



SIMULAÇÃO 3D DE FENÔMENOS FÍSICOS

Thiago Mendes Campos – thiagododo@unis.edu.br

UNIS-MG, Instituto de Tecnologia e Ciências Exatas

Av. Coronel. José Alves, 256 – Bairro Vila Pinto

37010-540 – Vaginha - MG

Marcelo Maneschy Horta Barreira– marcelo@unis.edu.br marcelomhb@pucpcaldas.br

UNIS-MG, Instituto de Tecnologia e Ciências Exatas

Av. Coronel. José Alves, 256 – Bairro Vila Pinto

37010-540 – Vaginha - MG

PUC-Minas, *campus* de Poços de Caldas, Engenharia Elétrica ênfase em Telecomunicação

Av. Pe. Francis Cletus Cox, 1661 – Bairro Jardim Country Club

CEP 37701-355 – Poços de Caldas – Minas Gerais

***Resumo:** Este trabalho reporta o uso da modelagem e simulação de fenômenos físicos em três dimensões como uma ferramenta pedagógica no ensino de Física nos Cursos de Engenharia. Em particular usa-se uma linguagem apropriada para tal, VPython, por ser de fácil aprendizado e portanto viável de ser empregada nos primeiros semestres onde a Física básica é ensinada. O principal argumento é de que o aluno, com o uso da modelagem e da programação, começa a entender melhor o significado das equações da física, tem melhor idéia das aproximações feitas para compor o modelo, tornando-se assim uma ferramenta importantel a ser adicionada as capacidades analíticas do aluno. Este trabalho se fundamenta fortemente no trabalho desenvolvido pelo grupo do Center For Innovation in Learning da Carnegie Mellon University, em específico, no trabalho de Bruce Sherwood e Ruth Chabay.*

***Palavras-chave:** Novas tecnologias, Modelagem e Simulação, Ensino em Engenharia*

1. INTRODUÇÃO

O uso de novas tecnologias na educação vem sendo um assunto de grande debate no meio educacional. Em especial, há um grande favorecimento em torno do uso da Internet como um meio de pesquisa bibliográfica, ou seja, incentiva-se a pesquisa bibliográfica eletrônica. No entanto, isto não se reflete naturalmente como um uso muito diferenciado desta tecnologia quando comparado com a pesquisa bibliográfica real. Um outro uso do computador no ensino de ciências tem sido no uso de ambientes virtuais de laboratório de Física, ou seja, simulações de sistemas físicos, onde o aluno pode observar determinado fenômeno e ler diretamente no display as mudanças da quantidades físicas envolvidas. Tal procedimento é um procedimento válido no âmbito de demonstrações em salas de aula com acesso computador e projeção e também para estudos independentes. No entanto, como o próprio nome já diz, é uma simulação, edeve ficar a dúvida na mente do estudante se aquele fenômeno realmente acontece da forma como está representado na tela do computador e, o que é mais importante

ainda enquanto ensino de física, será que a Natureza se comporta tão precisamente assim, ou aquela representação é um modelo repleto de aproximações?

Este trabalho é um breve relato do início de um projeto de Iniciação Científica em que o objeto de pesquisa é o uso da programação computacional como ferramenta pedagógica no ensino das disciplinas introdutórias de Física, que compõem as matrizes curriculares dos cursos de engenharia. A idéia central do trabalho é envolver o aluno no processo de modelagem e simulação de diversas situações físicas estudadas nestes cursos. A diferença entre esta abordagem e a de apenas usar os aplicativos já existentes, está exatamente em que, para fazer a modelagem do sistema, o aluno terá que discutir aproximações a serem feitas, tr  que refletir sobre as equa es a serem usadas e ao executar um determinado loop do programa, ele ter  que entender fisicamente o que determinada equa o realmente “diz”.

