

DESENVOLVIMENTO DE OBJETOS DE APRENDIZAGEM COM O SOFTWARE MACROMEDIA FLASH®

Leandro Resende de Pádua

ITA–Instituto Tecnológico de Aeronáutica , H8-C, 309

Pç. Marechal Eduardo Gomes, n 50 – Campus do CTA, 12228-900, São José dos Campos – SP

Bolsista PIBIC-CNPq

leandrorpadua@yahoo.com.br

José Silvério Edmundo Germano

ITA–Instituto Tecnológico de Aeronáutica , Departamento de Física – IEFF

Pç. Marechal Eduardo Gomes, n 50 – Campus do CTA, 12228-900, São José dos Campos – SP

silverio@ita.br

Resumo: Dentro do ambiente da sala de aula, o microcomputador pode ser utilizado com grande eficiência, tanto no ensino da Física básica quanto nos estudos mais avançados. Devido à necessidade de se criar novas ferramentas de visualização para estudar comportamentos de diversos sistemas e, do mesmo modo, transmitir esse conhecimento para alunos de graduação, esse projeto contempla a simulação de fenômenos físicos utilizando o software Macromedia Flash®. Na solução das equações envolvidas, utilizamos técnicas de cálculo numérico, dentro da linguagem de programação do Flash, que é o Actionscript. Para tanto, utilizaremos um assunto dos mais complexos, a Teoria do Caos, como tema desse trabalho de maneira a exemplificar como implementar técnicas de cálculo dentro desse software e, paralelamente a isso, como utilizar essa ferramenta no desenvolvimento de Objetos de Aprendizagem.

Palavras-chave: Cálculo numérico, Flash, Caos, Objetos de Aprendizagem

1. Introdução

O uso de computadores e recursos de informática no ensino de engenharia e física é cada vez mais comum e em alguns casos necessário, devido às inúmeras facilidades trazidas para a aprendizagem (Fiolhais e Trindade, 2003). O contato dos alunos com o computador e seus recursos, bem como o costume e a facilidade de utilização dos mesmos, é algo que vem se tornando parte do cotidiano das escolas de engenharia. Dentro desse aspecto, a simulação de fenômenos físicos com o auxílio do computador pode ser utilizada com grande eficiência desde que *softwares* de qualidade sejam desenvolvidos para tal propósito.

Diversos assuntos abordados dentro dos cursos de engenharia são muitas vezes abstratos e algumas vezes contra-intuitivos, o que os tornam de difícil compreensão para os alunos. A teoria do caos, por exemplo, trata de fenômenos que apresentam um comportamento que à primeira vista parece ser aleatório e imprevisível, dificultando o entendimento desse assunto por parte dos alunos. Dentro desse contexto, este projeto vem apresentar uma nova ferramenta a ser utilizada no ensino da engenharia e física, utilizando para tanto, o *Macromedia Flash®* no desenvolvimento de objetos de aprendizagem que auxiliem no ensino dessas ciências.

As facilidades apresentadas por esse *software* para a criação de recursos visuais e multimídia em geral, são inúmeras, e simulações físicas podem se tornar muito didáticas com um bom aproveitamento de tais recursos. Contudo, a maior dificuldade aparente seria a resolução de problemas que apresentam uma matemática complexa, que é uma característica presente, por exemplo, em sistemas não-lineares. Essa dificuldade deve-se ao fato do *Flash* ser primeiramente um programa de animações gráficas. Contornando tal situação, este projeto apresenta uma solução para esse problema, mostrando como utilizar o cálculo numérico para simular fenômenos físicos em aplicações criadas com *software* em questão.

Dentro desse contexto, é descrito nesse trabalho como foi feita a implementação do método numérico de Runge-Kutta no *Flash*, a qual é exemplificada com a resolução do sistema caótico do pêndulo duplo, mostrando que com essa idéia, pode-se contornar o problema da solução matemática sem perda de qualidade e interatividade para o programa, de tal forma que o *software* pode ser visto como uma nova ferramenta para a criação de objetos de aprendizagem.

2. Colocando o problema

2.1. A teoria do caos

Dentro da classe dos sistemas matemáticos, temos aqueles que são ditos lineares, aos quais o princípio da superposição é sempre aplicável, e um outro conjunto mais abrangente no qual em geral tal princípio não pode ser aplicado, mas que explicam fenômenos naturais aparentemente aleatórios, os sistemas não-lineares. Entretanto, esses

dois tipos de sistemas estão relacionados de tal forma que para um estudo aprofundado sobre o assunto, eles não podem ser desvinculados.

Para sistemas não-lineares a resposta a um distúrbio não é necessariamente proporcional à intensidade do mesmo e por esse motivo, apesar de serem governados por leis exatas, a evolução com o tempo desses sistemas é bastante sensível às condições iniciais. Esse estado de aparente desordem e irregularidade é o alvo de estudo da Teoria do Caos (Thompson e Stewart, 2002), a qual estende suas ramificações nos mais diversos campos do conhecimento científico, incluindo Física, Medicina, Economia, Matemática, etc.

2.2. Análise do problema

Conforme supracitado, neste projeto será analisado o movimento de um pêndulo duplo, o qual é um exemplo de sistema estudado dentro da teoria do caos e mostraremos a seguir como foi possível a implementação de uma simulação do problema dentro do *software Micromedia Flash®*. Nesse ponto é importante, portanto, que seja feita uma análise matemática do problema em questão. Assim, consideremos o pêndulo duplo mostrado na Fig. (1).

