

Propriedades Mecânicas e Geométricas de Objetos Homogêneos Delgados e Poligonais

(Mechanical and Geometric Properties of Thin and Polygonal Homogeneous Objects)

Hendrik T. Macedo¹ e Cláudio A. Macedo²

¹*Laboratoire d'Informatique de Paris 6, Université Pierre et Marie Curie*

8, rue du Capitaine Scott - F 75015, Paris, France

hendrik.macedo@poleia.lip6.fr

²*Departamento de Física, Universidade Federal de Sergipe*

49100-000, São Cristóvão, SE, Brasil

cmacedo@ufs.br

Recebido em 6 de março, 2002. Aceito em 11 de abril, 2002.

Analizamos o desenvolvimento e a utilização de um software computacional para o cálculo de áreas, centros de massa, momentos de inércia e raios de giração de objetos delgados planos. A interação direta do estudante com um processamento visual das grandezas físicas e geométricas é valorizada como instrumento educacional para alunos universitários de física geral. É mostrada visualmente a solução de problemas didáticos e são destacadas as possibilidades de uso do software em outros ramos do conhecimento.

We analyze the development and the usage of a computational software for the calculation of areas, centers of mass, moments of inertia and radii of gyration of flat thin objects. The direct interaction of the student with a visual processing of physical and geometrical quantities is valued as an educational instrument for university students of general physics. They are shown visually the solution of didactic problems and the possibilities of usage of this software in other branches of knowledge is emphasized.

I Introdução

O desenvolvimento de sistemas computacionais como ferramenta de auxílio no ensino torna-se dia a dia uma prática mais comum. Um software educacional bem articulado potencializa o aprendizado e o entendimento de conceitos e grandezas, visto que permite ao estudante realizar alterações em variáveis de entrada para o programa e observar, em resposta, o efeito dessas mudanças numa interface amigável.

Ambientes de programação como Java [1] provêm grande suporte ao desenvolvimento de softwares para simulação de superfícies, por meio de rotinas para tratamento e manipulação de pontos de tela (*pixels*).

Tendo-se o *pixel* como a unidade de área, é possível, por exemplo, desenvolver visualmente o estudo de propriedades físicas que possuem uma representação direta em termos de grandezas geométricas planas.

Neste trabalho, consideramos objetos planos delgados, homogêneos e poligonais, e determinamos geometricamente a área da superfície (equivalente à massa), o centro de massa (equivalente ao centro geométrico),

o momento de inércia e o raio de giração, a partir das coordenadas dos vértices do polígono representativo do objeto.

A motivação do trabalho está em permitir ao estudante universitário de física geral perceber a relação estreita existente em alguns aspectos essenciais, entre física e geometria, melhorar o entendimento dos conceitos de centro de massa, momento de inércia e raio de giração, aprofundar o entendimento do uso de cálculo numérico de grandezas físicas, além de possibilitar a criação de um software computacional de uso acadêmico (área, centro de massa, momento de inércia e raio de giração) e de uso prático importante, tendo em vista a aplicação no cálculo de áreas e centros de terrenos em geral, e regiões geográficas através de mapas. Ou seja, o software pode ter uso didático-científico, econômico e geográfico.

Na seção II, definimos e descrevemos as grandezas físicas envolvidas no cálculo; a metodologia empregada e os algoritmos desenvolvidos são descritos na seção III; na seção IV, são apresentados alguns resultados e aplicações. As conclusões são expressas na seção V.

II Propriedades Mecânicas e Geométricas

Para objetos planos delgados e homogêneos, o simples cálculo de área (A) permite a determinação da massa (M) tendo em vista que a propriedade de homogeneidade corresponde a uma densidade (ρ) constante.

