela 1.

**Tabela 1.** Equações do método Ruge-Kutta 4ª ordem.

|  |  |
| --- | --- |
| Problema de Valor Inicial (PVI) |  |
| Método Runge-Kutta de 4ª ordem |  |
|  |
| Inclinações estimadas |  |
|  |
|  |
|  |

h: passo de Ruge-Kutta (s), g: aceleração da gravidade (m/s²), l: comprimento do fio (m), k1: inclinação no início do intervalo, k2: inclinação no ponto médio do intervalo, k3: inclinação no ponto médio do intervalo, k4: inclinação no final do intervalo, θ: ângulo que o pêndulo faz com a vertical (graus), ω: velocidade angular (rad/s).

**5. RESULTADOS E DISCUSSÃO**

Como etapa do projeto foi realizado a simulação do pêndulo simples usando o *software Visual Studio Code*, o código finalizado pode ser observado no Apêndice A, baseado no método Ruge-Kutta para solução de Equações Diferenciais Ordinárias.

Para essa simulação tanto no Simufísica quanto no *software Visual Studio Code* foram usados os parâmetros dispostos na Tabela 2:

**Tabela 2.** Parâmetros usados para realizara a comparação.

|  |  |  |  |
| --- | --- | --- | --- |
| Condições iniciais | Valores | Parâmetros | Valores |
| θ0 | 15° | g | 10 m/s² |
| ω0 | 0 rad/s | h | 0,004 s |
| - | - | L | 150 cm |

A determinação de parâmetros inicias foram importantes para que fosse possível a comparação e verificação das equações usadas, bem como para entender e compreender os procedimentos matemáticos e físicos envolvidos, além disso, foi uma prática relevante para o aprendizado da programação.

Na Tabela 3 pode ser observado o teste realizado na multiplataforma Simufísica e no *software Visual Studio Code*.

**Tabela 3.** Imagem dos testes realizados – Pêndulo Simples.

|  |  |
| --- | --- |
| Teste realizado no Simufísica | Código da programação: *software Visual Studio Code* |
|  |  |

**Fonte:** Simufísica (2023).

De posse dos dados obtidos nos testes foi possível montar o gráfico que pode ser visto na Figura 1. O que se observa é o ajuste dos dois testes, isso mostra que o código de programação construído para a simulação do pêndulo simples está correto e pode ser usado para a próxima etapa do projeto.

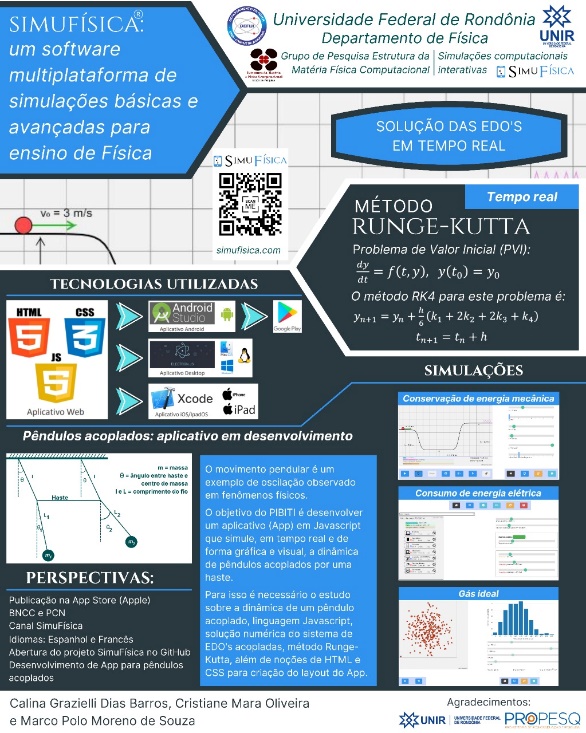
**Figura 1.** Ângulo do pêndulo em função do tempo com dados obtidos do Simufísica (linha sólida) e do código desenvolvido durante a realização do projeto (linha tracejada). Parâmetros e condições iniciais Tabela 2.

**Fonte:** Simufísica (2023).

**6. PRODUÇÃO CIENTÍFICA**

Participação no I Simpósio de Cursos da Universidade Federal de Rondônia Campus Ji-Paraná, no Anexo A pode ser verificado o certificado de participação com carga horária de 40h, nesse evento ocorreu simultaneamente a XIV Semana da Física, na qual além de participar de oficinas e palestras, também realizei a apresentação do banner intitulado: SIMUFÍSICA: um software multiplataforma de simulações básicas e avançadas para o estudo da física, na Figura 2 pode ser observado o banner organizado e apresentado.