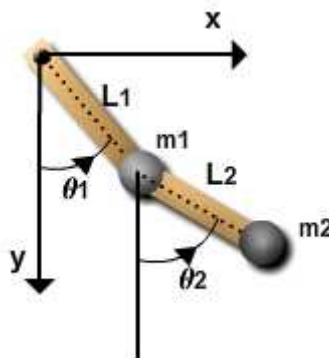


Figura 1. Pêndulo duplo e referenciais adotados.

Como pode-se observar na Fig.1, temos:

x = posição horizontal das massas do pêndulo

y = posição vertical das massas do pêndulo

θ = ângulo do pêndulo com a vertical, sendo positivo no sentido anti-horário.

L = comprimento do braço rígido e de massa desprezível.

Onde temos que x_1, y_1, θ_1 e L_1 são referentes a m_1 e x_2, y_2, θ_2 e L_2 a m_2 .

Analisando a cinemática do problema, pode-se demonstrar que as equações de movimento do pêndulo são:

$$x_1'' = -\theta_1'^2 \sin \theta_1 + \theta_1'' L_1 \cos \theta_1 \quad (1)$$

$$y_1'' = \theta_1'^2 L_1 \cos \theta_1 + \theta_1'' L_1 \sin \theta_1 \quad (2)$$

$$x_2'' = x_1'' - \theta_2'^2 L_2 \sin \theta_2 + \theta_2'' L_2 \cos \theta_2 \quad (3)$$

$$y_2'' = y_1'' + \theta_2'^2 L_2 \cos \theta_2 + \theta_2'' L_2 \sin \theta_2 \quad (4)$$

2.3. Solucionando o problema

A estratégia utilizada para solucionar as Eq. (1), (2), (3) e (4) foi a de utilizar métodos numéricos. Escolheu-se essa forma, pois a implementação do método no programa dava uma maior liberdade de escolha dos parâmetros do problema para o usuário. Suponha, por exemplo, que os programas criados para simular os sistemas caóticos devam ter uma solução analítica para cada conjunto de parâmetros e valores iniciais. Sabe-se, entretanto, que isso nem sempre é possível e quando o é, torna-se algo extremamente trabalhoso, pois obtém-se soluções totalmente diferentes em cada caso, que podem variar para uma simples alteração dos parâmetros do problema e, para facilitar ou até mesmo tornar possível a implementação da aplicação, o programador poderia dar uma menor liberdade de escolha para o usuário.

Métodos numéricos evitam esse tipo de transtorno criando uma solução única aproximada para qualquer conjunto de valores colocados ao problema. Dessa forma, o usuário do programa entra com os valores dos parâmetros e condições iniciais e para qualquer combinação desses valores, o mesmo bloco de código gera a solução apropriada.

Para a resolução das Eq. (1), (2), (3) e (4) foi utilizado o método numérico de Runge-Kutta de ordem quatro (Boyce e DiPrima, 1998). Sua escolha para o projeto foi devida à sua grande precisão, estabilidade e convergência do processo de solução, e pelo fato de poder ser utilizado na resolução de equações diferenciais ordinárias não-lineares.

Os métodos de Runge-Kutta são uma família de métodos numéricos para solucionar equações diferenciais ordinárias. São métodos que podem ser obtidos pela série de Taylor sem a necessidade de calcular qualquer derivada.

Esse método consiste em estimar o valor da função em vários pontos intermediários e o valor solução é encontrado pela média ponderada entre esses pontos. Por sua dedução bastante trabalhosa, limitamo-nos a enunciar sua expressão utilizada no projeto.

Para o pêndulo duplo, temos que a partir das equações (1), (2), (3) e (4), pode-se obter a expressão de θ_1'' e θ_2'' em função de θ_1 , θ_2 , θ_1' e θ_2' utilizando um programa como o *Mathematica*. Assim temos que:

$$\theta_1'' = \frac{-2g(m_1 + m_2) \sin \theta_1 + 2m_2 g \sin(\theta_2) \cos(\theta_1 - \theta_2) - 2 \sin(\theta_1 - \theta_2) m_2 (\theta_2'^2 L_2 + \theta_1'^2 L_1 \cos(\theta_1 - \theta_2))}{L_2(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))} \quad (5)$$

$$\theta_2'' = \frac{2 \sin(\theta_1 - \theta_2) (\theta_1'^2 L_1 (m_1 + m_2) + \theta_2'^2 L_2 m_2 \cos(\theta_1 - \theta_2)) + 2g(m_1 + m_2) (\sin \theta_1 \cos(\theta_1 - \theta_2) \sin \theta_2)}{L_1(2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))} \quad (6)$$

Fazendo que:

$$\theta_1' = w_1 = k(t^i, \theta_1^i, \theta_2^i, w_1^i, w_2^i) \quad (7)$$

$$\theta_2' = w_2 = j(t^i, \theta_1^i, \theta_2^i, w_1^i, w_2^i) \quad (8)$$

$$\theta_1'' = m(t^i, \theta_1^i, \theta_2^i, w_1^i, w_2^i) \quad (9)$$

$$\theta_2'' = n(t^i, \theta_1^i, \theta_2^i, w_1^i, w_2^i) \quad (10)$$

e sendo h o passo de indução e θ_1^i o valor de θ_1 no instante t^i , analogamente para θ_2^i , w_1^i e w_2^i , estamos então nas condições de aplicar o algoritmo de Runge-Kutta, o qual se encontra a seguir.