Assim,

$$M = \rho A. \quad (1)$$

As coordenadas do centro de massa [$r_c = (x_c, y_c)$] do corpo são determinadas dividindo-se a área do corpo em elementos de área dA , e sendo a massa de cada um desses elementos $dm = \rho dA$. Nestes termos, o centro de massa é dado por [2]

$$x_c = \frac{\int x dm}{\int dm} = \frac{\int x \rho dA}{\int \rho dA} = \frac{\int x dA}{\int dA} = \frac{\int x dA}{A} \quad (2)$$

e

$$y_c = \frac{\int y dm}{\int dm} = \frac{\int y \rho dA}{\int \rho dA} = \frac{\int y dA}{\int dA} = \frac{\int y dA}{A}. \quad (3)$$

Nesse caso, $\rho = \text{constante}$, o centro de massa é determinado, exclusivamente, pela geometria do corpo.

Na dinâmica de rotação de um corpo rígido, considera-se um eixo de rotação, e denominando-se R como sendo a distância de cada elemento de massa dm do corpo plano e delgado até esse eixo, definimos o momento de inércia por [2]

$$I = \int R^2 dm = \int R^2 \rho dA. \quad (4)$$

Neste estudo, consideramos um eixo Z perpendicular ao corpo passando pelo centro de massa. Dessa forma, o corpo plano e delgado deve ser considerado como estando no plano XY . O eixo assim escolhido é chamado de eixo principal de inércia Z_o .

A partir da Fig. 1, podemos escrever $R^2 = x^2 + y^2$, e em conseqüência, o momento de inércia ao redor do eixo Z_o é

$$I = \int \rho(x^2 + y^2) dA. \quad (5)$$

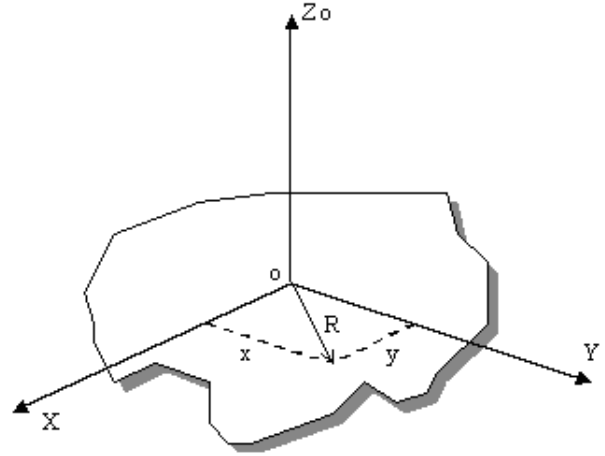


Figura 1. Eixo Z passando pelo centro de massa.

Tendo em vista que o corpo é considerado homogêneo ($\rho = \text{constante}$),

$$I = \rho \int (x^2 + y^2) dA. \quad (6)$$

O cálculo do momento de inércia se reduz a um fator geométrico, igual para todos os corpos de mesma forma e tamanho. Assim sendo, costuma-se definir para corpos homogêneos a grandeza raio de giração do corpo (K), de tal modo que seja válida a relação

$$I = M K^2, \quad (7)$$

ou

$$K = \sqrt{\frac{I}{M}}. \quad (8)$$

K representa a distância ao eixo de giração em que toda massa poderia ser concentrada sem variar o momento de inércia. A vantagem do uso dessa quantidade é que ela pode ser determinada, para corpos homogêneos, inteiramente pela geometria.

No caso de corpos planos delgados e homogêneos, podemos escrever

$$I = \left(\frac{M}{A}\right) \int (x^2 + y^2) dA = M K^2. \quad (9)$$

Assim,

$$K = \sqrt{\frac{\int (x^2 + y^2) dA}{A}}. \quad (10)$$

III Metodologia e Algoritmos

Consideramos a tela do monitor como constituída de 640 pontos (*pixels*) em cada linha horizontal e de 480 pontos (*pixels*) em cada linha vertical, perfazendo um total de 640×480 *pixels*.

Sendo cada ponto considerado de área unitária, uma superfície como a representada na Fig. 2 possuiria $9 \times 7 = 63$ unidades de área.