Nada disto   muita novidade e ainda ficaria a quest o, al m de ter que aprender a disciplina F sica o aluno ter  que aprender uma linguagem de programa o, que geralmente em nosso meio s o FORTRAN ou C, o que torna a tarefa question vel quanto a efetividade. No entanto, uma linguagem VPython (Visual Python) foi desenvolvida a partir da linguagem Python, por educadores do Departamento de F sica da Carnegie Mellon University, e usada para os fins mencionados acima. Nela, as simula es aparecem em tr s dimens es em um ambiente gr fico bem excitante para o aluno, e a parte dif cil, a de escrever o c digo,   realmente muito simplificada na linguagem. A sensa o   a de que o aluno est  realmente brincando com as equa es. Mais, como o resultado da simula o   mostrado tela graficamente em 3 dimens es, a visualiza o   mais uma forma de verificar se o que voc  espera que aconte a em um fen meno f sico est  realmente sendo simulado no c digo, e o aluno percebe ent o quais aproxima es foram relevantes para a modelagem. Esta brincadeira de “situa o esperada” sempre   uma quest o importante quando resolvemos um problema anal ticamente e para a qual o aluno geralmente n o adquire o h bito de inspecionar. Aqui este   um procedimento inevit vel. Portanto o aprendizado da linguagem   realmente r pido, e pode ser confirmado no trabalho do bolsista de Inicia o Cient fica e, sempre as discuss es em nossos encontros concentravam-se em quest es sobre a f sica do problema, e comparativamente pouco na linguagem.

A id ia do trabalho   construir um uma s rie de simula es de fen menos f sicos percorrendo os cont dos abordados nos cursos introdut rios de F sica, para que, uma vez j  conhecendo as dificuldades de cada programama o, o professor possa prop -los como problemas dentro destas disciplinas e torne facilitado a administra o deste trabalho junto  s turmas, tradicionalmente grandes.

Nas pr ximas se es seguem as etapas percorridas neste trabalho de Inicia o Cient fica.

2- A LINGUAGEM VPYTHON

Primeira Etapa. Inicialmente o aluno tomou contato com a linguagem de programa o VPython, que   um aprimoramento de linguagem de programa o cT, f cil de aprender e poderoso em simula es 3D, ap s o desenvolvimento pelo *Center for Innovation in Learning* na *Carnegie Mellon University*. Neste per odo de aproximadamente um m s, o aluno estudou a linguagem e as aplica es j  desenvolvidas pelo grupo da Carnegie Mellon com o objetivo do aprendizado em programa o.

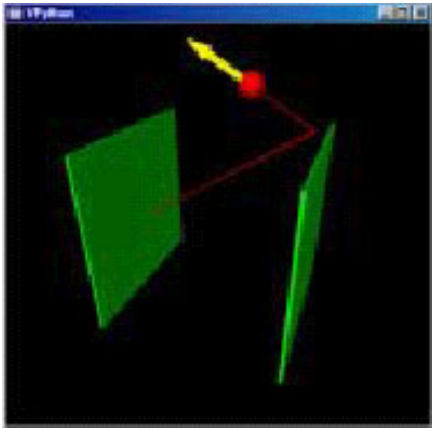
Abaixo mostramos dois dos programas estudados para fim familiariza o com a linguagem. O primeiro   do tutorial “VPython Introduction” que simula uma bola rebatendo entre duas paredes. O segundo   programa “Orbits”, existente no site do curso de F sica da Carnegie Mellon University, e que simula a  rbita de uma estrela bin ria.

2.1 Bola rebatendo em duas paredes

```

from visual import *
ball = sphere(pos=(-5,0,0), radius=0.5, color=color.red)
wallR = box(pos=(6,0,0), size=(0.2,12,12), color=color.green)
wallL = box(pos=(-6,0,0), size=(0.2,12,12), color=color.green)
dt = 0.05
ball.velocity = vector(2,1.5,1)
bv = arrow(pos=ball.pos, axis=ball.velocity, color=color.yellow)
ball.trail = curve(color=ball.color)
while (1==1):
    rate(100)
    ball.pos = ball.pos + ball.velocity*dt
    if ball.x > wallR.x:
        ball.velocity.x = -ball.velocity.x
    if ball.x < wallL.x:
        ball.velocity.x = -ball.velocity.x
    bv.pos = ball.pos
    bv.axis = ball.velocity
    ball.trail.append(pos=ball.pos)