$$\theta_1^{i+1} = \theta_1^i + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)$$

$$\theta_2^{i+1} = \theta_2^i + \frac{h}{6} (j_1 + 2j_2 + 2j_3 + j_4)$$

$$w_1^{i+1} = w_1^i + \frac{h}{6} (m_1 + 2m_2 + 2m_3 + m_4)$$

$$w_2^{i+1} = w_2^i + \frac{h}{6} (n_1 + 2n_2 + 2n_3 + n_4)$$

$$k_1 = k(t^i, \theta_1^i, \theta_2^i, w_1^i, w_2^i);$$

$$k_2 = k(t^i + \frac{h}{2}, \theta_1^i + \frac{h}{2} k_1, \theta_2^i + \frac{h}{2} j_1, w_1^i + \frac{h}{2} m_1, w_2^i + \frac{h}{2} n_1);$$

$$k_3 = k(t^i + \frac{h}{2}, \theta_1^i + \frac{h}{2} k_2, \theta_2^i + \frac{h}{2} j_2, w_1^i + \frac{h}{2} m_2, w_2^i + \frac{h}{2} n_2);$$

$$k_4 = k(t^i + h, \theta_1^i + h k_3, \theta_2^i + h j_3, w_1^i + h m_3, w_2^i + h n_3);$$

Analogamente a k_p temos para j_p , m_p e n_p , onde $p = 1, 2, 3, 4$.

2.4. Implementação da solução

A solução foi implementada no *Flash* com um passo de indução $h = 0.1$ de forma que para cada posição do pêndulo, o algoritmo do Runge-Kutta executa um passo e imediatamente depois mostra o resultado na tela, ou seja, o valor da posição do pêndulo é plotado assim que calculado, sem guardar valores em vetores.

Foi feito dessa forma devido ao número relativamente grande de cálculos que são feitos em cada passo do Runge-Kutta tendo em vista as Eq. (5) e (6), ficando inviável calcular vários valores para θ_1 e θ_2 inicialmente para serem mostrados posteriormente. A simulação feita dessa forma tem também a vantagem de poder ser executada pelo tempo que o usuário desejar, dando a liberdade dele poder observar padrões apresentados pelo movimento do pêndulo depois de decorrido um grande espaço de tempo.

A simulação desenvolvida não apresentou atrasos perceptíveis para o usuário, indicando que o método de Runge-Kutta associado a esse modelo de implementação se adequou muito bem ao problema do pêndulo duplo. Com isso também, foi possível testar a capacidade do *Flash* de suportar cálculos como os do Runge-Kutta, tendo sido obtido esse resultado bastante satisfatório e estimulante utilizando a seguinte configuração para o computador durante o teste: AMD Athlon™ XP 2200+ 1,79GHz 128MB de RAM que, como pode ser notado, é uma configuração que se encaixa nos padrões dos computadores pessoais atuais. A seguir, temos um trecho simplificado do código utilizado no programa.

```
// Para a função runge  $\theta_1 = xn$ ,  $\theta_2 = yn$ ,  $w_1 = wn$  e  $w_2 = sn$ 
// Essa função executa o cálculo do runge-kutta
function runge(t:Number, xn:Number, yn:Number, wn:Number, sn:Number):Void {
    k1 = k(t, xn, yn, wn, sn);
    j1 = j(t, xn, yn, wn, sn);
    m1 = m(t, xn, yn, wn, sn);
    n1 = n(t, xn, yn, wn, sn);

    k2 = k(t + h/2, xn + (h/2)*k1, yn + (h/2)*j1, wn + (h/2)*m1, sn + (h/2)*n1);
    j2 = j(t + h/2, xn + (h/2)*k1, yn + (h/2)*j1, wn + (h/2)*m1, sn + (h/2)*n1);
    m2 = m(t + h/2, xn + (h/2)*k1, yn + (h/2)*j1, wn + (h/2)*m1, sn + (h/2)*n1);
    n2 = n(t + h/2, xn + (h/2)*k1, yn + (h/2)*j1, wn + (h/2)*m1, sn + (h/2)*n1);

    k3 = k(t + h/2, xn + (h/2)*k2, yn + (h/2)*j2, wn + (h/2)*m2, sn + (h/2)*n2);
    j3 = j(t + h/2, xn + (h/2)*k2, yn + (h/2)*j2, wn + (h/2)*m2, sn + (h/2)*n2);
    m3 = m(t + h/2, xn + (h/2)*k2, yn + (h/2)*j2, wn + (h/2)*m2, sn + (h/2)*n2);
    n3 = n(t + h/2, xn + (h/2)*k2, yn + (h/2)*j2, wn + (h/2)*m2, sn + (h/2)*n2);

    k4 = k(t + h, xn + h*k3, yn + h*j3, wn + h*m3, sn + h*n3);
    j4 = j(t + h, xn + h*k3, yn + h*j3, wn + h*m3, sn + h*n3);
    m4 = m(t + h, xn + h*k3, yn + h*j3, wn + h*m3, sn + h*n3);
    n4 = n(t + h, xn + h*k3, yn + h*j3, wn + h*m3, sn + h*n3);

    teta1 = xn + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
    teta2 = yn + (h/6)*(j1 + 2*j2 + 2*j3 + j4);
    omega1 = wn + (h/6)*(m1 + 2*m2 + 2*m3 + m4);
    omega2 = sn + (h/6)*(n1 + 2*n2 + 2*n3 + n4);
}
// A função anima é executada a cada 3milissegundos e é responsável por executar
// o algoritmo do runge-kutta e atualizar a tela para os novos valores.
var idAnima:Number = setInterval(anima, 3);
function anima():Void {
    runge(tempo, teta1, teta2, omega1, omega2);
    pendulo(); // Acerta a posição do pêndulo

    // Plotar o gráfico
    mcGrafico.lineTo(mcGraficoMaior._x+30*teta2, mcGraficoMaior._y+30*teta1);

    // Atualização da tela
    updateAfterEvent();
    tempo++;
}
}
```