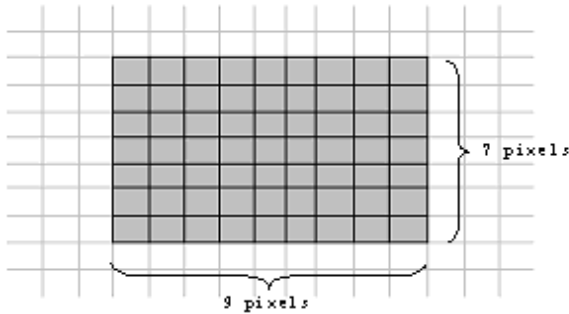


Figura 2. Superfície composta por 63 unidades de área.

O programa desenvolvido requer a entrada de dados sob a forma de pares de coordenadas (x, y) em qualquer sistema de unidades de medida. Em seguida, esses valores reais fornecidos são transformados no sistema de coordenadas da tela e convertidos em quantidades de *pixels*, utilizando-se o canto superior esquerdo do monitor como origem, o eixo X positivo na direção horizontal e o eixo Y positivo na direção vertical.

O pseudocódigo a seguir ilustra como se dá esse processo.

```

Função Translada(float x, float y) {
    int TelaMin = 0; //menor valor possível p/ pixels
    int TelaMax = 640; //maior valor possível p/ pixels
    float FatorEscala = (TelaMax-TelaMin)/(RealMax-RealMin);
    adiciona(r(FatorEscala*(x-RealMin)+TelaMin-0.5)) a TelaCoordX
    adiciona(r(FatorEscala*(y-RealMin)+TelaMin-0.5)) a TelaCoordY
}
onde r = arredondamento, TelaCoord = vetor de pontos de tela

```

Por exemplo, admitindo-se que, dentre todas coordenadas reais fornecidas, o maior e menor valores foram 10.0 e 2.5, respectivamente. Se entrarmos com a coordenada (4.0, 7.5) na função acima, ela será convertida na coordenada de tela (128, 426).

Após ter aplicado a função para todos os vértices reais fornecidos, teremos dois vetores com os valores respectivos em pontos de tela. Em seguida, o programa constrói o polígono correspondente baseado nesses pontos, preenchendo-o com cor pré-determinada:

...

```

fillPolygon(TelaCoordX, TelaCoordY, nVertices),
com nVertices = número de vértices.

```

...

Utilizamos a cor cinza para os *pixels* da área interna do polígono e a cor preta para os *pixels* limites. Dessa forma, a área do objeto (A) é calculada somando-se o número de *pixels* cinzas (P_c) com a metade do número de *pixels* pretos (P_p):

$$A = \sum P_c + \sum \left(\frac{P_p}{2} \right). \quad (11)$$

Esta é uma forma de obtermos um valor mais preciso para a área, visto que a linha que delimita o objeto não deve ter área, e o *pixel* é uma unidade indivisível. A contagem da metade do número de *pixels* pretos simula uma suposta divisão dos *pixels*-limite como ilustrado na Fig. 3 a seguir.

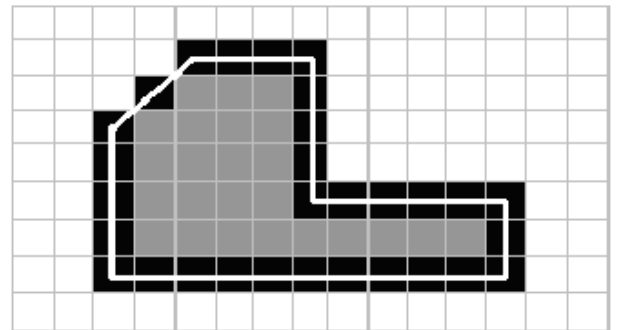


Figura 3. Apenas a metade dos *pixels* pretos é contada.

