

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE MATEMÁTICA

**ESTIMANDO NÚMEROS TRANSCENDENTAIS
ATRAVÉS DA PROBABILIDADE GEOMÉTRICA**

WENDEL MAFRA GOMES DOS SANTOS

BELÉM-PA

2015

Certificado de Avaliação

Wendel Mafra Gomes dos Santos

ESTIMANDO NÚMEROS TRANSCENDENTAIS ATRAVÉS DA PROBABILIDADE GEOMÉTRICA

Monografia apresentada na Universidade Federal do Pará como requisito para a obtenção do grau de graduado em Licenciatura Plena em Matemática, para a seguinte banca examinadora:

Prof. Dr. Marcos Monteiro Diniz (orientador)
Faculdade de Matemática, UFPA

Profa. Dra. Cristina Lúcia Dias Vaz
Faculdade de Matemática, UFPA

Prof. Dr. Márcio Lima do Nascimento
Faculdade de Matemática, UFPA

DATA DA AVALIAÇÃO: ____/____/____

CONCEITO: _____

Agradecimentos

Agradeço em primeiro lugar a Deus, o qual enviou ao mundo seu filho Jesus Cristo para mostrar-me como viver de modo significativo e realmente feliz, e me deu sua maravilhosa palavra que ensina a importância do estudo, do conhecimento, da sabedoria, do trabalho duro, do altruísmo, da humildade e disciplina para a realização de tais objetivos.

Agradeço grandissimamente a todos os meus familiares que de forma direta ou indireta contribuíram com minha formação. Especialmente a minha avó materna, a qual exerceu o verdadeiro papel de mãe para mim, e a meu pai, cujos defeitos nunca suplantaram o amor, a dedicação e o empenho no suporte oferecido para a realização dos meus estudos.

Sou profundamente grato a todos os meus amigos, sejam eles de infância, de escola, da fé cristã, das faculdades ou de trabalho, os quais me proporcionaram incontáveis alegrias ao longo da vida, e mostram-se verdadeiros esteios nos momentos mais difíceis. Para não arriscar ser injusto esquecendo de alguém, não citarei nomes. Entretanto, abro uma exceção para Adriane Gomes, a qual nos últimos anos tem sido minha leal companheira e grande ajudadora, e cujo amor nunca falhou comigo.

Deixo um muitíssimo obrigado a todos os meus professores, cujos conselhos e sabedoria constituem-se em fatores determinantes na minha formação, sejam eles da educação básica, de cursos livres, da UPenn ou da UFPA. Dentre todos estes grandes mestres, destaco o Prof. Marcos Monteiro Diniz, que além de muito bem me orientar e aconselhar durante os últimos cinco anos, se mostrou um grande amigo, estendendo seus cuidados para muito além do campo profissional.

A todos aqueles que tive o privilégio de conhecer e tiveram parte nos momentos mais importantes até aqui, muito obrigado!

*"E este é o verdadeiro conhecimento: que Cristo Jesus veio ao mundo para salvar os pecadores,
dos quais eu sou o principal"*

1 Timóteo 1:15

Sumário

Apresentação	1
1 Problemas Iniciais	2
1.1 Introdução à probabilidade Geométrica	2
1.2 O problema da agulha de Buffon	3
1.2.1 Formalizando o Problema	4
1.3 A generalização de Laplace.	6
1.3.1 O problema da agulha de Buffon-Laplace	6
2 Explorando o aplicativo matemático MAXIMA	12
2.1 Alguns comandos introdutórios	13
2.1.1 Definição e manipulação de variáveis	14
2.1.2 Operadores Relacionais	16
2.1.3 Funções Matemáticas	16
2.1.4 Criação de Listas	17
2.2 Programação no MAXIMA	19
2.2.1 Comando <i>if</i>	19
2.2.2 Comando <i>for</i>	20
2.2.3 Comando <i>block</i>	21
3 Implementação computacional dos Problemas Iniciais	23
3.1 Simulação do problema de Buffon	23
3.1.1 Implementação do algoritmo	25
3.2 Simulação do problema da agulha de Buffon-Laplace	27
4 O Problema das Moedas Neperianas	31
4.1 Solucionando o problema	31

4.2	Aproximações para o número transcendental e	34
	Considerações Finais	38
	Referências Bibliográficas	39

Apresentação

A Probabilidade Geométrica é um ramo da teoria das probabilidades cujos problemas fazem uso de considerações geométricas. Teve como marco inicial o *Problema da Agulha de Buffon*, proposto em 1777 pelo cientista francês Georges Louis Leclerc (1707 - 1788), conde de Buffon. O problema diz que se um alfinete for lançado sobre um feixe de retas paralelas, onde a distância entre duas retas consecutivas é igual ao comprimento do alfinete, então a probabilidade de o alfinete cair sobre uma das retas é surpreendentemente $P = 2/\pi$. A surpresa se dá quando lembra-se da definição de probabilidade como sendo os favoráveis sobre os possíveis e observa-se que o número de casos possíveis é proporcional ao número transcendental π .

Quase 50 anos mais tarde, o problema de Buffon foi generalizado a um reticulado com duas coleções perpendiculares entre si de retas paralelas equidistantes. Tal generalização, chamada de *o Problema da Agulha de Buffon-Laplace*, é devida a Pierre-Simon Laplace (1749 - 1827). Estes dois problemas, bem como uma introdução aos conceitos de probabilidade geométrica, serão abordados no primeiro capítulo deste trabalho.

Laplace também usou o problema proposto pelo conde para dar uma estimativa para o valor numérico de π . Esta estimativa e uma aproximação para o valor do número transcendental e estão entre os principais focos do texto aqui apresentado. Para tais fins, utilizaremos a aplicativo matemático *MAXIMA*, cuja apresentação terá o capítulo 2 inteiramente dedicado.

No terceiro capítulo, utilizaremos o *MAXIMA* para criar uma programação para simular os problemas do capítulo 1, e a partir daí, criaremos um algoritmo computacional para estimar o valor numérico de π .

Os outros objetivos aqui perseguidos serão a solução de um problema proposto pelo professor Marcos Monteiro Diniz e a criação de uma rotina para aproximações do valor numérico de e . Estes dois assuntos serão abordados no último capítulo, originando assim o grande momento do presente trabalho.

Capítulo 1

Problemas Iniciais

1.1 Introdução à probabilidade Geométrica

Iniciaremos com uma breve discussão sobre o significado da palavra “probabilidade”. Em matemática, probabilidade é uma medida numérica P da possibilidade de certo acontecimento ocorrer [7]. Esta medida é obtida tomando-se o quociente entre o número de casos favoráveis e o número total de casos. Por exemplo, em um lançamento de um dado, a probabilidade de se obter um número maior que 2 é dada por $P = 4/6 = 2/3$. Portanto, temos $P \in [0, 1]$, sendo $P = 0$ a probabilidade de um evento *impossível* ocorrer, e $P = 1$ a probabilidade da ocorrência de um acontecimento *certo*. De posse destas idéias, considere a figura 1.1:

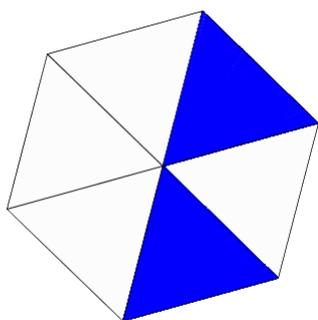


Figura 1.1: Hexágono Regular

Ao escolher-se um ponto do hexágono ao acaso, qual a probabilidade dele estar localizado na região azul? Para responder a esta pergunta, segundo a discussão inicial, precisa-se determinar o número de pontos na região em azul e o número total de pontos do hexágono. Entretanto, estes dois conjuntos de pontos não são enumeráveis.

Vemos assim que as definições iniciais sobre probabilidade não são suficientes no que diz respeito a problemas envolvendo situações geométricas. Do estudo destes tipos de problemas origina-se a *Probabilidade Geométrica*. Esta constatação é apresentada em [3]:

“Em problemas de probabilidade geométrica, os possíveis acontecimentos podem ser representados por pontos de um segmento de reta, por figuras planas ou ainda

por sólidos. Desde que o número de acontecimentos seja usualmente não contável, não podemos definir probabilidade como a razão entre o número de casos favoráveis e o número total de casos. Todavia, podemos ainda definir a probabilidade de um evento de uma maneira natural e calculá-la por meio de considerações geométricas.”

Esta maneira natural acima mencionada para se definir a probabilidade de um evento é aquela que concorda com nossa noção intuitiva. Por exemplo, no problema do hexágono em questão é razoável esperar que a probabilidade de escolha de um ponto na região em azul seja dada pelo número de triângulos em azul sobre o número total de triângulos, ou seja, $P = 2/6$. Em outras palavras, a probabilidade será dada pela razão entre a área da região em azul e área total do hexágono.

Segundo [3], se tivermos uma região A do plano contida em uma região B, admitimos que a probabilidade de um ponto de B também pertencer a A é proporcional à área da região A, $ar(A)$, e não depende da posição que A ocupa em B. Portanto, selecionando ao acaso um ponto de B, a probabilidade P de que ele pertença a A será dada pela razão entre a área de A e a área da região B, $ar(B)$.

$$P = \frac{ar(A)}{ar(B)}$$

■

1.2 O problema da agulha de Buffon

Segundo [1], Georges Louis Leclerc (1707 - 1788), conde de Buffon, foi um cientista francês que se destacou pela autoria de uma notória História natural em vários volumes e uma avaliação de cerca de 75.000 anos para idade da terra, indo de encontro a estimativa de 6.000 anos comumente aceita. Na matemática, suas duas grandes contribuições se deram na tradução para o francês do “*Método dos fluxos de Newton*” e a elaboração do famoso “*Problema da agulha de Buffon*”, proposto no ano de 1777 em seu livro “*Essai d’Arithmétique Morale*”. O problema marcou o surgimento do novo ramo da teoria das probabilidades aqui estudada e foi usado nas primeiras tentativas de aproximar o número transcendental π (Seção 1.3).