3. O programa construído

O programa criado consiste em um pêndulo duplo no qual o usuário imprime as condições iniciais, escolhendo os valores das massas – m_1 e m_2 , posição inicial do pêndulo, que é dada por θ_1 e θ_2 e aceleração da gravidade, g . Essa interação é feita pelo movimento do *mouse*, que pode ser arrastado pelas barras que se encontram na parte inferior do programa, como pode ser visto na Fig. 2, ou inserindo os valores diretamente pelo teclado.

Depois de acertados os parâmetros supracitados, basta clicar no botão “iniciar” que o pêndulo passa a executar o seu movimento, sendo que paralelamente a esse fato é gerado um gráfico que mostra o comportamento de θ_1 em função de θ_2 durante a simulação.

Além disso, encontra-se na interface do programa um botão “Ajuda” que ao ser clicado mostra o funcionamento do programa e a física que está por trás do fenômeno representado, bem como um botão “Reiniciar” que traz a simulação de volta para o início.

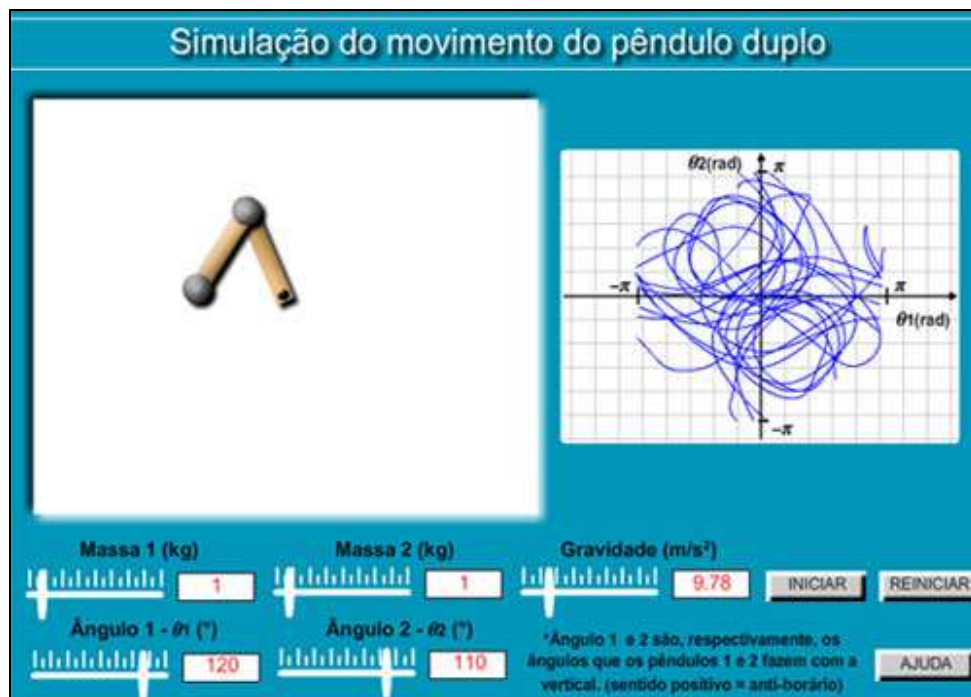


Figura 2. Objeto de aprendizagem simulando o movimento do pêndulo duplo

4. Métodos Numéricos e o Actionscript

O *Macromedia Flash* é uma ferramenta de desenvolvimento com muitos recursos, possibilitando inclusive a entrega de uma variedade de conteúdos dinâmicos na *web*. Uma pesquisa realizada pela *Macromedia* informa que o *Flash Player*, *plug-in* necessário para a execução de animações *Flash*, está instalado em 98% dos *browsers* conectados à internet. Ao contrário de um código HTML estático, uma aplicação feita no *Macromedia Flash* pode responder rapidamente sem a necessidade de se fazer algum processamento no servidor. Essa ferramenta atende aos requisitos para os nossos estudos de caso para criação de Objetos de Aprendizagem em física.

É importante observar também que a linguagem utilizada no *Flash*, o *ActionScript*, é de fácil utilização e tornou o ambiente do *Flash* muito mais interessante de se trabalhar, pois faz com que se possa desenvolver programas bem elaborados com ambientação gráfica.