O centro de massa é obtido convertendo-se as integrais (2) e (3) em somatórios

$$x_c = \frac{\sum x_i \cdot \Delta A_i}{A} \quad (12)$$

e

$$y_c = \frac{\sum y_i \cdot \Delta A_i}{A} \quad (13)$$

onde (x_i, y_i) são coordenadas de cada *pixel* que compõe o objeto na tela e tomamos $\Delta A_i = 1$ para a área do *pixel* de cor cinza e $\Delta A_i = 1/2$ para a área do *pixel* de cor preta.

Por sua vez, o raio de giração é determinado convertendo-se a integral contida em (10) em um somatório

$$K = \sqrt{\frac{\sum (x_i^2 + y_i^2) \cdot \Delta A_i}{A}}, \quad (14)$$

onde x_i , y_i e ΔA_i têm o mesmo significado que o expresso nas equações (12) e (13).

Finalmente, o momento de inércia é determinado, utilizando-se (9) e considerando-se a densidade $\rho = M/A = 1$, com a fórmula

$$I = AK^2. \quad (15)$$

IV Resultados e Aplicações

Como resultado do trabalho, obtemos um software desenvolvido modularmente, com componentes agrupados em classes [3] e, portanto, suscetível a expansões. Sendo desenvolvido na linguagem de programação Java, o software é multiplataforma e possui total interoperabilidade entre sistemas operacionais, podendo ser executado em qualquer ambiente que disponha de uma máquina virtual Java [4], seja este uma estação UNIX, um Macintosh ou um PC.

A versão atual do software dispõe de duas possibilidades para entrada de dados: o usuário pode entrar com os dados, ou seja os pares de coordenadas x , y , manualmente (Fig. 4), ou fazer a leitura a partir de um arquivo contendo esses valores (Fig. 5). Estamos trabalhando para adicionar outras duas possibilidades de entrada de dados. Uma é permitir utilizar o *mouse* para desenhar a figura desejada na tela diretamente, e a outra é obter uma imagem a partir de um scanner diretamente para o software.

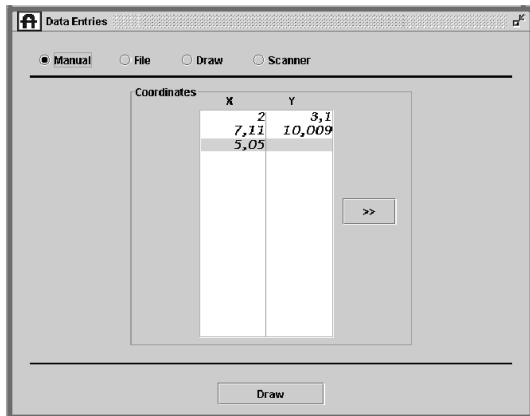


Figura 4. Entrando com coordenadas manualmente em tempo de execução.

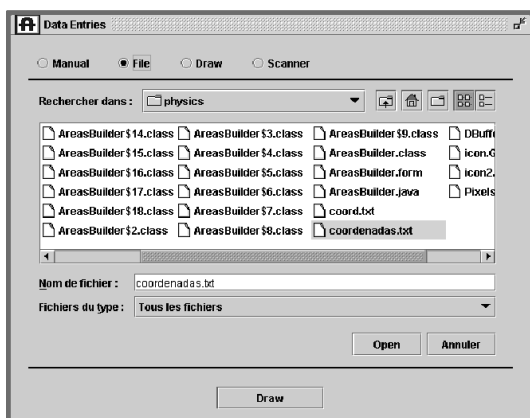


Figura 5. Dados contidos em um arquivo texto simples.

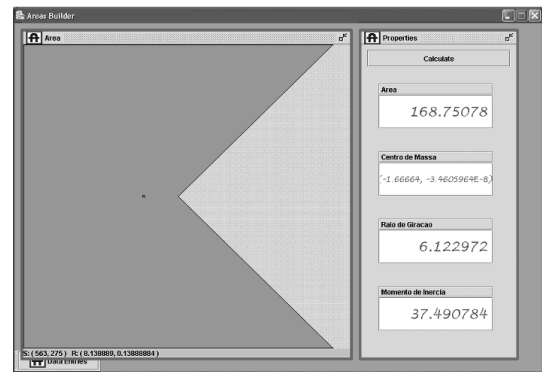


Figura 6. Resultado para o problema 4.51(a) de Alonso & Finn [2].

