

# Aplicativo para representação da função dos Harmônicos Esféricos

André Luiz Aparecido RA991397

Projeto de F809 - Instrumentação para Ensino

Orientador: Prof. Silvio Vitiello

## 1 Descrição

O projeto consiste na elaboração de um Aplicativo (Applet) escrito em Java para a representação do quadrado do módulo da função dos Harmônicos Esféricos:  $|Y_m^l(\theta, \phi)|^2$ . Um Applet é um tipo de programa que pode ser incorporado a uma página na Web e proporciona uma maior interação com os usuários de que uma simples página estática.

## 2 Introdução

Os elementos gráficos durante uma aula são de grande ajuda para a compreensão dos alunos. Hoje em dia, a maior parte das universidades possui recursos tecnológicos que permitem uma melhor utilização destes elementos gráficos. Em específico, nas salas do IFGW na Unicamp podemos acessar a Web diretamente para o projetor. Desta forma, os recursos utilizados em sala podem ser adaptados de forma a tirar total proveito desta tecnologia disponível.

A utilização de aplicativos dispostos na Web, permitem que o conteúdo das aulas seja facilmente compartilhado com outras disciplinas ou até mesmos outra universidades.

Buscando obter um bom aproveitamento dos recursos, a utilização de miniaplicativos feitos em Java (applets), que rodam diretamente pela Web e podem ser associados a uma página em Html, é de grande importância. Neste projeto, o objetivo é dar um exemplo deste tipo de aplicação através de um aplicativo que será utilizado para ilustrar um tema de Mecânica

Quântica que também é muito abordado em outras disciplinas ou cursos.

Em Mecânica Quântica, procuramos encontrar uma autofunção que seja comum entre os operadores  $L_z$  e  $L^2$ . Denotamos esta função por  $Y_l^m(\theta, \phi)$  que corresponde aos autovalores  $l(l+1)\hbar^2$  e  $m\hbar$ :

$$L^2 Y_l^m(\theta, \phi) = l(l+1)\hbar^2 Y_l^m(\theta, \phi)$$

$$L_z Y_l^m(\theta, \phi) = m\hbar Y_l^m(\theta, \phi)$$

Esta autofunção é conhecido com função dos “Harmônicos Esféricos” é dada por:

$$Y_l^m(\theta, \phi) = (-1)^l \sqrt{\frac{2m+1}{4\pi} \frac{(m-l)!}{(m+l)!}} P_l^m(\cos\theta) e^{im\phi}$$

Onde:

$$P_l^m(\cos\theta) = \frac{1}{2^l l!} (1 - \cos^2\theta)^{m/2} \frac{d^{m+l}}{d(\cos\theta)^{m+l}} (\cos^2\theta - 1)^l$$

### 3 Relações de Recorrência

Para facilitar o cálculo através de um algoritmo computacional, podemos utilizar de relações de recorrência, que são propriedades da função  $Y_l^m(\theta, \phi)$ .

Podemos escrever:

$$\cos(\theta) Y_l^m(\theta, \phi) = \sqrt{\frac{(l+m+1)(l-m+1)}{(2l+1)(2l+3)}} Y_{l+1}^m(\theta, \phi) + \sqrt{\frac{(l+m)(l-m)}{(2l+1)(2l-1)}} Y_{l-1}^m(\theta, \phi)$$

Através das relações de recorrência podemos escrever o algoritmo para  $|Y_m^l(\theta, \phi)|^2$ . Para tal, primeiramente faz-se uma função dos coeficientes  $coeff(x)$  que será usada para simplificar o código. Assim esta função fica:

$$coeff(x) = \frac{(-1)^x}{2^x x!} \sqrt{\frac{(2x+1)!}{4\pi}}$$

Utilizando  $coeff(x)$  podemos escrever os seguintes valores iniciais:

$$Re[Y_1^m(\theta, \phi)] = coeff(m) \sin(\theta)^m \cos(m\phi)$$

$$Im[Y_1^m(\theta, \phi)] = coeff(m) \sin(\theta)^m \sin(m\phi)$$

$$Re[Y_0^m(\theta, \phi)] = 0$$

$$Im [Y_0^m(\theta, \phi)] = 0$$

Fazemos então a recursão:

$$Re [Y_{i+1}^m(\theta, \phi)] = \sqrt{\frac{(2i+1)(2i+3)}{(i+m+1)(i+m+1)}} \left( \cos(\theta) Re [Y_i^m(\theta, \phi)] - \sqrt{\frac{(i+m)(i-m)}{(2i+1)(2i-1)}} Re [Y_{i-1}^m(\theta, \phi)] \right)$$

$$Im [Y_{i+1}^m(\theta, \phi)] = \sqrt{\frac{(2i+1)(2i+3)}{(i+m+1)(i+m+1)}} \left( \cos(\theta) Im [Y_i^m(\theta, \phi)] - \sqrt{\frac{(i+m)(i-m)}{(2i+1)(2i-1)}} Im [Y_{i-1}^m(\theta, \phi)] \right)$$

Reptimos para  $[i = m, m+1, \dots, l-1]$ . Então podemos calcular  $|Y_m^l(\theta, \phi)|^2$ :

$$|Y_m^l(\theta, \phi)|^2 = Im[Y_m^l(\theta, \phi)]^2 + Re[Y_m^l(\theta, \phi)]^2$$

## 4 Gráficos

Para a interface gráfica do aplicativo estão sendo utilizados métodos de desenho de pontos na tela “DrawOval” pertencente as classe básicas da linguagem Java.

A interface consiste em um gráfico, campos para dar entrada nos valores de  $l, m$  e do ângulo  $\phi$  e de um botão que dispara o evento que calcula os pontos do gráfico e redesenha a interface.

Após as entradas serem digitadas pelo usuário e o botão de cálculo ser acionado, começa-se o algoritmo recursivo que calcula  $|Y_m^l(\theta, \phi)|^2$ . A interface é atualizada ponto-a-ponto e para isto são necessários três métodos auxiliares.

O primeiro método redesenha os eixos e a área de plotagem do gráfico. O segundo procura pelo maior valor de  $|Y_m^l(\theta, \phi)|^2$  e encontra um fator de multiplicação para ajustar o gráfico de forma a ocupar 90% do comprimento total do eixo. Já o terceiro método transforma as cordenadas polares em cartesianas para que os pontos possam ser inseridos no gráfico, e já os desenha utilizando o fator de multiplicação de escala calculado no método anterior.

## 5 Progresso

No momento o projeto se encontra em uma forma simplificada, com os elementos gráficos mínimos e com os métodos descritos na seção anterior. Para finalizar falta uma maior elaboração dos elementos gráficos e adicionar uma condição de interface mais explicativa para o usuário. Serão feitas também melhorias na página que hospeda o projeto, a fim de permitir uma melhor

apresentação do trabalho final.

Uma versão atualizada do projeto será mantida no endereço:

<http://www.geocities.com/aapparec/yml.htm>

Esta página poderá ser visualizada em qualquer navegador que tenha suporte à Java.

## 6 Referências

COHEN-TANNOUJDI, C. & DIU, B. & LAOË, F. "Quantum Mechanics", Vol. 1.

COHEN-TANNOUJDI, C. & DIU, B. & LAOË, F. "Quantum Mechanics", Vol. 2.

ARFKEN, G. "Mathematical Methods for Physicists", Third Edition.

HUNT, J. "Java and Object Orientation - An Introduction"