```



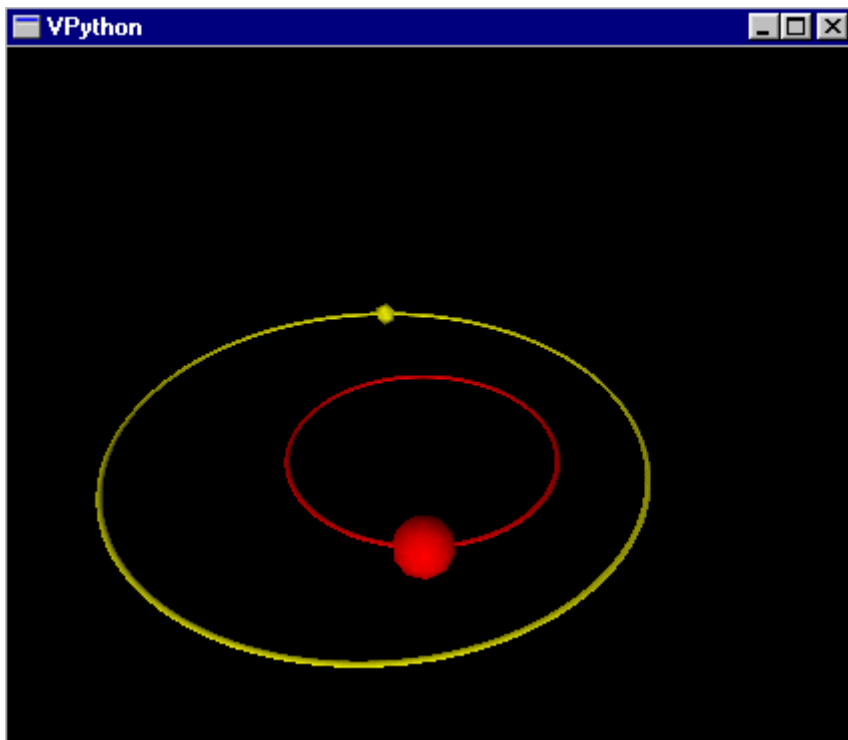
2.2 Órbitas

```

from visual import *
giant = sphere()
giant.pos = vector(-1e11,0,0)
giant.radius = 2e10
giant.color = color.red
giant.mass = 2e30
giant.p = vector(0, 0, -1e4) * giant.mass
dwarf = sphere()
dwarf.pos = vector(1.5e11,0,0)
dwarf.radius = 1e10
dwarf.color = color.yellow
dwarf.mass = 1e30
dwarf.p = -giant.p
for a in [giant, dwarf]:
    a.orbit = curve(color=a.color, radius = 2e9)
dt = 86400
while 1:

```

```
rate(100)
dist = dwarf.pos - giant.pos
force = 6.7e-11 * giant.mass * dwarf.mass * dist / mag(dist)**3
giant.p = giant.p + force*dt
dwarf.p = dwarf.p - force*dt
for a in [giant, dwarf]:
    a.pos = a.pos + a.p/a.mass * dt
    a.orbit.append(pos=a.pos)
```



2- SELEÇÃO DE TÓPICOS

Segunda Etapa. O segundo momento foi o de selecionar os tópicos em Física nos quais os alunos tem maior dificuldade de visualização dos fenômenos e sua aplicação nos recentes desenvolvimentos da Física e em particular da Física da Ciência de Materiais. Foi programada uma simulação, que de agora em diante chamaremos de experimento, para cada um dos temas:

- 1) Força gravitacional – exemplo da relação força inverso quadrado e energia potencial
- 2) Força elétrica
- 3) Lei de Hooke
- 4) Conservação de energia
- 5) Momento angular
- 6) Entropia
- 7) Gases
- 8) Campo elétrico
- 9) Potencial elétrico