Devido sua tamanha importância, um grande número de variações do Problema de Buffon tem sido estudadas. Segundo [2], Duncan (1967) estudou o caso em que um alfinete é lançado sobre um conjunto de retas radiais. Gnedenko (1962) generalizou o problema primeiramente a polígonos convexos de n lados com diâmetros menores que d , e em seguida a curvas convexas fechadas com diâmetros menores que d , considerando tais curvas como limites de polígonos

inscritos, obtendo como probabilidade de uma interseção $P = l/\pi d$. Em [2] é discutido ainda o caso em que o alfinete é lançado sobre um plano coberto por círculos concêntricos. Esta versão do problema também foi proposta de modo independente pelo professor Marcos Monteiro Diniz. Outras variações do Problema da Agulha de Buffon foram discutidas por Ramaley (1969), Perlman e Wichura (1975) e Robertson e Siegel (1986).

Apresentamos na próxima seção a formulação matemática e a solução do tão importante problema proposto pelo conde.

1.2.1 Formalizando o Problema

Considere um alfinete de comprimento l e um plano coberto por um feixe de retas paralelas equidistantes a uma distância d , com $d \geq l$. Ao lançarmos o alfinete aleatoriamente sobre o plano, qual a probabilidade do mesmo intersectar uma das retas?

Solução:

Seja y a distância da extremidade inferior do alfinete à reta mais próxima, à direita (Figura 1.2). Temos $y \in [0; d[$. Seja θ o menor ângulo, no sentido anti-horário, formado por uma reta perpendicular às retas do feixe e a reta que contém o alfinete. Temos $\theta \in]\frac{-\pi}{2}; \frac{\pi}{2}]$.

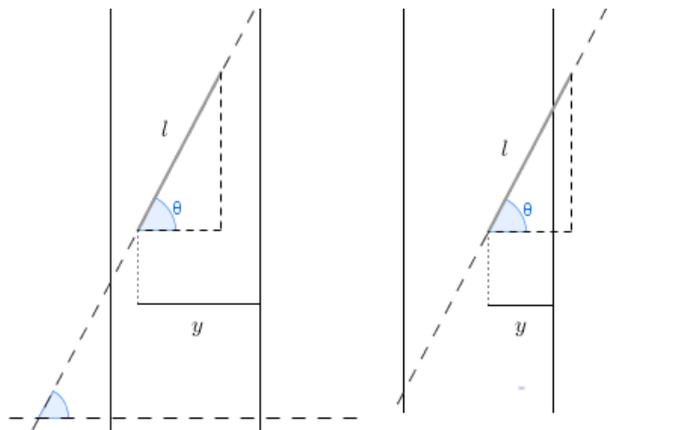


Figura 1.2: Condição de interseção.

Observe que o alfinete cortará uma das retas, se e somente se,

$$y \leq l \cdot \cos \theta$$

Assim, os pares ordenados que satisfazem tal condição, são os pontos acima da reta $y = 0$ e abaixo do gráfico da curva $y = l \cdot \cos \theta$, para $\theta \in] -\pi/2; \pi/2]$. Veja a figura 1.3.

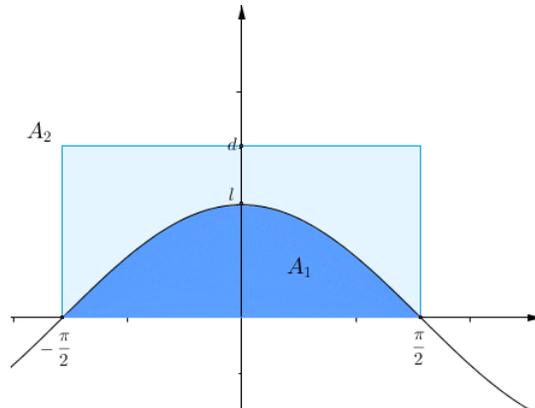


Figura 1.3: Espaço de Configurações.

Logo, a probabilidade P de um alfinete cortar uma reta, é dada por:

$$P = \frac{A_1}{A_2}$$

onde,

A_1 : Área limitada por $y = 0$ e $y = l \cdot \cos \theta$.

A_2 : Área do retângulo (Espaço de configurações).

Como,

$$\begin{aligned} A_1 &= \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} l \cdot \cos \theta d\theta \\ &= l \cdot \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \cos \theta d\theta \\ &= 2l \end{aligned}$$

e

$$A_2 = \pi d$$

temos,

$$P = \frac{2l}{\pi d}.$$

■

1.3 A generalização de Laplace.

Embora o conde tivesse sido o pioneiro nesta área, Pierre-Simon Laplace (1749 - 1827) foi o matemático de maior destaque na teoria das probabilidades. De acordo com [1], é dele a frase “no fundo a teoria das probabilidades é apenas o senso comum expresso em números”. Segundo [7], depois de o problema proposto pelo conde passar por quase meio século de esquecimento, Laplace o utilizou para dar uma estimativa para o valor numérico de π , observando que se o alfinete for lançado n vezes e se s for o número de vezes que ele intersecta uma reta, então espera-se que a razão s/n seja aproximadamente a probabilidade obtida na seção anterior:

$$\frac{s}{n} \approx \frac{2l}{\pi d}$$

Esta última relação sugere como obter aproximações para π , pois ela implica que

$$\pi \approx \frac{2nl}{sd}.$$

Este método pode ser executado de modo empírico, lançando um alfinete sobre um plano coberto por retas paralelas equidistantes e contando o número de lançamentos em que há interseção e o total deles. Há rumores de que um francês realizou o experimento subindo na torre Eiffel e lançando gravetos sobre o solo. Entretanto, para se obter boas aproximações é necessário um grande número de lançamentos. Pode-se levar dias para contar mil lançamentos e verificar se houve ou não interseção. Com auxílio do computador, podemos simular milhões de lançamentos em apenas alguns minutos. Nisto consiste a importância do presente trabalho. Tais simulações serão apresentadas no Capítulo 3.

As contribuições de Laplace não pararam por aí. Ele estendeu o problema original ao sugerir que fosse considerado também um novo feixe de retas paralelas equidistantes ortogonais às retas do primeiro feixe. Este novo desafio será apresentado na seguinte Seção:

1.3.1 O problema da agulha de Buffon-Laplace

Sejam Γ e Λ dois feixes de retas paralelas equidistantes, com distâncias a e b , respectivamente, e α um plano. Considere um alfinete de comprimento l , com $l \leq b \leq a$, e que as retas de Γ sejam perpendiculares às retas de Λ . Ao lançarmos o alfinete sobre o plano α , qual a probabilidade dele intersectar pelo menos uma reta?

Solução:

Seja x distância da extremidade inferior do alfinete à reta mais próxima, à esquerda. Seja y a distância da extremidade inferior do alfinete à reta mais próxima abaixo (Figura 1.4).

Temos $x \in [0; a[$ e $y \in [0; b[$. Seja θ o menor ângulo, tomado no sentido anti-horário, formado por uma reta de Λ (considere que Λ é o feixe de retas horizontais), e a reta que contém o alfinete. Temos $\theta \in] - \pi/2; \pi/2]$.

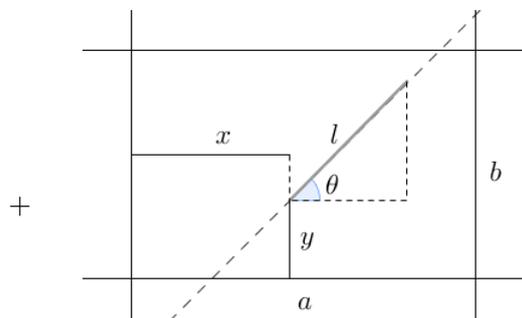


Figura 1.4: Elementos do Problema

Como podemos ver, a posição relativa do alfinete é determinada por três parâmetros (x, y, θ) , e o espaço de configurações é o conjunto R dado por

$$R : \{(x, y, \theta) \in \mathbb{R}^3 \mid 0 \leq x \leq a, 0 \leq y \leq b \text{ e } -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}\}.$$

Considerar os três seguintes eventos:

A : O alfinete corta uma reta de Γ ;

B : O alfinete corta uma reta de Λ ;

C : O alfinete corta simultaneamente uma reta de Γ e uma reta de Λ .

Assim, a probabilidade P do alfinete cortar uma das retas é dada por

$$P = P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Segue da seção anterior que: $P(A) = \frac{2l}{a\pi}$ e $P(B) = \frac{2l}{b\pi}$.

