UFMG — Universidade Federal de Minas Gerais	**Universidade Federal de Minas Gerais** **Turma:** Ciência da Computação **Prof.:** Denise **Nome:** Rubia Alice Moreira de Souza

Aplicação de Equações Diferenciais no Curso

Belo Horizonte
2024

1. Aplicação em Desenvolvimento de Jogos

A área de desenvolvimento de jogos abrange diversas outras, desde das artes, para a criação de cenários, músicas e roteiro, até a computação, para programar a interação do jogador e as mecânicas disponíveis para ele. Uma parte fundamental é a matemática necessária para descrever o comportamento dos objetos, a posição deles no plano cartesiano e conseguir emular comportamentos físicos de maneira mais realista.

Nesse último caso, é onde podemos encontrar a aplicação de diversas EDOs. Por exemplo, em jogos de tiro em primeira pessoa, é necessário calcular a trajetória dos projéteis que são disparados de maneira realista, flexível e que possa ser facilmente adaptado para diferentes tipos de armas que o jogador pode utilizar. Para isso, podemos utilizar uma EDO que descreve um modelo de movimento de um projétil e programá-la utilizando métodos numéricos a fim de conseguir simular esse comportamento.

2. EDOs

Por questões de simplicidade, iremos ignorar algumas características como o efeito da curvatura da terra e a resistência do ar. Além disso, iremos considerar que a força da gravidade é constante e aplicada sempre verticalmente. Por fim, o eixo X irá representar a distância percorrida pelo projétil.

Neste caso, iremos trabalhar com duas EDOs. A primeira irá descrever o movimento do projétil em relação ao eixo X e a outra em relação ao eixo Y. Como a velocidade horizontal é constante devido às características do problema, a EDO que podemos utilizar para representar o movimento no eixo X é:

$$\frac{d^2x}{dt^2} = 0$$

Ela é uma EDO de 2ª ordem, linear, homogênea e separável.

Já para o eixo Y, ele é afetado pela aceleração da gravidade que é a única força presente neste eixo. Portanto, o movimento será descrito por:

$$\frac{d^2y}{dt^2} = -g$$

Esta é uma EDO de 2ª ordem, linear, não homogênea

3. Resolução

Agora que já temos as duas EDOs que iremos trabalhar e a classificação delas, podemos resolvê-las.

3.1. Horizontal

Como temos uma EDO separável, podemos integrar nos dois lados da equação:

$$\int \frac{d^2x}{dt^2}\, dt + c_1 = \int 0\, dt + c_2$$
$$\frac{dx}{dt} + c_1 = c_2$$
$$\frac{dx}{dt} = c_2 - c_1$$

Podemos definir "$c_2 - c_1$", como a constante "c_3":

$$c_3 = c_2 - c_1$$
$$\frac{dx}{dt} = c_3$$

Considerando que as condições iniciais, nós temos uma velocidade inicial "v_0" e um anglo de lançamento inicial "α_0", nós teremos em "t = 0":

$$\frac{dx}{dt} = v_0 cos(\alpha)$$

Portanto, temos que:

$$c_3 = v_0 cos(\alpha)$$

Assim, teremos uma EDO de 1ª ordem, linear, não homogênea separável:

$$\frac{dx}{dt} = v_0 cos(\alpha)$$

Podemos então, novamente, integrar os dois lados da equação:

$$\int \frac{dx}{dt}\, dt = \int v_0 cos(\alpha)\, dt + c_4$$
$$\int \frac{dx}{dt}\, dt = v_0 cos(\alpha) \int 1\, dt + c_4$$
$$x(t) = v_0 cos(\alpha) t + c_4$$

Podemos encontrar a constante "c_4" avaliando a equação no tempo "t = 0":

$$x(0) \ = 0 * cos(0) * 0 \ + \ c_4$$
$$c_4 \ = \ 0$$

Por fim, temos que o movimento horizontal é descrito por:

$$x(t) \ = v_0 cos(\alpha)t$$

3.2. Vertical

Agora para o movimento vertical, também temos uma EDO separável:

$$\frac{d^2y}{dt^2} \, dt \ = \ -g$$

Podemos aplicar a integral nos dois lados da equação e desenvolver:

$$\int \frac{d^2y}{dt^2} \, dt \ = \int \ -g \, dt \ + \ c_5$$
$$\int \frac{d^2y}{dt^2} \, dt \ = \ -g \int \ 1 \, dt \ + \ c_5$$
$$\frac{dy}{dt} \ = \ -gt \ + \ c_5$$