- 10) Campo magnético
- 11) Ondas
- 12) Radiação eletromagnética
- 13) Vibrações
- 14) Estrutura da matéria e mecânica quântica.
- 15) Ótica clássica e quântica

3- SIMULAÇÕES DESENVOLVIDAS

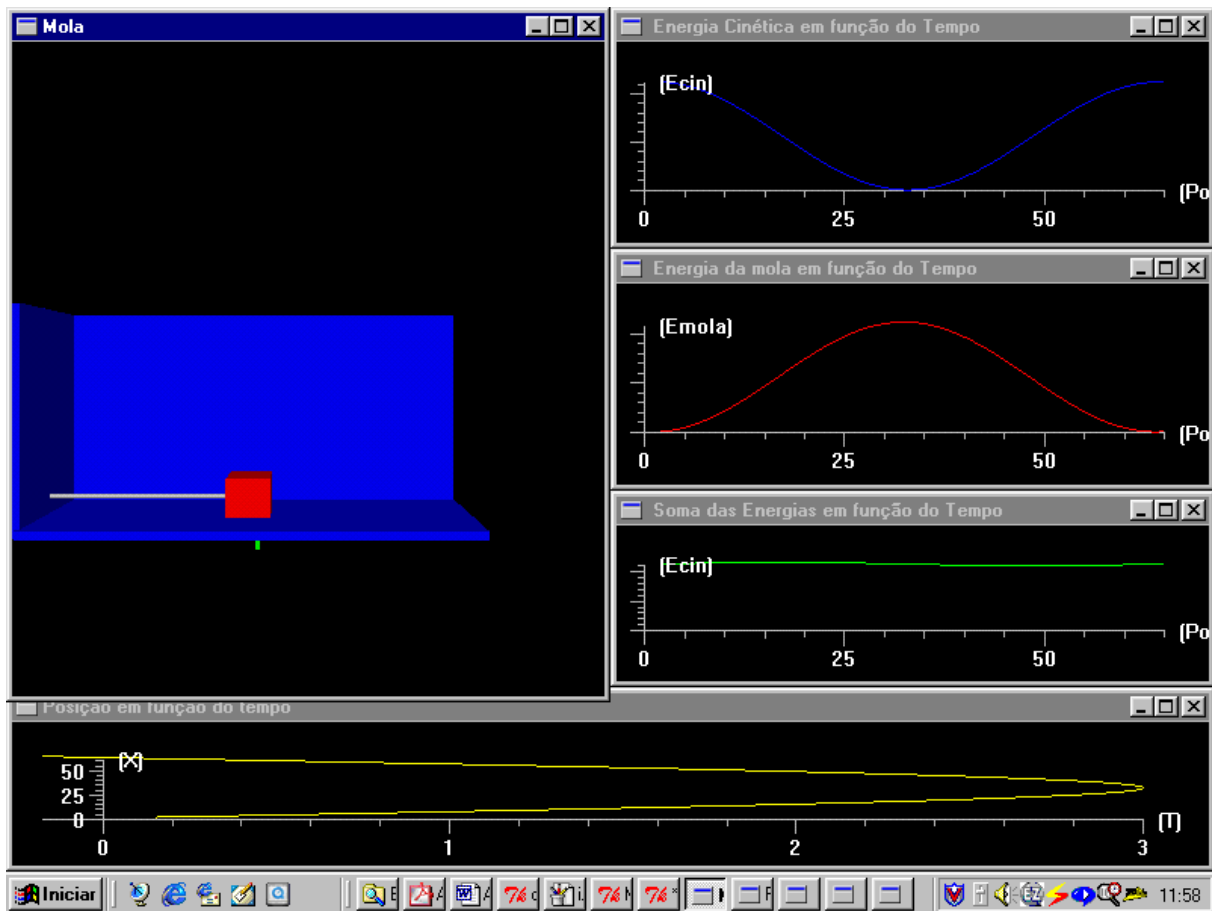
Terceira Etapa. O terceiro momento foi o de efetivamente realizar a modelagem matemática dos fenômenos físicos propostos nos temas definidos da etapa anterior. Para isto o bolsista resolveu os seguintes problemas propostos no livro “Matter and Interactions”, que é o resultado de pesquisa em educação desenvolvido pelos professores Ruth Chabay e Bruce Sharewood, com o intuito de trazer temas modernos da Física para os cursos introdutórios, utilizando apenas os princípios fundamentais da Física, e tem forte apelo a modelagem e simulação como ferramenta para o aprendizado.