Para calcular $P(A \cap B)$, notemos que o evento C ocorre se, e somente se,

$$a \leq x + l \cdot \cos \theta$$

$$b \leq y + l \cdot |\text{sen} \theta|$$

ou, equivalentemente,

$$x \geq a - l \cdot \cos \theta \tag{1.1}$$

$$y \geq b - l \cdot |\text{sen} \theta| \tag{1.2}$$

Observe que (1.1) é a condição para que o evento A ocorra, e (1.2) é a condição para que B ocorra. Assim, as ternas (x, y, θ) que satisfazem as condições (1.1) e (1.2) simultaneamente são

os pontos do conjunto R que são internos a interseção das superfícies cilíndricas

$$x = a - l \cdot \cos \theta \quad (1.3)$$

$$y = b - l \cdot |\operatorname{sen} \theta| \quad (1.4)$$

A Figura 1.5 representa a superfície cilíndrica (1.3) onde $l = 1$ e $a = 2$.

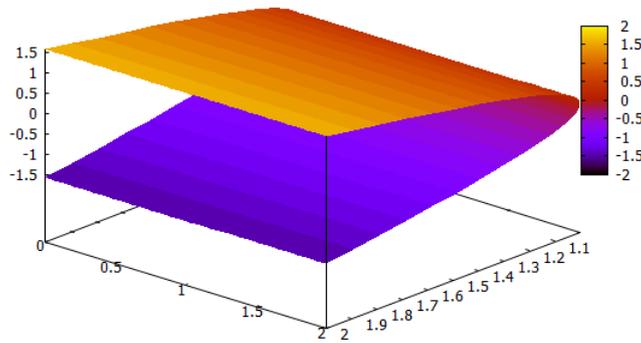


Figura 1.5: $x = 2 - \cos \theta$

Observe que os pontos pertencentes ao conjunto R que satisfazem a condição (1.1) são os pontos internos ao sólido convexo delimitado pela superfície (1.3). Seja S_1 tal sólido. Assim, uma outra forma de calcular $P(A)$ seria através da razão entre o volume de S_1 , denotado por $v(S_1)$, e volume $v(R)$ do paralelepípedo formado pelos pontos do conjunto R . Temos,

$$\begin{aligned} v(S_1) &= ab\pi - \int_0^b \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} a - l \cdot \cos \theta \, d\theta \, dy \\ &= ab\pi - \int_0^b dy \int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} a - l \cdot \cos \theta \, d\theta \\ &= ab\pi - b \left(a\theta - l \cdot \operatorname{sen} \theta \right) \Big|_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \\ &= ab\pi - b(a\pi - l(1 - (-1))) \\ &= ab\pi - (ab\pi - 2bl) \\ &= 2bl \end{aligned}$$

e assim,

$$P(A) = \frac{v(S_1)}{v(R)} = \frac{2bl}{ab\pi} = \frac{2l}{a\pi}.$$

■

Analogamente, tem-se na figura 1.6 uma representação da superfície descrita por (1.4), onde $l = 1$ e $b = 3$.

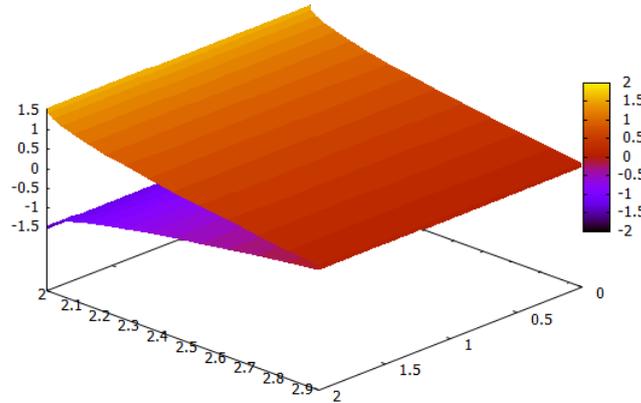


Figura 1.6: $y = 3 - |\text{sen}\theta|$

Os pontos pertencentes ao conjunto R que satisfazem a condição (1.2) são aqueles localizados “à direita” da superfície (1.4), considerando que o “lado direito” é o semi-eixo positivo do eixo y . Se S_2 é sólido formado por tais pontos, então a probabilidade $P(B)$ do evento B ocorrer é dada por:

$$P(B) = \frac{v(S_2)}{v(R)} = \frac{2la}{\pi ab} = \frac{2l}{\pi b}.$$

■

Seja S_3 o sólido formado pela interseção dos sólidos S_1 e S_2 .

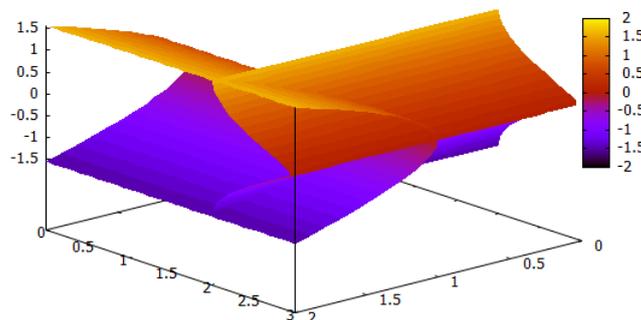


Figura 1.7: Interseção dos Sólidos

Assim, a probabilidade de ocorrência do evento C é dada pela razão entre o volume do sólido S_3 e o volume do paralelepípedo R . Para determinar o volume $v(S_3)$, vamos calcular a integral

dupla da função $f(x, \theta) = b - l \cdot |\text{sen}\theta|$ na região B do plano $x\theta$ dada por

$$B = \{(x, \theta) \in \mathbb{R}^2 \mid a - l \cdot \cos(\theta) \leq x \leq a \text{ e } \frac{-\pi}{2} \leq \theta \leq \frac{\pi}{2}\}.$$

Então, o volume do sólido S_3 será a diferença entre o volume do sólido S_1 e o volume abaixo do gráfico de f na região B . Temos

$$\begin{aligned} v(S_3) &= 2bl - \iint_B b - l \cdot |\text{sen}\theta| \, dx d\theta \\ &= 2bl - \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \int_{a-l \cdot \cos\theta}^a b - l \cdot |\text{sen}\theta| \, dx d\theta \\ &= 2bl - \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} b - l \cdot |\text{sen}\theta| \left(\int_{a-l \cdot \cos\theta}^a dx \right) d\theta \\ &= 2bl - \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} b - l \cdot |\text{sen}\theta| [a - (a - l \cdot \cos\theta)] d\theta \\ &= 2bl - \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} bl \cdot \cos\theta - l^2 \cdot |\text{sen}\theta| \cdot \cos\theta d\theta \\ &= 2bl - \left(bl \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \cos\theta d\theta - l^2 \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} |\text{sen}\theta| \cdot \cos\theta d\theta \right) \\ &= 2bl - bl[\text{sen}(\frac{\pi}{2}) - \text{sen}(\frac{-\pi}{2})] + l^2 \int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} |\text{sen}\theta| \cdot \cos\theta d\theta \\ &= l^2 \left(\int_{\frac{-\pi}{2}}^0 (-\text{sen}\theta) \cdot \cos\theta d\theta + \int_0^{\frac{\pi}{2}} \text{sen}\theta \cdot \cos\theta d\theta \right) \\ &= \frac{l^2}{2} \left(\int_0^{\frac{-\pi}{2}} \text{sen}2\theta d\theta + \int_0^{\frac{\pi}{2}} \text{sen}2\theta d\theta \right) \\ &= \frac{l^2}{2} \left(\frac{-\cos(2\frac{-\pi}{2})}{2} - \frac{-\cos(0)}{2} + \frac{-\cos(2\frac{\pi}{2})}{2} - \frac{-\cos(0)}{2} \right) \\ &= \frac{l^2}{2} \left(\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} \right) \\ &= l^2 \end{aligned}$$

■

É interessante notar que o volume do sólido S_3 não depende dos espaçamentos a e b entre as retas mas somente do comprimento l do alfinete.

Logo, a probabilidade de alfinete cair cortando duas retas simultâneamente é dada por

$$P(C) = P(A \cap B) = \frac{v(S_3)}{v(R)} = \frac{l^2}{\pi ab}$$

■

Portanto, a resposta ao questionamento elaborado por Laplace é dada por

$$\begin{aligned} P &= P(A) + P(B) - P(A \cap B) \\ &= \frac{2l}{\pi a} + \frac{2l}{\pi b} - \frac{l^2}{\pi ab} \\ &= \frac{l(2(a+b) - l)}{\pi ab} \end{aligned}$$

■

Capítulo 2

Explorando o aplicativo matemático MAXIMA

O objetivo deste capítulo é apresentar o software *MAXIMA*, bem como os principais comandos deste aplicativo a serem utilizados neste trabalho. Tais informações se fazem necessárias para um maior entendimento das idéias aqui apresentadas para criação dos algoritmos que determinam aproximações para os valores numéricos de π e e , que serão abordados nos próximos capítulos. Segundo [6],

“O MAXIMA é uma linguagem computacional que permite realizar cálculos numéricos e simbólicos, representações gráficas e efetuar programação, possuindo uma grande variedade de comandos para os mais variados fins em Matemática e aplicações. É um software de livre acesso, disponível para os sistemas operativos usuais.”