Uma novidade trazida por este projeto é a implementação de métodos numéricos em aplicações *Flash*, que só foi possível devido à evolução da linguagem *Actionscript* que agora se encontra na sua versão 2.0. Na Fig. (3), temos uma comparação feita através de gráficos gerados pelo próprio *Flash*, da solução dada pelo Runge-Kutta com 500 iterações e da solução analítica para o problema do sistema massa-mola realizando um MHS que pode ser descrito pela Eq. (11).

$$m \frac{d^2 y}{dt^2} + ky = 0 \tag{11}$$

Sendo y a coordenada vertical e os parâmetros: constante da mola $k = 10 \text{ N/m}$ e massa do bloco $m = 5 \text{ kg}$. Pode-se notar que a diferença entre as soluções só passa a ser ligeiramente perceptível na parte final dos gráficos onde o número de iterações já executadas pelo Runge-Kutta está próximo de 500, sendo mesmo assim uma diferença muito suave.

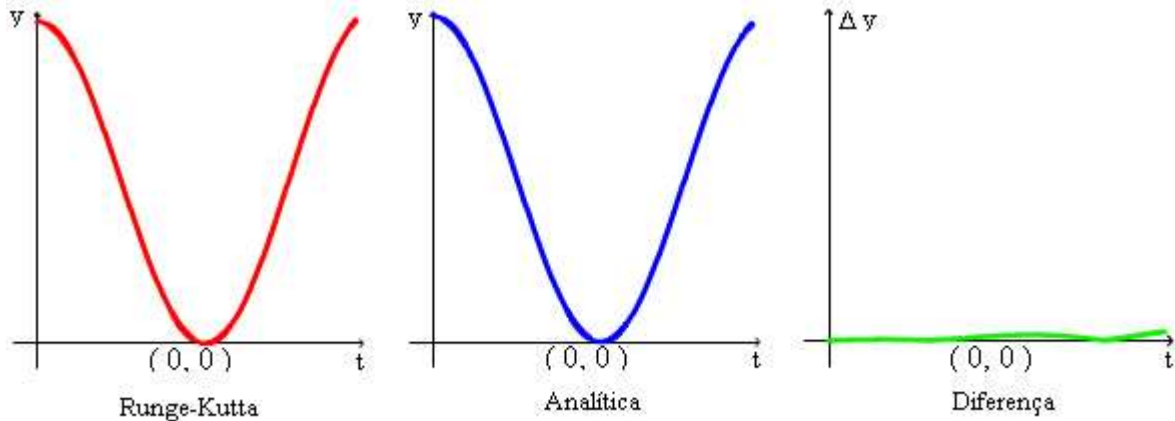


Figura 3. Comparação entre solução numérica e analítica utilizando *Actionscript*.

Grandes avanços podem ser alcançados com essa técnica tanto na área de simulação quanto na utilização em tecnologias de ensino, com a construção de objetos de aprendizagem. Tendo em vista que somente com a utilização do Runge-Kutta para a resolução das equações diferenciais foi possível realizar o projeto, pode-se notar assim, a importância da implementação dos métodos numéricos em *Actionscript*. Uma consequência direta dessa inovação também, é que agora sem a necessidade de hipóteses simplificadoras, o *software* pode ser explorado ao máximo, de forma a aproveitá-lo na criação de objetos de aprendizagem e simuladores.

5. O Flash como ferramenta na criação de objetos de aprendizagem.

Utilizando a idéia de criar simulações físicas utilizando o *Flash* como ponto de partida, apresentamos aqui outros exemplos de como desenvolveu-se a partir do software em questão, alguns outros objetos de aprendizagem.

O primeiro deles segue os mesmos padrões do pêndulo duplo apresentado e também é um fenômeno oscilatório estudado pela teoria do caos, o qual é comumente chamado de pêndulo caótico. Trata-se de um corpo de massa m , o qual está suspenso por um cabo ideal de comprimento $L = 1m$. Esse corpo está sujeito três torques:

- $-L m g \text{ sen } \theta$, realizado pela força gravitacional, onde g é a aceleração da gravidade e θ o ângulo que o pêndulo faz com a vertical.
- $-b \omega$, realizado por uma força amortecedora devido ao fato do corpo estar mergulhado a um líquido viscoso, onde b é a constante de amortecimento e ω , a aceleração angular.
- $A \cos(k t)$, devido ao fato do eixo do pêndulo estar acoplado a um pistão, onde A é a amplitude máxima do torque e k a frequência angular de aplicação do torque.

Igualando a soma desses torques ao momento de inércia rotacional total $I = m L^2$, temos a equação que rege o movimento do pêndulo caótico:

$$\theta'' = -\frac{g \text{ sen } \theta}{L} + \frac{-b\theta' + A \cos(kt)}{mL^2} \quad (12)$$

Para a implementação desse problema foi utilizado da mesma forma que no pêndulo duplo, o método numérico de Runge-Kutta de quarta ordem. Como esse método de implementação já foi demonstrado nesse trabalho, omitiremos aqui a resolução da Eq. (12).

Na Fig. (4) encontra-se uma tela da simulação desenvolvida para este exemplo. Novamente temos, de um lado o experimento simulado e do outro um gráfico relativo ao mesmo que é gerado enquanto o fenômeno ocorre. O importante de mostrarmos esse exemplo é deixar claro que com a associação da idéia de se usar cálculo numérico no *Flash*, podemos facilmente desenvolver outros tipos de simulações de forma a serem abordados os mais diversos assuntos, principalmente nos campos da Engenharia e da Física.