Foram realizadas modelagens de 13 problemas diferentes do livro mencionado acima (porém conectados em um eixo comum de entendimento dos princípios fundamentais da Física e na modelagem de estrutura da matéria), pudemos simular em 3D:

- 1) Mover um objeto através da tela de um computador deixando o seu rastro,
- 2) Órbita terrestre em torno do Sol,
- 3) Sistema massa-mola e integração numérica do seu movimento,
- 4) Viagem Terra-Lua: cálculo do trabalho realizado,
- 5) Energia no sistema massa-mola,
- 6) O experimento de Rutherford,
- 7) Momento angular em órbitas elípticas,
- 8) Distribuição de probabilidade: distribuição de 100 quanta entre dois blocos contendo 200 e 300 osciladores cada; Entropia: número de maneiras de arranjar a energia; Temperatura: Gráfico da temperatura em função do número de quanta de cada bloco; Capacidade calorífica: modelo de um bloco com 35 átomos (105 osciladores) com até 300 quanta de energia e comparação da constante de mola efetiva e os resultados experimentais,
- 9) Campo elétrico de um dipolo,
- 10) Campo elétrico de uma barra uniformemente carregada,
- 11) Cálculo do campo magnético de um solenóide,
- 12) Modelo computacional de um ciclotron,
- 13) Modelo computacional de uma onda eletromagnética.

Vamos agora transcrever o código de duas destas simulações, como exemplo.

3.1 Sistema massa-mola e integração numérica de seu movimento



```

from visual import *
from visual.graph import*
L=6
#scene.width = 430
#scene.height = 700# ajustar tela da sena
scene = display(title="Mola", width=400, height=460, x=0, y=0,range=L,
center=(L/5.,L/2.,L/2.))
#scene=display(length=450,height=100)
fundo = box(pos=(0,-0.5,0),length=10, height=0.2, width=3, color=color.blue)
fundo = box(pos=(0,1.9,-1.5),length=10, height=5, width=0.2, color=color.blue)
fundo = box(pos=(-5,1.9,-0.1),length=0.2, height=5, width=3, color=color.blue)#fundos azuis
marca = box(pos=(0,-1,0),length=0.1,height=0.5,width=0.1,color=color.green)
peso = box (length = 1, heigth = 1, width = 1, color=color.red)# bloco vermelho

peso.mass = 4 # massa do bloco
peso.p = vector(1.5,0,0) * peso.mass# momento do bloco velocidade * massa

dt = 0.1# intervalo de tempo tem que ser pequeno porque senão não roda direito
k = 1 # constante da mola
x = peso.pos

```



```
mola = curve(pos=[(-5,0,0),peso.pos],radius=0.05)# linha imitando mola
#trail=curve(pos=(-5,0,0))
```

```
grafico = gdisplay(title="Posição em função do
tempo",height=130,x=0,y=450,xtitle="(T)", ytitle="(X)")
grafico = gcurve(color=color.yellow)# Tentativa de gráfico
graficoEcin = gdisplay(title=" Energia Cinética em função do
Tempo",height=160,x=400,y=-500,xtitle="(Pos)",ytitle="(Ecin)")
graficoEcin = gcurve(color=color.blue)
graficoEmola = gdisplay(title=" Energia da mola em função do
Tempo",height=160,x=400,y=160,xtitle="(Pos)",ytitle="(Emola)")
graficoEmola = gcurve(color=color.red)
graficoEtotal = gdisplay(title=" Soma das Energias em função do
Tempo",height=130,x=400,y=320,xtitle="(Pos)",ytitle="(Ecin)")
graficoEtotal = gcurve(color=color.green)
```