Você pode baixar este aplicativo e obter informações de como instalá-lo no seguinte website:

<http://MAXIMA.sourceforge.net/>

Durante a instalação, pode-se optar pelas interfaces:

- *WxMaxima*
- *xMAXIMA*

Por maior familiaridade, utilizaremos a primeira delas neste trabalho. Se optar pela mesma, após a instalação, esta será a tela inicial do *MAXIMA*:

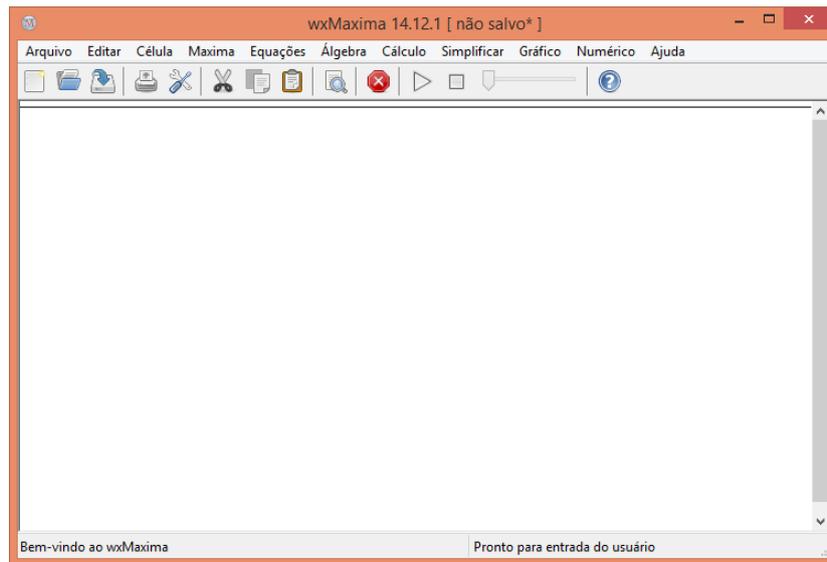


Figura 2.1: Interface WxMaxima

2.1 Alguns comandos introdutórios

Por possuir tantas funcionalidades matemáticas, é indubitável que o *MAXIMA* é uma “super calculadora”. Inicialmente, vamos esquecer da palavra “super” e nos concentrar em entender como usar este aplicativo como sendo apenas uma simples calculadora. Vejamos alguns exemplos de operações aritméticas básicas:

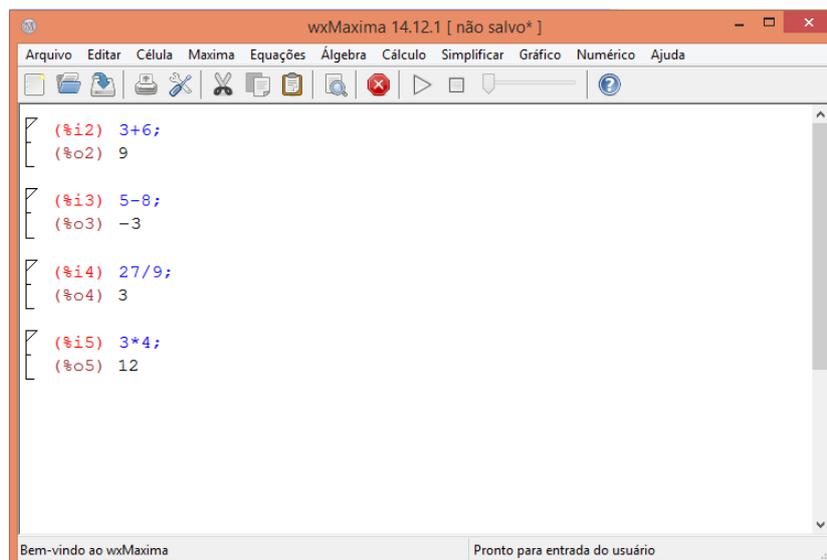


Figura 2.2: Operações aritméticas

Como podemos ver, os comandos de entrada (*inputs*) são indicados pelo símbolo $(\%i_j)$ e

representados em *azul*, e as respostas fornecidas pelo *MAXIMA* (*outputs*) são representadas em *preto* e estão acompanhadas do símbolo `(%ok)`, onde *j* e *k* são números naturais indicando a ordem em que tais comandos foram criados.

Para realizar as operações acima, deve-se escrever a expressão e pressionar simultaneamente as teclas “*shift*” e “*enter*”. Um sinal de “*;*” é gerado automaticamente após a execução do comando.

```
(%i1) 9+3-5;
```

```
(%o1) 7
```

O quadro abaixo mostra os comandos para realização de algumas operações matemáticas.

Entradas	Operações
$a + b$	soma de a com b
$a - b$	subtração de a com b
$a*b$	produto de a por b
a/b	divisão de a por b
a^b	o número a elevado ao número b
$a^{\{1/n\}}$	a raiz n -ésima de a

Tabela 2.1: Operações Básicas

2.1.1 Definição e manipulação de variáveis

Podemos definir variáveis utilizando o sinal “*:*” de um modo bem simples. Veja:

```
(%i2) a:2;
```

```
(%o2) 2
```

```
(%i3) c:7;
```

```
(%o3) 7
```

Tendo defido-as, podemos visualizar seu valor através do comando “*print*”, bem como manipulá-las e operar com a mesmas.

```
(%i4) print(c);
```

```
(%o4) 7
```

```
(%i5) a+3;
```

```
(%o5) 5
```

```
(%i6) a+c/3;
```

```
(%o6)  $\frac{13}{3}$ 
```

```
(%i7) c^a;
```

```
(%o7) 49
```

Também podemos operar com algumas constantes. Iniciaremos com aquelas que serão de particular interesse, os números π e o e . No *MAXIMA*, o identificador destes números é o símbolo “%”, juntamente com seu respectivo nome.

```
(%i8) %pi;
```

```
(%o8)  $\pi$ 
```

```
(%i9) %e;
```

```
(%o9)  $e$ 
```

```
(%i10) %alpha;
```

```
(%o10)  $\alpha$ 
```

```
(%i11) %beta;
```

```
(%o11)  $\beta$ 
```

Observe que output para letras gregas é simbólico, enquanto que para o e é o próprio input. O comando “float” nos garante que estes são os velhos conhecidos pois determina as 16 primeiras casas decimais destes números.

```
(%i12) float(%e);
```

```
(%o12) 2.718281828459045
```

```
(%i13) float(%pi);
```

```
(%o13) 3.141592653589793
```

Se quisermos melhorar a aproximação para um número “ n ” de casas decimais, usamos o comando “fpprec : n ”, que ajusta a precisão para n . O comando “bfloat” exibe a nova aproximação.

```
(%i14) fpprec : 50;
```

```
(%o14) 50
```

```
(%i15) bfloat(%pi);
```

```
(%o15) 3.1415926535897932384626433832795028841971693993751b0
```

2.1.2 Operadores Relacionais

O *MAXIMA* possui os operadores relacionais usuais para comparação de números, variáveis, expressões, etc. Alguns deles são mostrados no quadro abaixo:

Operador	Relação
>	Maior
>=	Maior ou igual
<	Menor
<=	Menor ou igual
=	Igualdade
#	Diferença

Tabela 2.2: Operadores Relacionais

2.1.3 Funções Matemáticas

O quadro abaixo mostra como o *MAXIMA* interpreta algumas das funções matemáticas mais usuais.

Operador	Relação
sqrt(x)	Raiz quadrada de x
abs(expressão)	Valor absoluto de <i>expressão</i>
n!	Fatorial do número n
log(x)	Logarítmo neperiano de x
exp(x)	Exponencial de x
sin(x)	Seno do número x
cos(x)	Cosseno de x
tan(x)	Tangente de x

Tabela 2.3: Funções

Além destas, é possível definir outras funções combinando as anteriores com operações:

```
(%i2) f(x) := 2*sqrt(x)-5*sin(x)/3 + x;
```

```
(%o2) f(x) := 2√x -  $\frac{5 \sin(x)}{3}$  + x
```

```
(%i3) f(3);
```

```
(%o3) -  $\frac{5 \sin(3)}{3}$  + 2√3 + 3
```

O comando “float” avaliado em “%” retorna uma aproximação numérica para a última expressão.

```
(%i4) float(%);
```

```
(%o4) 6.228901601704642
```

Finalizamos esta seção com a apresentação da função “random(x)” que será de grande utilidade neste trabalho. Quando avaliada em um número natural, ela retorna um número natural pseudoaleatório menor que x . Se x é um número racional positivo escrito em forma decimal, esta função avaliada em x retorna um número de mesma natureza menor do que x .

```
(%i5) random(6);
```

```
(%o5) 5
```

```
(%i6) random(1);
```

```
(%o6) 0
```

```
(%i7) random(1.0);
```

```
(%o7) 0.6125738892538246
```

Vale ressaltar que as funções trigonométricas são sempre avaliadas em radianos. Além disso, o *MAXIMA* não possui comandos para calcular logaritmos em outras bases. Para tal fim utiliza-se a propriedade de mudança de base dos logaritmos. Apesar disto, o *MAXIMA* apresenta ainda comandos para várias outras funções, como as trigonométricas inversas e hiperbólicas. Para mais informações, consulte [9] ou [6].