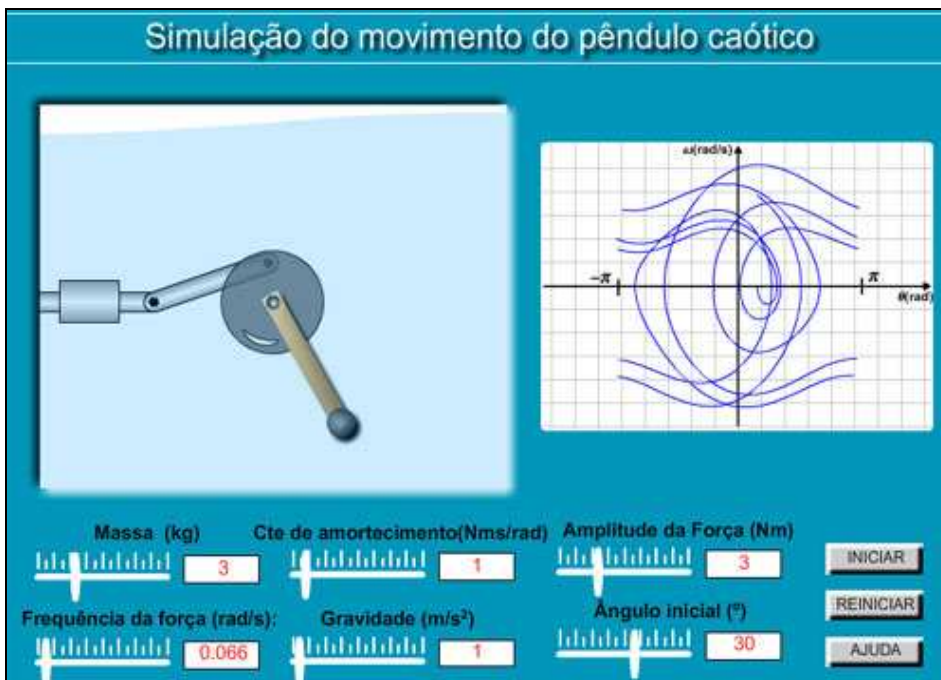


Figura 4. Objeto de Aprendizagem simulando o movimento de um pêndulo caótico.

Um outro modo de utilização do *software* em questão no desenvolvimento de objetos de aprendizagem é criando-se pequenos filmes auto-explicativos, de maneira que eles possam ser utilizados até mesmo durante o primeiro contato de um aluno com um determinado assunto. Como exemplo disso, foi desenvolvida uma animação (Fig. (5)) que pode ser controlada pelo usuário, ou seja, este pode voltar os quadros quando não entender completamente o que foi ensinado ou avançar se já possuir o conhecimento que está sendo transmitido. Essa animação aborda o movimento oscilatório com amortecimento viscoso de um sistema massa-mola e demonstra passo a passo como, a partir da observação do fenômeno, deduzir a equação que rege o movimento, deixando bem claro a interpretação física de cada parâmetro da mesma.

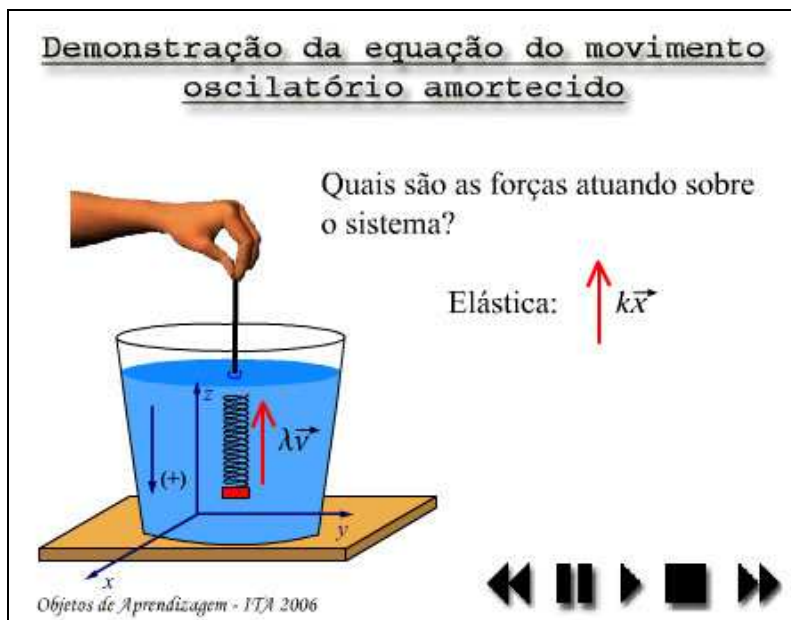


Figura 5. Objeto de Aprendizagem apresentando uma mini Aula sobre movimento amortecido.

Desenvolveu-se também um aplicativo que simula esse mesmo movimento, de forma que logo após o usuário ter tido uma explicação sobre o tema, ele tem a oportunidade de testar o que aprendeu em uma simulação, observando

como uma variação dos parâmetros iniciais influencia o sistema com o tempo, fixando os conceitos físicos envolvidos no fenômeno estudado. Na Fig. (6) temos uma tela do programa desenvolvido.