Considerando que as condições iniciais, nós temos uma velocidade inicial “v_0” e um anglo de lançamento inicial “α_0”, nós teremos em “t = 0” para o movimento vertical:

$$\frac{dy}{dt}(0) \ = \ v_0 sin(\alpha)$$
$$\frac{dy}{dt}(0) \ = \ -g \, 0 \ + c_5$$
$$c_5 \ = \ v_0 sin(\alpha)$$

Assim, chegamos a uma EDO de 1ª ordem, linear, não homogênea e separável:

$$\frac{dy}{dt} \ = \ -gt \ + v_0 sin(\alpha)$$

Então, podemos aplicar a integral nos dois lados da equação para resolver a EDO:

$$\int \frac{dy}{dt}\, dt \; = \int \; -gt \; + v_0 sin(\alpha)\, dt \; + \; c_6$$
$$\int \frac{dy}{dt}\, dt \; = \int \; -gt\, dt \; + \int \; v_0 sin(\alpha)\, dt \; + \; c_6$$
$$\int \frac{dy}{dt}\, dt \; = -g \int \; t\, dt \; + \; v_0 sin(\alpha) \int \; 1\, dt \; + \; c_6$$
$$y(t) \; = \; -gt^2 \; + \; v_0 sin(\alpha)t \; + \; c_6$$

Podemos definir a constante “c_6” avaliando “y(t)” no instante “t = 0”:

$$y(0) \; = \; -g0 \; + \; v_0 sin(\alpha)0 \; + \; c_6$$
$$c_6 \; = \; 0$$

Portanto, temos que o movimento vertical é descrito por:

$$y(t) \; = \; -gt^2 \; + \; v_0 sin(\alpha)t$$

4. Implementação com Métodos Numéricos

A fim de colocar esses conceitos em prática, desenvolvemos um site simples que implementa o comportamento desse projétil. Os parâmetros de ângulo de lançamento “α”, velocidade inicial “v_0” e gravidade “g” podem ser alterados.

O código fonte com a implementação está disponível neste link: https://github.com/Rubia-Souza/UFMG-EDC-Aplicacao-Curso.

O site com a simulação está disponível neste link: https://rubia-souza.github.io/UFMG-EDC-Aplicacao-Curso/.

O código para atualização das forças implementado foi:

```
atualizarPosicao(timeStep) {
    // Velocidade horizontal é constante
    const atualizacaoHorizontal = this.getVelocidadeHorizontal();
    const atualizacaoVertical = this.getVelocidadeVertical() - this.getGravidade() * timeStep;

    const novaPosicaoX = this.getPosicaoX() + atualizacaoHorizontal * timeStep;
    const novaPosicaoY = this.getPosicaoY() + this.getVelocidadeVertical() * timeStep;

    this.setPosicao(novaPosicaoX, novaPosicaoY);
    this.setVelocidadeHorizontal(atualizacaoHorizontal);
    this.setVelocidadeVertical(atualizacaoVertical);
}
```

Considerando que as velocidades iniciais seguem o movimento de senos e cossenos:

```
#setVelocidadeInicialHorizontal(velocidadeInicial) {
    this.velocidadeInicialHorizontal = velocidadeInicial * Math.cos(parseGrausParaRadianos(this.getAnguloLancamento()));
}

#setVelocidadeInicialVertical(velocidadeInicial) {
    this.velocidadeInicialVertical = velocidadeInicial * Math.sin(parseGrausParaRadianos(this.getAnguloLancamento()));
}
```

5. Referências

Kaelyn Parris. Numerical Simulation of Projectile Motion: Exploring Trajectories and Dynamics with Euler Method, 2023. Disponível em: <https://medium.com/@gerbald_6941/numerical-simulation-of-projectile-motion-exploring-trajectories-and-dynamics-with-euler-method-226f18aaa2c3>. Acesso em: 19/07/2024.

Math Insight. Another differential equation: projectile motion. Disponível em: <https://mathinsight.org/another_differential_equation_projectile_motion_refresher>. Acesso em: 19/07/2024.

Zaahir Ali. Using ODEs to Model Projectile Motion, 2023. Disponível em: <https://osf.io/czkhf/download&ved=2ahUKEwix8e2QubiHAxWvqpUCHel6DxMQFnoECBsQAQ&usg=AOvVaw3iHXlt86tuWa2tORperKHk>. Acesso em: 19/07/2024.