2.1.4 Criação de Listas

A representação de vários tipos de dados agrupados é a principal utilidade das listas. Para defini-las, basta dar um nome qualquer, colocar o símbolo “:” indicador da definição e os seus elementos entre colchetes separados por vírgulas. Os elementos podem ser qualquer símbolos, inclusive outras listas.

```
(%i16) nomequalquer: [X,Y,3,&,b,f,5,8,90,2,23];
```

```
(%o16) [X,Y,3,&,b,f,5,8,90,2,23]
```

Para verificar se a expressão é uma lista ou não, utiliza-se o comando “`listp`”:

```
(%i17) listp(nomequalquer);
```

```
(%o17) true
```

Em caso afirmativo, o MAXIMA retorna *true*. Caso contrário, o MAXIMA retorna *false*.

```
(%i18) listafalsa: 3;
```

```
(%o18) 3
```

```
(%i19) listp(listafalsa);
```

```
(%o19) false
```

Para se obter o n -ésimo elemento de uma lista, digita-se o nome da lista juntamente com o número n entre colchetes.

```
(%i20) nomequalquer[1];
```

```
(%o20) X
```

```
(%i21) nomequalquer[4];
```

```
(%o21) &
```

O comando “`rest(nomedalista,n)`” gera uma lista com os n primeiros elementos removidos.

```
(%i22) rest(nomequalquer,4);
```

```
(%o22) [b,f,5,8,90,2,23]
```

Para saber quantos elementos possui uma lista, utiliza-se:

```
(%i23) length(nomequalquer);
```

```
(%o23) 11
```

Para ordenar os elementos de uma lista, usamos o comando:

```
(%i24) sort(nomequalquer);
```

```
(%o24) [2,3,5,8,23,90,&,b,f,X,Y]
```

Em muitas situações, é necessário criar listas que obedecem a certos critérios e padrões. Para tais fins, utilizamos os comandos “`create_list`” ou “`makelist`”, que produzem os mesmos resultados. A criação deste tipo de lista se dá do seguinte modo: Define-se a expressão que

originará cada elemento da lista, bem como as variáveis que tal expressão irá depender seguidas de seus intervalos de variação. Vejamos alguns exemplos:

```
(%i25) create_list(n!,n,1,6);
```

```
(%o25) [1, 2, 6, 24, 120, 720]
```

```
(%i26) create_list(n!,n,[3,5,6]);
```

```
(%o26) [6, 120, 720]
```

```
(%i27) makelist(n!,n,1,6);
```

```
(%o27) [1, 2, 6, 24, 120, 720]
```

```
(%i28) makelist(2*x+y=j,j,1,5);
```

```
(%o28) [y + 2x = 1, y + 2x = 2, y + 2x = 3, y + 2x = 4, y + 2x = 5]
```

```
(%i29) create_list(g[i,j],i,7,10,j,1,3);
```

```
(%o29) [g7,1, g7,2, g7,3, g8,1, g8,2, g8,3, g9,1, g9,2, g9,3, g10,1, g10,2, g10,3]
```

```
(%i30) create_list(random(2.0),i,1,4);
```

```
(%o30) [1.159515628635807, 1.088061463973348, 0.7280549419224522, 1.213079475216045]
```

Existem ainda vários outros comandos para criação, operação e manipulação de listas. Apesar de uma profunda descrição dos mesmos não ser o principal foco deste trabalho, maiores informações podem ser encontradas em [9] e [6].

2.2 Programação no MAXIMA

Esta seção é dedicada a descrição de três comandos básicos de programação que mostrarão como implementar algoritmos no MAXIMA.

2.2.1 Comando *if*

Como o próprio nome diz, o comando `if` (que significa “se” em português) é usado mediante a situações condicionais. Sua descrição no MAXIMA é:

```
--> if "condição" then "instrução"
```

Em outras palavras, é como se estivéssemos dizendo ao MAXIMA: “Se isto ocorrer então (que significa ‘then’ em inglês) faça aquilo”. Vejmos um exemplo:

```
(%i1) a:4;
(%o1) 4

(%i2) if a>3 then print(Verdadeiro);
(%o2) Verdadeiro
```

O comando `print`, usado inicialmente para mostrar o valor numérico de variáveis definidas, pode ser usado de maneira mais geral, mostrando o valor do argumento em que é avaliado. Em outras palavras, o comando `print` diz ao MAXIMA o que mostrar no output.

Mas o que fazer se a condição imposta não se verifica? Para tal finalidade, temos o comando complementar `else` (que quer dizer “caso contrário”, em português).

```
(%i3) a:3;
(%o3) 3

(%i4) if a>3 then print("verdadeiro") else a^2;
(%o4) 9
```

Neste caso, foi dada ao MAXIMA a seguinte instrução: “Se ‘a’ for maior que 3, imprima a palavra verdadeiro. Caso contrário, eleve ‘a’ ao quadrado”.

2.2.2 Comando *for*

O comando `for` é usado quando deseja-se avaliar uma instrução, que depende de uma variável, certo número de vezes. Esta variável admite somente valores inteiros, sendo necessário especificar os valores inicial e final, bem como os valores em que a instrução irá ser avaliada dentro do intervalo especificado. Estes valores são estabelecidos pela opção `step` (que significa degrau), funcionando da seguinte maneira: se `step:1`, então a avaliação dentro do intervalo especificado é feita de 1 em 1; se `step:2`, a avaliação da instrução é feita de 2 em 2, e assim sucessivamente. O comando tem a seguinte estrutura:

```
--> for "Variavel:Valor_inicial" step "padrão_da_avaliação"
      thru "valor_final" do "instrução"
```

É como se estivesse dito ao MAXIMA: “Para esta variável, que vai de ‘valor inicial’ ao ‘valor final’, variando de ‘step’ em ‘step’, faça ‘instrução’”. Vejamos um exemplo:

```
(%i5) for i:2 step 3 thru 12 do print(log(i));
log(2)
```

```
log (5)
```

```
log (8)
```

```
log (11)
```

```
(%o5) done
```

Para obter os valores numéricos, adicione o comando `float`.

```
(%i6) for i:2 step 3 thru 12 do print(float(log(i)));
```

```
0.6931471805599453
```

```
1.6094379124341
```

```
2.079441541679836
```

```
2.397895272798371
```

```
(%o6) done
```

Outra maneira de utilização deste comando é através da opção `while`, ao invés da opção `thru`. Em vez de especificar o valor final, com a opção `while` pode-se determinar uma condição que determinará o valor final para a variável.

```
(%i7) for x:1 step 2 while x<10 do print(create_list(x^2+i,i,1,4));
```

```
[2,3,4,5]
```

```
[10,11,12,13]
```

```
[26,27,28,29]
```

```
[50,51,52,53]
```

```
[82,83,84,85]
```

```
(%o7) done
```

2.2.3 Comando *block*

Apesar do emprego e grande utilidade dos comandos `if` e `for` na criação de rotinas computacionais, o comando `block` é o principal responsável por esta tarefa. Sua função é exatamente esta. Ele cria procedimentos bem elaborados que englobam sequências de instruções. A estrutura deste comando é:

```
--> Nome_do_algoritmo(argumento1,argumento2,...,argumentok):=block([
    variável_local1,variável_local2,...,variável_localK],
    instrução1,
    ...
    instruçãoN)
```

Por exemplo, vamos criar um algoritmo para determinar o número de diagonais e a soma dos ângulos internos de um polígono de n lados:

```
(%i8) Poligono(n):=block([diagonais,soma_angulos],

    diagonais:n*(n-3)/2,

    soma_angulos:(n-2)*180,

    return([diagonais,soma_angulos]));
```

```
(%o8) Poligono(n) := block([diagonais,soma_angulos], diagonais :  $\frac{n(n-3)}{2}$ , soma_angulos :
(n-2) 180, return([diagonais,soma_angulos]))
```

O algoritmo foi criado. Agora, para saber o número de diagonais e a soma dos ângulos internos de um polígono de n lados, basta avaliar a função `Poligono` em n .

```
(%i9) Poligono(6);
```

```
(%o9) [9, 720]
```

```
(%i10) Poligono(60);
```

```
(%o10) [1710, 10440]
```

Capítulo 3

Implementação computacional dos Problemas Iniciais

Como foi visto na Seção 1.2, no problema da agulha de Buffon considera-se o lançamento de um alfinete sobre um plano coberto por retas paralelas equidistantes. Tem-se que a probabilidade do alfinete cair intersectando uma das retas é $2l/\pi d$, onde l é o comprimento do alfinete, d é a distância entre as retas e $l \leq d$. Assim, espera-se que em n lançamentos do alfinetes, o número s de vezes em que ele intersecta uma das retas é tal que s/n é aproximadamente $2l/\pi d$, ou seja,

$$\frac{s}{n} \approx \frac{2l}{\pi d}$$

Esta última relação nos dá uma sugestão de como obter aproximações para π , pois ela implica que

$$\pi \approx \frac{2nl}{sd}$$

Segundo [7], esta idéia foi concebida por Laplace quase meio século depois da criação do problema de Buffon. Vejamos então como simular estes n lançamentos usando o MAXIMA.

