Programação Orientada a Objetos

Maio de 2016

Quaternions

Quaternions são números com álgebra não-comutativa (isto é, $ab \neq ba$ em geral) que generalizam números complexos. Um quaternion pode ser representado como:

$$q = q_1 + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k},$$

onde q_1, q_2, q_3 e q_4 são números reais, enquanto $\mathbf{i}$, $\mathbf{j}$ e $\mathbf{k}$ são números distintos (imaginários) que satisfazem as relações:

$$\begin{aligned}
\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} &= -1,\\
\mathbf{ij} = -\mathbf{ji} &= \mathbf{k},\\
\mathbf{jk} = -\mathbf{kj} &= \mathbf{i},\\
\mathbf{ki} = -\mathbf{ik} &= \mathbf{j}.
\end{aligned}$$

(As três últimas linhas são redundantes com a primeira.)

O *conjugado* de $q = q_1 + q_2\mathbf{i} + q_3\mathbf{j} + q_4\mathbf{k}$ é definido como

$$q^* = q_1 - q_2\mathbf{i} - q_3\mathbf{j} - q_4\mathbf{k}.$$

A soma de $x = x_1 + x_2\mathbf{i} + x_3\mathbf{j} + x_4\mathbf{k}$ com $y = y_1 + y_2\mathbf{i} + y_3\mathbf{j} + y_4\mathbf{k}$ é

$$x + y = (x_1 + y_1) + (x_2 + y_2)\mathbf{i} + (x_3 + y_3)\mathbf{j} + (x_4 + y_4)\mathbf{k}.$$

O produto é dado por

$$\begin{aligned}
xy = \ & (x_1y_1 - x_2y_2 - x_3y_3 - x_4y_4)\\
& +(x_1y_2 + x_2y_1 + x_3y_4 - x_4y_3)\mathbf{i}\\
& +(x_1y_3 - x_2y_4 + x_3y_1 + x_4y_2)\mathbf{j}\\
& +(x_1y_4 + x_2y_3 - x_3y_2 + x_4y_1)\mathbf{k}.
\end{aligned}$$

(Esta expressão pode ser deduzida usando as relações dos números imaginários acima.)

Note que $qq^* = q^*q = q_1^2 + q_2^2 + q_3^2 + q_4^2$ é sempre real.[1] O inverso de um quaternion é definido como:

$$q^{-1} = \frac{1}{qq^*}q^*$$

[1] Esta expressão, apesar de uma consequência da fórmula de multiplicação, **deve ser usada em calculos em computador sempre que for adequada**: Os valores dos coeficientes de $\mathbf{i}, \mathbf{j}, \mathbf{k}$ se anulam, mas devido a problemas de arredondamento, se qq^* for calculado pela fórmula de multiplicação podem restar coeficientes imaginários não-nulos (o que ocasiona problema, por exemplo, no cálculo da norma).

e seu valor absoluto (ou norma) como:

$$|q| = \sqrt{qq^*}.$$

A partir do inverso podemos definir a divisão:

$$\frac{x}{y} = xy^{-1} = \frac{1}{yy^*}xy^*.$$

Rotações

Quaternions são úteis para representar rotações em espaços tridimensionais. Uma forma de realizar isso é a seguinte:

1. Considere um ponto P com coordenadas cartesianas (x, y, z). Esse ponto pode ser representado pelo quaternion $p = 0 + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ (repare na parte real igual a 0).

2. Uma rotação é especificada através de um vetor $\mathbf{v}$, que indica o eixo da rotação, e por um ângulo α, que indica o ângulo de rotação em torno desse eixo. Representamos uma rotação definida dessa forma através do quaternion:

$$r = \cos\frac{\alpha}{2} + \frac{\sin\frac{\alpha}{2}}{\|\mathbf{v}\|}(v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}),$$

   onde (v_x, v_y, v_z) são as componentes de $\mathbf{v}$ e $\|\mathbf{v}\| = \sqrt{v_x^2 + v_y^2 + v_z^2}$.

3. Para realizar uma rotação no ponto representado pelo quaternion p (gerando o novo ponto p'), basta fazer:[2]

$$p' = rpr^{-1}.$$

   (Lembre-se que o produto não é comutativo!)

Trabalho

1. Escreva uma classe para lidar com quaternions, cuidando para que objetos dessa classe possam trabalhar conjuntamente com valores inteiros, reais e complexos (um complexo $a + b\mathbf{i}$ é equivalente a um quaternion $a + b\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$), quando adequado (i.e. deve ser possível realizar operações aritméticas entre quaternions ou entre inteiros, número de ponto flutuante e complexos e quaternions).

   A implementação deve ser feita em um módulo denominado `quaternions` e a classe deve se chamar `Quaternion`.

2. Escreva um programa que usa esse módulo para realizar o seguinte:

   (a) Cria um cubo com vértices nos pontos $(1, 1, 1)$, $(1, 1, -1)$, $(1, -1, 1)$, $(-1, 1, 1)$, $(1, -1, -1)$, $(-1, 1, -1)$, $(-1, -1, 1)$ e $(-1, -1, -1)$. (O cubo é representado pelos pontos dos seus oito vértices, os pontos são representados por quaternions.)

   (b) Repetidamente pede ao usuário um eixo e um ângulo de rotação. Realize a rotação especificada no cubo (acumuladamente, isto é, sobre a posição do cubo após a última rotação) e mostre os novos valores dos vértices (para cada rotação). Mostrar as coordenadas x, y, z dos pontos.

[2] Você consegue mostrar que p' terá parte real 0, como adequado para a representação de um ponto?

Algumas observações:

- Note que foi pedida uma classe para quaternions sem se referir ao programa de rotação. *Você deve implementar uma classe que se baseie nos conceitos básicos de quaternions expostos, e não apenas o que será necessário para o programa de rotação.*

- Use sobrecarga de operadores.

- No programa de rotação, implemente e use uma classe para representar o cubo, com um método para realizar a rotação especificada por um quaternion dado e métodos para acesso aos vértices. A classe para cubo não precisa ser geral, pode ter apenas o que é necessário ao programa.

- Você pode fazer o programa mais flexível ou geral do que o pedido.

- Entregar um arquivo com o módulo de quaternions e o programa de rotação pedido.