**Figura 2.** Banner apresentado na XIV Semana da Física.



**Fonte:** Simufísica (2022) e Autora (2022).

**7. CONCLUSÕES**

Conforme o cronograma das ações a serem desenvolvidas disposto no plano de trabalho, foram cumpridas as: etapa 1 – estudo da dinâmica de um pêndulo simples e parcialmente (dentro do proposto) a etapa 2 – estudo da linguagem JavaScript.

Nesse período de participação do projeto foi possível adquirir conhecimentos básicos sobre a linguagem JavaScript, verificar a aplicabilidade da física no estudo do pêndulo simples e melhorar o aprendizado no que diz respeito a investigações necessárias para aprofundamento do tema.

**8. REFERÊNCIAS**

Ji-Paraná, 12 de janeiro de 2023

Assinatura do(a) orientador(a)



Assinatura do(a) bolsista

**APÊNDICE A –** Código da programação do pêndulo simples usando Runge-Kutta 4ª ordem

<!DOCTYPE html>

<html lang="br\_pt">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Pendulo\_Simples</title>

</head>

<body>

<h1>

Simulação do pêndulo simples

</h1>

<p>

Comprimento do fio (L) = <input id="l" type="number" value="1.5"> (m)

</p>

<p>

Gravidade (g) = <input id="g" type="number" value="10"> (m/s²)

</p>

<p>

Ângulo inicial (theta) = <input id="theta0" type="number" value="15"> (grau)

</p>

<p>

<button onclick="iniciar()">Iniciar</button>

</p>

<p>

<canvas id="myCanvas" width="400" height="400" style="border:1px solid #000000;"></canvas>

</p>

<script>

//variáveis do desenho (canva)

var canvas = document.getElementById("myCanvas");

var pincel = canvas.getContext("2d");

//variáveis para cálculo

var l = 1.5//Number(document.getElementById("l").value);

var g = 10//Number(document.getElementById("g").value)

var theta0;

var pi = 3.14159265359;

var omega = 0, theta, h = 0.004, t=0, thetaA;

var kt1, kt2, kt3, kt4, kw1, kw2, kw3, kw4, x=200, y=265;

function desenhar()

{

pincel.moveTo(300, 100);

pincel.lineTo(100, 100);

pincel.moveTo(200, 100);

pincel.lineTo(x, 250);

pincel.stroke();

pincel.beginPath()

pincel.arc(x, 265, 15, 0, Math.PI \* 2);

pincel.stroke();

}

desenhar()

function rk() //função Runge-Kutta

{

// Cálculo das inclinações

kt1 = omega //inclinação no início do intervalo de theta

kw1 = -g/l\*Math.sin(theta) //inclinação no início do intervalo de omega

kt2 = omega + 0.5\*h\*kw1 //inclinação no ponto médio do intervalo de theta

kw2 = -g/l\*Math.sin(theta + 0.5\*h\*kt1) //inclinação no ponto médio do intervalo de omega

kt3 = omega + 0.5\*h\*kw2 //nova inclinação no ponto médio do intervalo de theta

kw3 = -g/l\*Math.sin(theta + 0.5\*h\*kt2) //nova inclinação no ponto médio do intervalo de omega

kt4 = omega + h\*kw3 //inclinação no final do intervalo de theta

kw4 = -g/l\*Math.sin(theta+h\*kt3) //inclinação no final do intervalo de omega

// Método Runge-Kutta

theta = theta + h/6.0\*(kt1 + 2\*kt2 + 2\*kt3 + kt4)

omega = omega + h/6.0\*(kw1 + 2\*kw2 + 2\*kw3 + kw4)

t = t + h

thetaA = theta0\*Math.cos(Math.sqrt(g/l)\*t)

}

function iniciar()

{

theta0 = (Number(document.getElementById("theta0").value) \* pi)/180;

theta = theta0

function simular()

{

pincel.clearRect(0, 0, 400, 400);

rk()

x = 100\*l\*Math.sin(theta)+200;

y = 100\*l\*Math.cos(theta)+165;

desenhar()

console.log(t, theta)

//console.log(x, y, aa, va, theta)

}

setInterval(simular, 1)

}

</script>

</body>

</html>

**ANEXO A –** Código da programação do pêndulo simples usando Runge-Kutta 4ª ordem