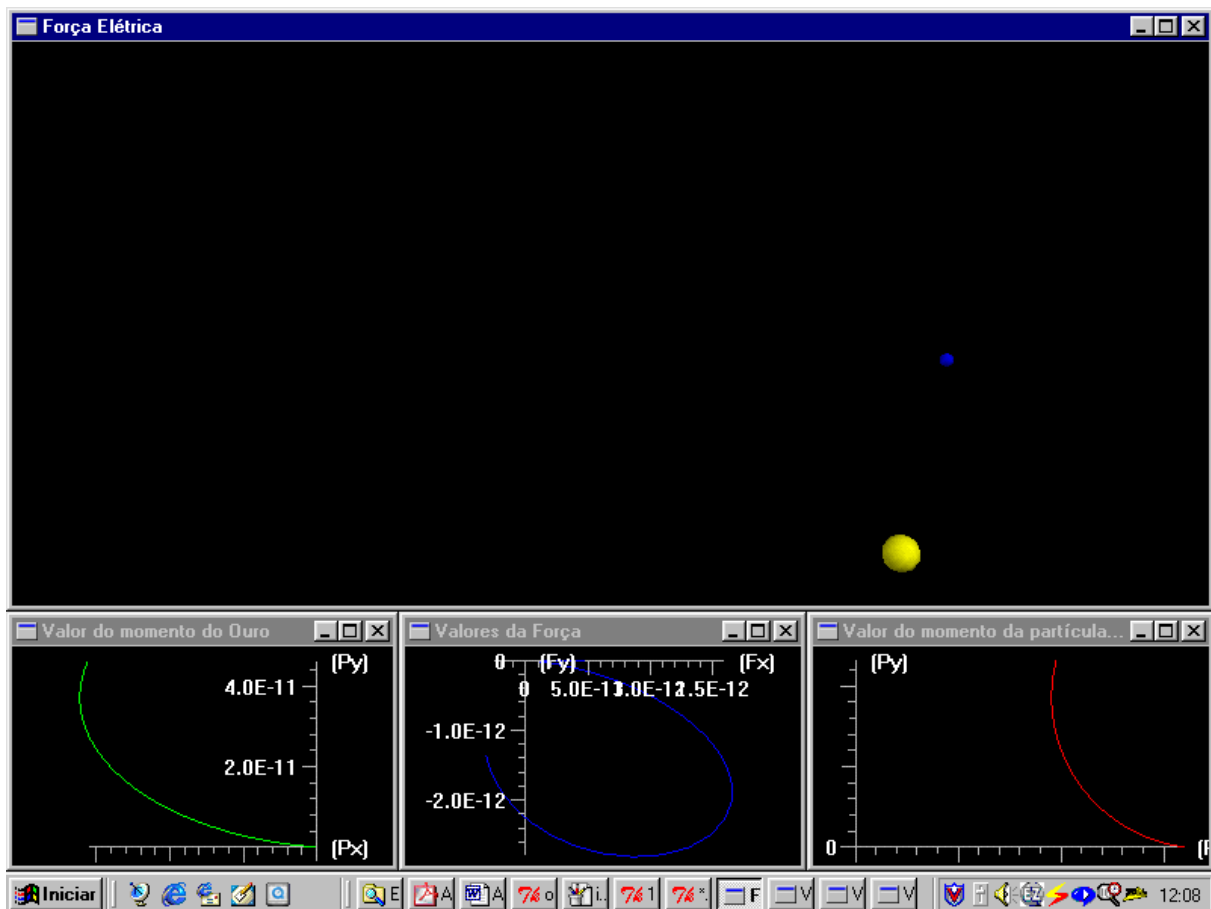
```
a=1
b=1
```

```
t=vector(0,1,0)# contagem de loop
h=vector(0,1,0)
```

```
while 1:
```

```
rate(10)#frequencia baixa porque senão não roda direito
F = -k*x# força de um amola
peso.p = peso.p + F * dt # atualização do momento com a força
peso.pos = peso.pos + peso.p/peso.mass * dt#atualização da posição com o momento
mola.color=color.green
mola.visible=0# apagar linha(mola) anterior
mola = curve(pos=[(-5,0,0),peso.pos],radius=0.05)#Criação da linha(mola)
mola.visible = 1# Linha (mola) visível
t=t+h#contagem de loop
a=a+b
gra = peso.pos + t
Ecin = 0.5*peso.mass*mag(peso.p/peso.mass)**2
Emola = 0.5*k*mag(peso.pos)**2
Etotal = Ecin + Emola
print " Energia Cinética "
print Ecin
print " Energia da Mola "
print Emola
print " Energia Total "
print Etotal
grafico.plot(pos=(gra))
graficoEcin.plot(pos=(a,Ecin,0))
graficoEmola.plot(pos=(a,Emola,0))
graficoEtotal.plot(pos=(a,Etotal,0))
```