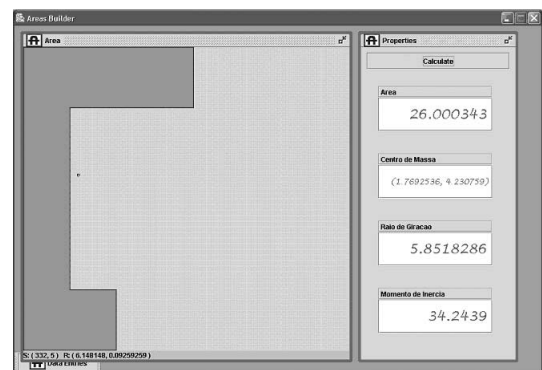


Figura 7. Resultado para o problema 4.53 de Alonso & Finn [2].

A seguir, à guisa de resultados, apresentamos dois exemplos de utilização do software.

Problema 4.51(a) de Alonso & Finn [2]: “Determine as coordenadas do centro de massa do corpo homogêneo representado pelos vértices: $A(0;0)$, $B(7,5;7,5)$, $C(-7,5;7,5)$, $D(-7,5;-7,5)$ e $E(7,5;-7,5)$ ”

Problema 4.53 de Alonso & Finn: “Determine as coordenadas do centro de massa do corpo homogêneo representado pelos vértices: $A(0;10)$, $B(3;10)$, $C(3;8)$, $D(1,5;8)$, $E(1,5;2)$, $F(5,5;2)$, $G(5,5;0)$ e $O(0;0)$ ”.

Note que um pequeno círculo na imagem indica o centro de massa do corpo. Utilizando-se o software [5], pode-se perceber que, à medida que se “passeia” com o *mouse* sobre o objeto, as coordenadas de tela e as coordenadas reais de cada ponto são mostradas no canto inferior esquerdo.

V Conclusões

O desenvolvimento e a utilização pelo estudante universitário de física geral de um software com as características do apresentado neste trabalho contribuem para o entendimento do cálculo numérico de integrais definidas via conversão da integral em somatórios, visualizam os limites computacionais que um objeto pode ter quando tratado como contínuo, permitem a verificação da íntima relação entre algumas grandezas físicas e a geometria com o cálculo do centro de massa e momento de inércia.

Concluimos, indicando o uso didático-científico do software computacional AreasBuilder [5], com a resolução de problemas sobre áreas, centro de massa, raios de giração e momento de inércia de objetos planos homogêneos e delgados, e o uso geográfico-econômico do software, com o cálculo de área e centros geométricos de superfícies e terrenos em geral, inclusive de municípios, estados e países – neste caso utilizando-se coordenadas dos vértices de mapas. Para este fim em especial, a versão 2.0 do software trará a opção de aquisição de figuras a partir de uma fonte externa como um scanner.

Agradecimentos

Agradecemos aos Profs Mário Everaldo de Souza e André M. C. de Souza por úteis discussões, e às agências CNPq, CAPES e FAP-SE pelo apoio financeiro.

Referências

- [1] K. Arnold and G. James. *The Java Programming Language*. Addison-Wesley, 1996.
- [2] N. Alonso and E. J. Finn. *Física: um curso universitário*. V.1. São Paulo: Edgard Blücher 1972.
- [3] P. Wegner. *Concepts and Paradigms of Object-Oriented Programming*. In ACM SIGPLAN OOPS Messenger, 1(1), 1990.
- [4] T. Lindholm and F. Yellin. *The Java Virtual Machine*. Englewood Cliffs: Addison-Wesley, 1996.
- [5] H. T. Macedo e C. A. Macedo. *AreasBuilder*. <http://www.fisica.ufs.br/dfi/CorpoDocente/cam/cmacedo.htm>, 2002.