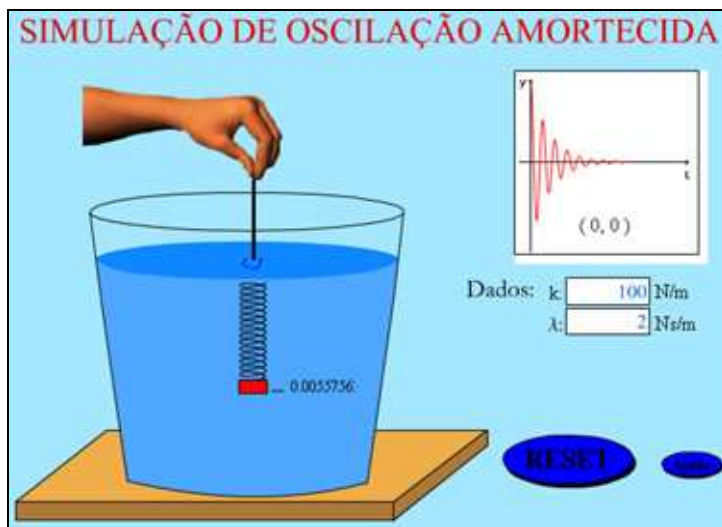


Figura 6. Objeto de aprendizagem que simula o movimento oscilatório com amortecimento viscoso.

Outras simulações desenvolvidas também nesse projeto estão representadas a seguir por uma imagem da tela de cada programa.

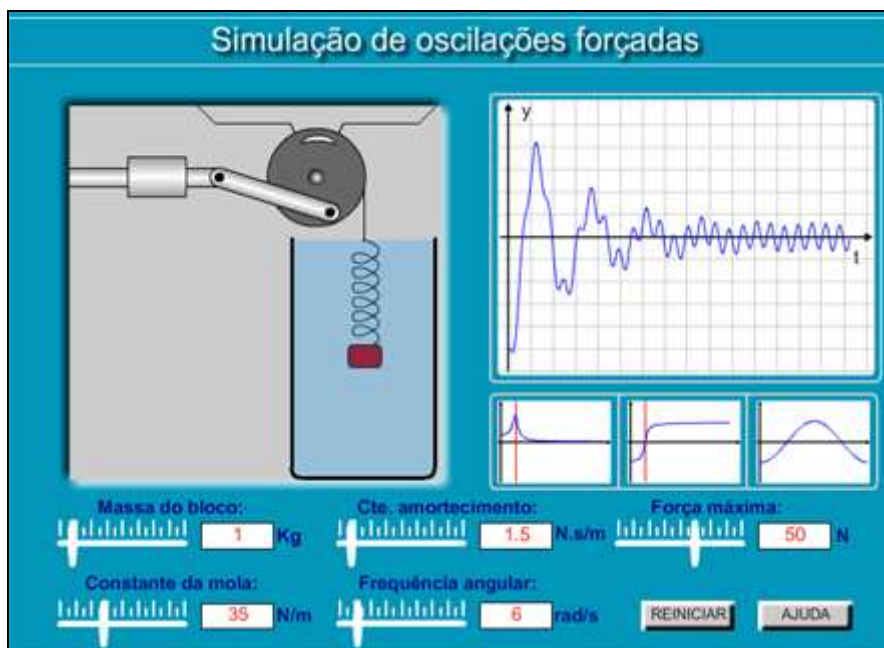


Figura 7. Objeto de Aprendizagem que simula um movimento oscilatório amortecido e forçado.

É interessante notar o fato de que uma característica do *Flash* auxilia especificamente essas aplicações sobre movimento oscilatório que utilizam um sistema massa-mola: como a mola é um produto do desenho vetorial (Pizzi, 2003), sua deformação durante a animação não ocasiona perda de qualidade gráfica, o que não aconteceria com um *bitmap*, por exemplo. Nesse caso, foi desenhada uma mola em tempo de criação e sua escala vertical modificada durante a animação, consumindo muito pouco processamento comparado ao fato de ter de gerar uma nova visualização da mola para cada posição.

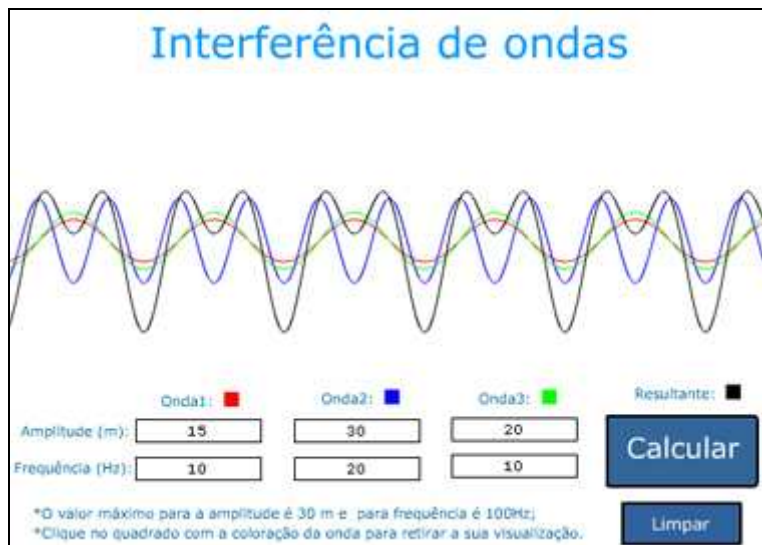


Figura 8. Objeto de Aprendizagem que simula interferência de ondas planas.