3.2 Espalhamento Rutherford



```

from visual import*
from visual.graph import*
L=1.2
scene = display (title="Força
Elétrica",width=800,height=400,x=0,y=0,range=L,center=(L/5,L/2,L/2))

#PalfaX = -1 # Para alterar a posição inicial da Particula Alfa altere esses tres campos
#PalfaY = 0.3
PouroX = 1
Palfa = sphere (radius = 0.02,color=color.blue)# Montagem das Bolas
Pouro = sphere (pos=(PouroX,0,0),radius = 0.05,color=color.yellow)
Palfa.mass = 2e-9 # Massas das partículas
Pouro.mass = 79e-9
Celetrica = 1.602e-19
Qalfa = 2 * Celetrica # Carga das partículas
Qouro = 79 * Celetrica
EPalfa = vector(16e-13,0,0) # Energia inicial da Partícula Alfa
EPouro = vector(0,0,0)
VPalfa = sqrt(2*EPalfa/Palfa.mass ) # Cálculo da velocidade inicial da Partícula alfa no
eixo X )
VPouro = sqrt(2*EPouro/Pouro.mass)
Palfa.p = Palfa.mass * VPalfa # Cálculo do momento
Pouro.p = Pouro.mass * VPouro

```




```
Palfa.p1 = Palfa.mass * VPalfa # Cálculo do momento
Pouro.p1 = Pouro.mass * VPouro
grafico=gdisplay(title="Valor do momento da partícula
alfa",height=500,x=530,y=400,xtitle="(Px)", ytitle="(Py)")
grafico=gcurve(color=color.red) # Gráfico do momento da partícula alfa
graficoF=gdisplay(title="Valores da
Força",height=500,width=270,x=260,y=400,xtitle="(Fx)", ytitle="(Fy)")
graficoF=gcurve(color=color.blue)# Gráfico da Força
graficoP=gdisplay(title="Valor do momento do
Ouro",height=500,width=260,x=0,y=400,xtitle="(Px)", ytitle="(Py)")
graficoP=gcurve(color=color.green)# Gráfico do momento do ouro
#Dist = Phelio.pos - Palfa.pos # Distancia das partículas
K = 9e22 # Constante elétrica
dt = 1

while 1:
    rate(10) # Frequencia
    if scene.mouse.events:
        Palfa.p = Palfa.p1 # Para quando clicar denovo ele voltas com a Energia inicial
        Pouro.p = Pouro.p1
        Palfa.pos = scene.mouse.getclick().pos
        Palfa.pos = Palfa.pos - vector(0,0,0.6)

    Dist = Pouro.pos - Palfa.pos # Cálculo da distancia vetorial das duas partículas
    F = (K * Qalfa * Qouro*Dist)/mag(Dist)**3 # cálculo da força

    Palfa.p = Palfa.p - F*dt # Atualização do Momento
    Palfa.pos = Palfa.pos + Palfa.p/Palfa.mass * dt # atualização da Posição
    Pouro.pos = Pouro.pos + Pouro.p/Pouro.mass * dt
    grafico.plot(pos=(Palfa.p)) # Plotagem dos gráficos
    graficoP.plot(pos=(Pouro.p))
    graficoF.plot(pos=(F)) #plota o gráfico da Força
    if Palfa.pos.x>1.2*Pouro.pos.x:# condição para para início do calculo do angulo
        Vetini = vector(0, Palfa.y, 0)# Ponto inicial para cálculo do angulo
        Dif = (Palfa.pos - Vetini)
        Tg = Dif.y/Dif.x
        hip = sqrt((Dif.y)*(Dif.y) + (Dif.x)*(Dif.x))# Cálculo da Hipotenusa
        Co = Dif.x/hip
```