3.1 Simulação do problema de Buffon

Da seção 1.2, segue que a posição relativa do alfinete à uma reta é determinada por dois parâmetros (y, θ) , onde $y \in [0; d[$ é a distância da extremidade inferior do alfinete à reta mais próxima à direita, e $\theta \in [-\frac{\pi}{2}; \frac{\pi}{2}[$ é o menor ângulo, no sentido anti-horário, formado por uma reta perpendicular às retas do feixe e a reta que contém o alfinete. Foi visto ainda que a condição

para que um alfinete intersecte uma das retas é dada por

$$l * \cos\theta \geq y$$

Assim, criaremos uma lista contendo cada uma das posições do alfinete em n lançamentos, e contaremos quantos são os lançamentos em que tal condição se verifica. Iniciamos por atribuir alguns valores para l , d e n , com possibilidade de alteração dos mesmos sempre que necessário.

```
(%i1) l:1;
(%o1) 1
(%i2) n:1000;
(%o2) 1000
(%i3) d:1.0;
(%o3) 1.0
```

Para criação da lista utiliza-se o comando `random(x)`. Aplicado a d , este comando nos dá um número aleatório no intervalo $[0; d[$. Portanto, esta função é ideal para determinação de y nos lançamentos. Infelizmente, não temos um comando que nos forneça um número aleatório no intervalo $[-\frac{\pi}{2}; \frac{\pi}{2}[$. Entretanto, podemos ainda usar o função `random(x)`, aplicado a $x = 1.0$, e pensar em uma função que leve o intervalo $[0; 1[$ no intervalo desejado. Para que não haja perda da uniformidade da variável θ , vamos usar uma função linear. A escolha feita é $f(x) = (2x - 1)\frac{\pi}{2}$. Assim, a partir da composição de funções, pode-se criar a lista com as posições de queda do alfinete em cada lançamento aleatório.

```
(%i4) lancamentos : create_list([random(d), (2*random(1.0)-1)*%pi/2], i, 1, n);
<<Expressão longa demais para ser exibida!>>
```

A lista foi criada. Temos os n lançamentos realizados. É necessário contar em quantos destes houve interseção. É preciso dizer ao MAXIMA: “Para cada elemento da lista ‘lancamentos’, verifique se este elemento satisfaz a condição de interseção e conte o número destes elementos!”. Utiliza-se os comandos de programação para esta tarefa. O comando `if` é utilizado na segunda parte da instrução ao MAXIMA, e o comando `for` é empregado na primeira.

```
(%i5) s:0;
(%o5) 0
```

```
(%i6) for i: 1 thru n do
      if l*cos(lancamentos[i][2]) >= lancamentos[i][1] then s: s + 1;
(%o6) done

(%i7) print(s);
(%o7) 630
```

A contagem foi realizada. Podemos comparar os resultados e verificar as relações apresentados no início do capítulo:

```
(%i8) float(s/n);
(%o8) 0.63

(%i9) float(2*l/(%pi*d));
(%o9) 0.6366197723675814
```

e finalmente, realizar a tão esperada aproximação para o número π :

```
(%i10) float(2*n*l/(s*d));
(%o10) 3.174603174603174
```

3.1.1 Implementação do algoritmo

Como foi visto na Seção 2.2.3, o comando `block` é utilizado na criação de comandos que englobam sequências de instruções. O objetivo desta seção é utilizar este comando para definir uma função que através de uma única instrução realize todas as tarefas desenvolvidas na seção anterior e nos forneça uma aproximação para π . Inicia-se por observar que, na última seção, a criação da lista é um passo que poderia ter sido ignorado, tendo em vista que a criação dos pares ordenados poderia ter sido realizada “ao mesmo tempo” em que os pares ordenados são avaliados. Em outras palavras, ao invés de criar todos os pares e depois avaliá-los, o MAXIMA poderia ter criado e avaliado cada um no “momento” da criação. Veja:

```
(%i11) s:0;
(%o11) 0

(%i12) for i: 1 thru n do
      if l*cos((2*random(1.0)-1)*%pi/2) >= random(d) then s: s + 1;
(%o12) done
```

```
(%i13) print(s);
```

```
(%o13) 631
```

Esta idéia será utilizada aqui. Em homenagem ao conde criador do problema, o comando terá seu nome. Este comando dependerá apenas das variáveis l , n , e d que foram definidas no começo da seção anterior, e nos retornará a probabilidade exata calculada na Seção 1.2, a probabilidade aproximada obtida e a aproximação para π . Além de englobar todas as instruções necessárias para tais resultados no comando `block`, emprega-se uma nova instrução que nos dará maior clareza ao interpretar o output. Pode-se nomear os resultados exibidos se o nome desejado for escrito entre aspas. Por exemplo,

```
(%i14) "Número de alfinetes que intersectam uma reta"=s;
```

```
(%o14) Número de alfinetes que intersectam uma reta= 631
```

De posse de todos os elementos necessários, tem-se a realização do objetivo da seção:

```
(%i15) Buffon(l,n,d):= block([Prob,AproxPob,AproxPi],
    s:0,
    for i: 1 thru n do
        (if l*cos((2*random(1.0)-1)*%pi/2) >= random(d) then s: s + 1),
        Prob:float(2*l/(%pi*d)),
        AproxProb:float(s/n),
        AproxPi: float(2*n*l/(s*d)),
        "[Probabilidade,Probab. Aproximada,
        Aproximação para PI]" = [Prob,AproxProb,AproxPi]
    );
```

```
(%o15) Buffon(l,n,d) := block([Prob, AproxPob, AproxPi], s : 0, for i thru n do
if l cos  $\left(\frac{(2 \text{random}(1.0)-1)\pi}{2}\right) \geq \text{random}(d)$  then s : s + 1, Prob : float  $\left(\frac{2l}{\pi d}\right)$ ,
AproxProb : float  $\left(\frac{s}{n}\right)$ , AproxPi : float  $\left(\frac{2nl}{sd}\right)$ , [Probabilidade, Probab. Aproximada,
Aproximação para Pi] =[Prob, AproxProb, AproxPi])
```

```
(%i16) Buffon(1,100000,2.0);
```

```
(%o16) [Probabilidade, Probab. Aproximada, Aproximação para PI]=[0.3183098861837907,
0.31724, 3.152187618207036]
```

Finalizamos esta seção com a apresentação de aproximações de π para alguns valores de n obtidas a partir do algoritmo acima implementado:

Valores de n	Aproximação para π
10^2	3.278688524590164
10^3	3.091190108191654
10^4	3.134304967873374
10^5	3.148168552944325
10^6	3.139308378971029
10^7	3.141974517329875

Tabela 3.1: Aproximações Através de Buffon

3.2 Simulação do problema da agulha de Buffon-Laplace

Na Seção 1.3, foi visto que o problema da agulha de Buffon-Laplace é a generalização do problema de Buffon a um reticulado contendo dois feixes ortogonais de retas paralelas equidistantes, sendo a a distância entre as retas do primeiro feixe e b a distância entre as retas do segundo, onde o comprimento l do alfinete é menor que a e b . Foi visto também que a condição para que haja interseção do alfinete com uma ou duas retas é dada por

$$l \cdot \cos \theta + x \geq a \quad \text{ou} \quad l \cdot |\sin \theta| + y \geq b$$

Onde $x \in [0; a[$ e $y \in [0; b[$ são as distâncias horizontal e vertical da extremidade inferior do alfinete à reta mais próxima à esquerda e abaixo respectivamente, e $\theta \in]-\frac{\pi}{2}; \frac{\pi}{2}]$ é o ângulo tomado no sentido anti-horário entre uma reta horizontal e a reta que contém o alfinete. De posse destes elementos, será criada uma rotina bastante semelhante àquela vista na seção anterior.

```
(%i86) l:1;
```

```
(%o86) 1
```

```
(%i87) n:100;
```

```
(%o87) 100
```

```
(%i88) a:1.0;
```

```
(%o88) 1.0
```

```
(%i89) b:2.0;
```

```
(%o89) 2.0
```

Para obtermos números aleatórios x , y e θ nos intervalos $[0; a[$, $[0; b[$ e $]-\frac{\pi}{2}; \frac{\pi}{2}]$, respectivamente, utilizaremos as seguintes definições:

```
(%i5) x:random(a);
```

```
(%o5) 0.9138095996128959
```

```
(%i6) y:random(b);
```

```
(%o6) 0.3903692355955775
```

```
(%i7) %theta: (2*random(1.0)-1)*%pi/2;
```

```
(%o7) - 0.0176094751483804  $\pi$ 
```

Entretanto, como cada um destes parâmetros indica a posição do alfinete em um lançamento, para se gerar os n lançamentos é necessário que o máximo crie cada um destes números n vezes. Assim, utilizaremos o comando `for` para um grupo de instruções. Para tanto, basta substituir a instrução neste comando pelo grupo de instruções entre parênteses separadas por vírgula:

```
(%i8) s:0;
```

```
(%o8) 0
```

```
(%i9) for i: 1 thru n do
```

```
    (x:random(a),
```

```
    y:random(b),
```

```
    %theta: (2*random(1.0)-1)*%pi/2,
```

```
    if 1*cos(%theta) + x >= a or
```

```
    1*abs(sin(%theta))+ y >= b then s: s + 1);
```

```
(%o9) done
```

Lembre-mos que, no contexto da linguagem dos conjuntos, o significado da palavra “ou” empregada na condição acima não exclui a possibilidade de ocorrência das condições intermediárias simultaneamente. O subcomando `or` (que significa “ou” em português), utilizado dentro do comando `if`, trata exatamente do mesmo modo as condições desta natureza.