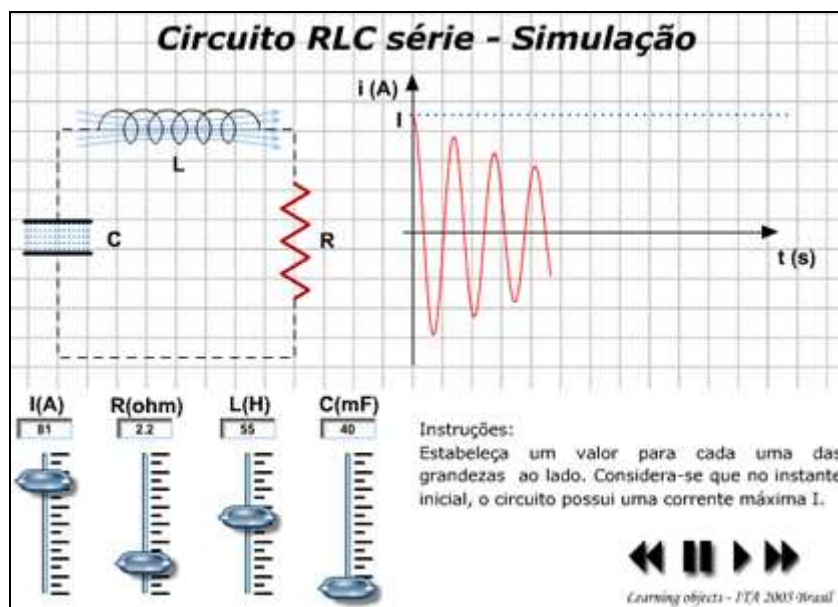


Figura 9. Objeto de Aprendizagem que simula um circuito RLC série



Figura 10. Objeto de Aprendizagem que simula o efeito Doppler de ondas sonoras no ar.

6. O Uso Pedagógico

Aplicações gráficas como a do movimento do pêndulo duplo ou do pêndulo caótico, nas quais pode-se observar a simulação do fenômeno físico em questão e, em tempo real, visualizar o traçado de um gráfico referente ao sistema, podem tornar a aula mais atrativa e dinâmica, onde o professor e os estudantes, em vez do trabalho de fazer cálculos e mais cálculos, podem concentrar – nestas ocasiões - seus esforços nos aspectos conceituais do assunto tratado. Além disso, podem ser feitas perguntas aos alunos que os induzam a pensar e a formular hipóteses, das quais a maior parte pode ser testada no ato. Toda esta atividade é viável mesmo em turmas com número elevado de alunos, neste caso pode ser utilizado um projetor multimídia, ou realiza-la em um laboratório de informática onde cada aluno pode ter acesso ao Objeto de Aprendizagem. Cabe citar que outra grande vantagem do *Flash* é a possibilidade de executar seus aplicativos via internet, facilitando o contato do aluno com o programa e estimulando assim o seu aprendizado.

7. Conclusão

Como um resultado dessa utilização do *Flash*, podemos apresentar o fato de que programas desenvolvidos estão sendo utilizados nas aulas de física do primeiro e segundo ano de engenharia do Instituto Tecnológico de Aeronáutica, inovando o ensino da matéria e colocando o aluno mais próximo de uma visão prática acerca do assunto. Os resultados foram muito bons, despertando o interesse dos alunos e tirando dúvidas sobre a representação real daquilo que estudavam em teoria.

A principal dificuldade encontrada para a confecção dos objetos de aprendizagem foi exatamente a resolução das equações diferenciais que representam o fenômeno físico. Dessa forma, tivemos a idéia de utilizar métodos numéricos dentro do *Flash* para resolver esse problema, cujos resultados são bastante satisfatórios, como podemos perceber diante da análise de processamento e precisão feitas e da proximidade com a realidade alcançada pelos simuladores. Estes resultados obtidos indicam que o Runge-Kutta contornou muito bem o problema encontrado, de forma que os objetos de aprendizagem criados utilizando tal método ficaram perfeitamente funcionais.

8. Agradecimentos

Os autores agradecem o incentivo do CNPq no desenvolvimento de novas tecnologias de ensino, pois graças à bolsa de iniciação científica fornecida pelo PIBIC, foi possível ter início o projeto, bem como neste momento continuar a desenvolvê-lo.

9. Referências

- BOYCE, W.E. e DiPRIMA, R.C., 1998, Equações Diferenciais Elementares e Problemas de Valores de Contorno, 6a.edição, Livros Técnicos e Científicos Editora S.A., Rio de Janeiro.
- FIOLHAIS, C. e TRINDADE, J., 2003, Física no computador: o computador como uma ferramenta no ensino e na aprendizagem das ciências físicas. Rev. Bras. Ens. Fis., vol.25, no.3, p.259-272. ISSN 0102-4744
- PIZZI, M., 2003, Dominando Macromedia Flash MX. Rio de Janeiro: Editora Ciência Moderna Ltda.
- THOMPSON, J. M. T. e STEWART, H. B., 2002, Nonlinear Dynamics and Chaos, John Wiley & Sons, Chichester, ISBN: 0-471-87684-4