4 – CONCLUSÕES FINAIS

O projeto continua em andamento e novas modelagens estão sendo feitas para os temas abaixo que , ao final comporão um corpo de problemas a serem propostos nas disciplinas de Introdução à Física:

- 14) Giroscópio.
- 15) Teoria cinética: cilindro com gás e pistão (pressão, temperatura e volume)
- 16) Modelagem de máquinas térmicas: ciclos de Carnot, Otto, Stirling e Diesel.



- 17) Circuitos elétricos.
- 18) Interferência construtiva e destrutiva de ondas eletromagnéticas.
- 19) Polarizadores.
- 20) Dualidade onda partícula: experimentos de dupla fenda com partículas, ondas e elétrons (“gedankhen experiment” do Feynmann)
- 21) Vibrações mecânicas: livres, amortecidas e com força periódica.
- 22) Phonons: vibrações da rede cristalina.
- 23) Difração de raios-X. Planos cristalinos.
- 24) Laser.

A experiência didático-pedagógica aqui proposta parece ser promissora e é possível encontrar relatos da viabilidade de sua implementação nos cursos, se visitarmos os sites da Internet listados na bibliografia. Para transformá-lo realmente em um pesquisa educacional e verificarmos a sua eficácia, junto com a sua prática nos cursos, devemos ter formas de acessar através de dados os resultados obtidos. Esta é a próxima etapa para implementação do projeto.

Agradecimentos

Os autores agradecem o apoio dado pela FAPEMIG, órgão de fomento do estado de Minas Gerais e financiador do projeto através de Bolsa de Iniciação Científica para Thiago Mendes Campos. Os autores também agradecem a conversas e ao apoio dado pelo prof. Ronei Ximenes.

REFERÊNCIAS BIBLIOGRÁFICAS

ASHCROFT, N. AND MERMIN, D. Solid State Physics, Saunders Colledge Publishing

CHABAY, R E SHERWOOD, B : Matter and Interactions, Volumes

FEYNMANN, R.The Feynmann Lectures on Physics, Volumes I, II e III, Addison Wesley

SCHERER, D., DUBOIS, P., & SHERWOOD, B. (2000). VPython: 3D Interactive Scientific Graphics for Students, *Computing in Science and Engineering*, Sept./Oct. 2000, 82-88.

Site da biblioteca do Los Alamos National Laboratory: <http://xxx.lanl.gov>

Site VPython : <http://www.vpython.org>

TIPLER, P. Physics for scientists and engineers, fourth edition, Freeman and Worth I e II, Willey

***Abstract:** This document presents a report about the utilization of modelling and simulation of physical phenomena in three dimensions as a pedagogical tool in teaching Physics for engineering Courses. In particular we use Vpython programming language, for it is easy to learn, henceforth, feasible to be used in the first years where the Physics courses are taught.*



The central point we make is that the student the practice of modelling and programming by the student, enables him to better understand the meaning of physical equations, gets a better idea of the approximations made to build up the model, making this tool worth to compound the analytical skills to be developed by the student. This work is strongly based on the ork of the Center for Innovation in Learning of Carnegie Mellon University, specially the work of Bruce Sherwood and Ruth Chabay.

Key-words: *New technologies, Modeling, Teaching in Engineering*