Denotaremos por **Prob** a probabilidade obtida na Seção 1.3, e por **AroxProb** a probabilidade obtida na simulação:

```
(%i10) P:s/n;
(%o10)  $\frac{4}{5}$ 
(%i11) AproxProb:float(s/n);
(%o11) 0.8
(%i12) Prob:float((2*l*(a+b)-l^2)/(%pi*a*b));
(%o12) 0.7957747154594768
```

Assim, a aproximação para π obtida é dada por:

```
(%i13) float((2*l*(a+b)-l^2)/(P*a*b));
(%o13) 3.125
```

Sintetizaremos todas as instruções ao MAXIMA aqui apresentadas com a criação do algoritmo cujo nome mais uma vez homenageia o mentor do problema:

```
(%i14) Laplace(l,n,a,b):= block([AproxPi,AproxProb],
  s:0,
  for i: 1 thru n do
    (x:random(a),
     y:random(b),
     %theta: (2*random(1.0)-1)*%pi/2,
     if l*cos(%theta) + x >= a or
        l*abs(sin(%theta))+ y >= b then s: s + 1),
    P:s/n,
    AproxPi: float((2*l*(a+b)-l^2)/(P*a*b)),
    AproxProb:float(P),
    Prob:float((2*l*(a+b)-l^2)/(%pi*a*b)),
    "[Probab., Aprox. Prob, Aprox. de Pi] "=[Prob,AproxProb,AproxPi]
  );
(%o14) Laplace(l,n,a,b) := block([AproxPi,AproxProb], s:0, for i thru n do (x:random(a),
y:random(b),  $\theta: \frac{(2\text{random}(1.0)-1)\pi}{2}$ , if  $l \cos(\theta) + x \geq a$  or  $l |\sin(\theta)| + y \geq b$  then  $s: s + 1$ ),
```

$P : \frac{s}{n}$, $AproxPi : \text{float} \left(\frac{2l(a+b)-l^2}{Pab} \right)$, $AproxProb : \text{float} (P)$, $Prob : \text{float} \left(\frac{2l(a+b)-l^2}{\pi ab} \right)$,
 [Probab., Aprox. Prob, Aprox. de Pi]=[Prob,AproxProb,AproxPi])

(%i15) Laplace(1,10000,2.0,1.0);

(%o15) [Probab., Aprox.Prob, Aprox.dePi] = [0.7957747154594768, 0.7976, 3.134403209628887]

Conclui-se esta seção com a apresentação de aproximações de π para alguns valores de n apartir do algoritmo obtido do problema de Buffon-Laplace:

Valores de n	Aproximação para π
10^2	3.205128205128205
10^3	3.144654088050314
10^4	3.134403209628887
10^5	3.133421069123269
10^6	3.142250957758092
10^7	3.14151297023894

Tabela 3.2: Aproximações Através de Laplace

Capítulo 4

O Problema das Moedas Neperianas

Este é um problema um tanto curioso, pois diferentemente do que acontece na maioria dos casos, ele é originado a partir de sua solução. Sim, é exatamente isto o que você está lendo. O problema que será apresentado neste capítulo inicia com sua solução. Tudo começou quando foi criada a rotina que determinava aproximações para π . Eu e meu orientador, Prof. Marcos Monteiro Diniz, ficamos tão entusiasmados com a descoberta que começamos a pensar no quanto seria legal se pudessemos criar um problema de probabilidade geométrica em cuja solução aparecesse o número e . Isto nos possibilitaria criar um algoritmo que determinasse aproximações para este número. O grande problema por trás de tudo então não foi o que será apresentado aqui. O desafio era criar um problema com as características mencionadas. O prof. Diniz mostrou que este número está intimamente relacionado a hipérbolas, falando dos gráficos das funções exponencial e logarítmica. Sua primeira sugestão então foi considerar o lançamento dos alfinetes em um plano coberto por hipérbolas transladadas de excentricidade constante e mesmo eixo real. Esta foi apenas uma das tentativas frustradas. Felizmente, algum tempo depois o orientador notou que ao integrar a função $y = 1/x$ temos os logaritmos neperianos no resultado. Então, a grande idéia foi pensar em um problema aonde esta função aparecesse na condição de acontecimento dos casos favoráveis. O problema pensado é mostrado a seguir e a rotina criada confirma a beleza de tal resultado.

4.1 Solucionando o problema

Para obter os resultados desejados, iniciaremos com uma versão preliminar do problema, e após uma breve discussão a respeito do mesmo, exibiremos o Problema das Moedas Neperianas. Assim, vamos analisar o seguinte enunciado:

Considere um plano coberto por um feixe de retas paralelas equidistantes a uma distância d e uma caixa com moedas de raio r . Ao retirarmos uma moeda qualquer da caixa e lançá-la sobre o plano coberto pelo feixe de paralelas, qual a probabilidade de ela intersectar uma das retas?

Inicialmente, observe que, como estamos falando de moedas, devemos ter necessariamente $r > 0$. Esta é a primeira condição imposta sobre os raios das moedas que não foi especificada no enunciado acima. Assim, pela propriedade arquimediana dos números reais, temos a garantia da existência de um natural N tal que $r > 1/N > 0$. Logo,

$$\frac{1}{N} < r \Rightarrow 0 < \frac{1}{r} < N.$$

Seja $x = 1/r$. Temos $x \in]0; N[$. Observe que como não foi especificado o comportamento dos raios na caixa, não sabemos ao certo como estão distribuídos os números x no intervalo $]0; N[$. Por exemplo, supondo que $N = 100$, se os raios fossem tomados como $r = 1/n$, $n = 1, 2, \dots, N - 1$, teríamos apenas uma moeda com raio maior que meio, e noventa e sete moedas com raio menor que meio. Vemos assim que estes raios não estão uniformemente distribuídos no intervalo $]0; 1[$. Entretanto, esta escolha de raios nos garante uma distribuição uniforme de seus inversos x no intervalo $]0; N[$, pois

$$x = 1/r \Rightarrow x = \frac{1}{1/n} = n, \quad n = 1, \dots, N - 1.$$

Tendo em vista a finalidade a qual o Problema das Moedas Neperianas se destina, a hipótese de que os inversos dos raios das moedas estejam distribuídos uniformemente no intervalo $]0; N[$ é essencial. A última observação diz respeito ao número natural N . Tal número pode ou não ser especificado a priori e incluído nas hipóteses do problema. O definiremos aqui apenas por finalidades práticas, pois sua determinação é indispensável na simulação computacional do problema.

De posse destas considerações, podemos então enunciar o Problema das Moedas Neperianas:

Seja α um plano coberto por um feixe de retas paralelas equidistantes a uma distância d , N um número natural e x um número real aleatório no intervalo $]0; N[$. Se uma moeda de raio $r = 1/x$ for lançada sobre o plano α , qual a probabilidade da mesma intersectar uma das retas?

Solução:

Temos $x = 1/r$ e $x \in]0; N[$. Se y é a distância do centro de uma moeda à reta mais próxima à esquerda, temos $y \in [0; d[$. Assim, um par ordenado (x, y) corresponde à uma moeda de raio $r = 1/x$ cujo centro se encontra à distância y da reta à esquerda. Portanto, uma moeda corta uma das retas se

$$y \leq r \quad \text{ou} \quad y + r \geq d.$$

Observe a figura 4.1 abaixo.

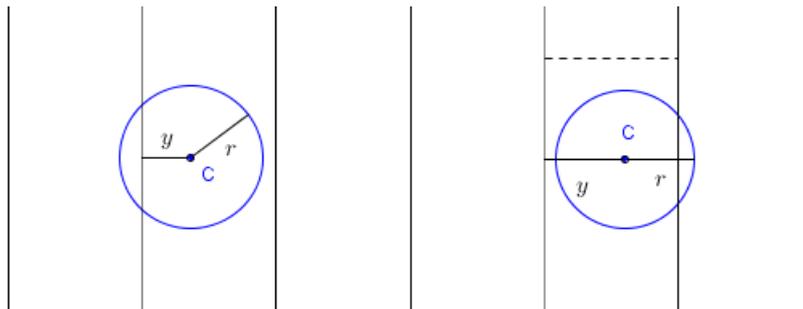


Figura 4.1: Condição de Interseção

Como $r = 1/x$, a condição de interseção pode ser reescrita da seguinte forma:

$$y \leq \frac{1}{x} \quad \text{ou} \quad y \geq d - \frac{1}{x}.$$

Portanto, o espaço de configurações é o produto cartesiano $]0; N[\times [0, d[$, ou seja, o conjunto $R : \{(x, y) \in \mathbb{R}^2 \mid 0 < x < N \text{ e } 0 \leq y < d\}$, e os pontos (x, y) que satisfazem a condição de interseção são os pontos pertencentes ao conjunto $S : \{(x, y) \in R \mid y \leq \frac{1}{x} \text{ ou } y \geq d - \frac{1}{x}\}$. A figura 4.2 Traz uma representação gráfica dos dois conjuntos.

Se A_R é a área do retângulo R , A_S é a área do conjunto S , e A_E é a área entre as hipérbolas, segue da Seção 1.1 que a probabilidade P da moeda intersectar uma das retas é dada por

$$P = \frac{A_S}{A_R} = \frac{A_R - A_E}{A_R} = 1 - \frac{A_E}{A_R}.$$

Como,

$$\begin{aligned} A_E &= \int_{\frac{2}{d}}^N \left(d - \frac{1}{x} \right) - \frac{1}{x} dx \\ &= \int_{\frac{2}{d}}^N d - \frac{2}{x} dx \\ &= dN - d \cdot \frac{2}{d} - 2 \left(\ln N - \ln \frac{2}{d} \right) \\ &= dN - 2 - 2 \ln \frac{dN}{2} \end{aligned}$$

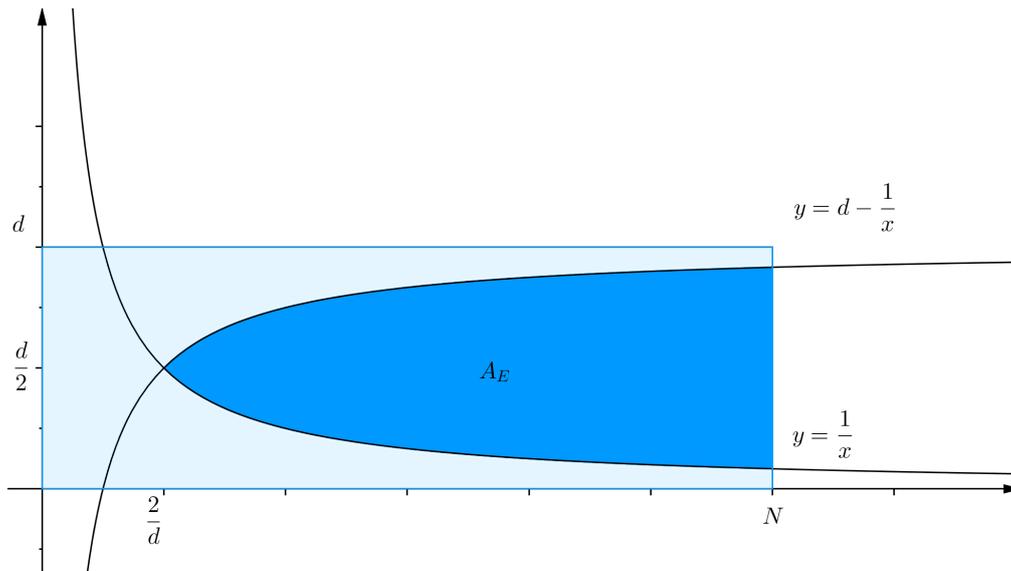


Figura 4.2: Espaço de Configurações

Segue que

$$\begin{aligned}
 P &= 1 - \frac{A_E}{A_R} \\
 &= 1 - \frac{dN - 2 - 2 \ln \frac{dN}{2}}{dN} \\
 &= \frac{2(1 + \ln \frac{dN}{2})}{dN}
 \end{aligned}$$

■

4.2 Aproximações para o número transcendental e

Da probabilidade obtida na seção anterior, tomando $d = 2$, tem-se

$$P = \frac{1 + \ln N}{N}.$$

A igualdade acima implica em uma outra bastante sugestiva:

$$NP = 1 + \ln N \Rightarrow e^{NP-1} = N \Rightarrow e = N^{\frac{1}{NP-1}}.$$

Portanto, se considerarmos o lançamento de n moedas, espera-se que o número s de vezes que uma moeda intersecte uma reta seja tal que $s/n \approx P$. A partir desta idéia, pode-se concluir o tão esperado resultado:

$$e \approx N^{\frac{1}{Ns/n-1}}.$$

Assim como no Capítulo 3, será criado um algoritmo para determinação de algumas destas aproximações. Começamos a simulação para mil moedas de raio maior que $1/100$.

```
(%i1) N:100.0;
```

```
(%o1) 100.0
```

```
(%i2) d:2.0;
```

```
(%o2) 2.0
```

```
(%i3) n:1000;
```

```
(%o3) 1000
```

Lembramos que um número aleatório y no intervalo $[0; d[$ pode ser determinado pelo comando `random(d)`. Análogamente, para determinar um número aleatório x no intervalo $]0; N[$, usamos o comando `random(N)`. Assim, com a utilização do comando `if` aplicado a condição de interseção podemos contar o número de moedas que intersectam uma reta.

```
(%i4) s:0;
```

```
(%o4) 0
```

```
(%i5) for i: 1 thru n do
      (y:random(d),
       x:random(N),
       if y <= 1/x or
       y >= d-1/x then s: s + 1);
```

```
(%o5) done
```

```
(%i6) P:s/n;
```

```
(%o6)  $\frac{53}{1000}$ 
```

Portanto, a probabilidade aproximada é dada por:

```
(%i7) float(P);
```

```
(%o7) 0.053
```

```
(%i8) float((1+log(N))/N);
```

```
(%o8) 0.05605170185988092
```

Tendo em vista que para $d = 2$, a probabilidade P é dada por $P = (1 + \ln(N))/N$, e que no MAXIMA, o logaritmo natural $\ln(x)$ é denotado por `log(x)`. Comparando os resultados nota-se que o valor obtido é razoavelmente aproximado. Com isto, obtem-se o belíssimo resultado:

```
(%i9) e:N^(1/(P*N-1));
(%o9) 2.918208237644093
```

Que é igualmente aproximado do número e , tendo em vista o número de lançamentos:

```
(%i10) float(%e);
(%o10) 2.718281828459045
```

Com a utilização do comando `block`, vamos reunir todas as instruções desenvolvidas aqui em um único comando que nos forneça os resultados esperados. O comando se chamará Neperiano.

```
(%i11) Neperiano(n,d,N):=block([e,AproxProb,Prob],
    s:0,
    for i: 1 thru n do
        (y:random(d),
         x:random(N),
         if y <= 1/x or
            y >= d-1/x then s: s + 1),
    P: s/n,
    AproxProb: float(P),
    Prob: float(2*(1+log(N*d/2))/(N*d)),
    e: float((N*d/2)^(1/(P*N*d/2-1))),
    "[Aprox. de e, Aprox. Prob., Prob]"= [e,AproxProb,Prob]);
```

```
(%o11) Neperiano(n,d,N) := block([e,AproxProb,Prob], s : 0, for i thru n do
(y : random(d), x : random(N), if y <= 1/x or y >= d - 1/x then s : s + 1), P : s/n,
AproxProb : float(P), Prob : float(2*(1+log(N*d/2))/(N*d)), e : float((N*d/2)^(1/(P*N*d/2-1))),
[Aprox. de e, Aprox. Prob., Prob]=[e,AproxProb,Prob])
```

```
(%i12) Neperiano(10000,2.0,5.0);
```

```
(%o12) [Aprox.dee, Aprox.Prob., Prob] = [2.738671288650087, 0.5195, 0.5218875824868201]
```

Encerramos o capítulo com a apresentação de um quadro contendo aproximações para e

obtidas da rotina acima implementada, considerando $d = 2$ e $N = 10$.

Valores de n	Aproximação
10^4	2.708936122068192
10^5	2.720967729290251
10^6	2.719571276505056
10^7	2.716478818834968
10^8	2.718135174852237
10^9	2.718202024116304

Tabela 4.1: Aproximações de e

Considerações Finais

Foram abordados três problemas de probabilidade geométrica, sendo dois deles os precursores deste campo do conhecimento: O problema de Buffon e o de Buffon-Laplace. Juntamente com a elaboração do segundo problema, Laplace cria uma idéia de como estimar de forma experimental o número transcendental π , utilizando para tanto a solução do primeiro problema. Baseando-se nestas idéias, o presente trabalho apresenta um novo problema de probabilidade geométrica, cuja solução possibilita a criação de um método para estimar o número transcendental e .

Os meios utilizados para estimar estes números é a implementação computacional de algoritmos no aplicativo matemático MAXIMA. Acredita-se que os algoritmos podem ser melhorados nos sentidos temporal, de estimativa e esforço computacional, se estes forem implementados de outras formas e outras ferramentas computacionais forem utilizadas. O MAXIMA foi empregado neste trabalho por se tratar de uma ferramenta de livre acesso, e por haver certa familiaridade do autor com o aplicativo.

Ressalta-se que a probabilidade obtida no terceiro problema, diferentemente das obtidas nos demais, não pode ser determinada de forma empírica por conter expressões exponenciais. Portanto, para este problema é essencial uma ferramenta computacional.

Acredita-se que novos problemas de probabilidade geométrica podem e devem ser desenvolvidos, sejam eles objetivando aplicações ou não. A idéia é apenas desenvolver e enriquecer ainda mais este belíssimo campo da matemática. Assim como Buffon e Laplace cumpriram essa missão, o trabalho aqui apresentado também deixa sua colaboração. É deixado para os colegas de computação as melhorias e aplicações nas áreas de programação.

Referências Bibliográficas

- [1] BOYER, C. B; *História da Matemática*. Editora Edgard Blücher Ltda. São Paulo. 1974. 488p.
- [2] KHAMIS, H. J; *On Buffon's Needle Problem Using Concentric Circles*. PI MU EPSILON Journal, Volume 8, Honorary Mathematical Fraternity, 1987.
- [3] LOPES, J. M; FILHO, I. F. B; SALVADOR, J. A; *O conceito de Probabilidade Geométrica por meio do uso de Fractais*. CNMAC2012 - XXXIV Congresso Nacional de Matemática Aplicada e Computacional, 17 a 21 de setembro 2012. Águas de Lindóia, São Paulo, Brasil.
- [4] MATHWORLD; *The Buffon-Laplace Needle Problem*. Disponível em <http://mathworld.wolfram.com/Buffon-LaplaceNeedleProblem.html>. Acesso em julho de 2013.
- [5] SANTALÓ, Luis A.; *Integral Geometry and Geometric Probability*. Cambridge Mathematical Library, 1976.
- [6] SANTOS, Bruna; *Introdução ao software MAXIMA*. Centro de Matemática da Universidade do Porto, Dezembro de 2009.
- [7] SCHEIDT, Alexandre; HACHMANN, Diego Rodrigo; *Agulha de Buffon*. Anais do I Seminário Internacional de Ciência, Tecnologia e Ambiente, 2009. UNIOESTE, Cascavel, Paraná, Brasil.
- [8] SOLOMON, Hebert; *Geometric Probability*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1978.
- [9] SOURCEFORGE; *Maxima Manual*. Disponível em <http://maxima.sourceforge.net/docs/manual/pt/maxima.html>. Acesso em 18 de julho de 2015.
- [10] VAZ, Cristina Lúcia Dias; *Aprendendo o aplicativo MAXIMA*. UFPA, Julho-2008.