

# MÉTODOS APROXIMADOS PARA A CONSTRUÇÃO DA ESPIRAL LOGARÍTMICA EM PROGRAMAS DE GEOMETRIA DINÂMICA

---

Daniel Wyllie Lacerda  
Rodrigues<sup>1</sup>

Vânia Ribas Ulbricht<sup>2</sup>

Resumo

Este trabalho apresenta um estudo sobre as diferentes possibilidades para a construção de uma espiral logarítmica em programas de geometria dinâmica. São divulgados os dois principais métodos de desenho da espiral: (1) baseado em triângulos áureos e (2) baseado em retângulos áureos. Como ambos são flexíveis para a adoção de vários algoritmos, são sugeridas mais duas soluções para cada caso específico. Obtém-se então (sem restringir outras possibilidades) um total de quatro alternativas plausíveis. Mostra-se também, que num programa de geometria dinâmica, é possível criar uma ferramenta associada a cada solução. Os passos necessários para a construção completa da espiral são escondidos, e o resultado passa a depender apenas da indicação de poucos parâmetros iniciais como, por exemplo, dois pontos.

Palavras Chave: algoritmos, espiral logarítmica, geometria dinâmica.

---

<sup>1</sup> Eng. de Computação, mestrando do Programa de Pós-Graduação em Engenharia de Produção, UFSC, dep. de

Eng. da Prod. - Campus Universitário - Trindade - Florianópolis - SC - 88040-900.  
email:dwyllie@hotmail.com

<sup>2</sup> Professora Titular de Geometria Descritiva da UFSC/EGR, professora do programa de Pós-Graduação em Eng. de Produção/PPGE/UFSC - Campus Universitário - Trindade - Florianópolis - SC - 88040-900.  
e-mail:ulbricht@mbox1.ufsc.br

## Abstract

This work analyses the different possibilities for the construction of a logarithmic spiral within dynamic geometry software. The two major spiral drawing methods are presented: (1) the golden triangle method and (2) the golden rectangle method. Since both methods are open to the use of several algorithms, two additional solutions are suggested for each specific case, making up four plausible alternatives. It is also shown that it is possible to create in a dynamic software a tool associated to each solution. The steps needed for the completion of the spiral are hidden and the solution becomes dependent on just a few initial parameters as, for example, two points.

Key words: algorithms, logarithmic spiral, dynamic geometry.

## Introdução

A natureza está cheia de exemplos que apontam para a existência de uma unidade ou ordem comum, cuja explicação só é encontrada através da matemática. Ao longo dos tempos, a chamada seção áurea, também conhecida como proporção divina, tem sido uma constante fonte de inspiração para artistas e arquitetos (GHYKA, 1977; HUNTLEY apud WELLS, 1987; DOCZI, 1990). A análise do formato de galáxias, de conchas, e também do centro de margaridas e girassóis mostra que um certo tipo de espiral representa um caso particular desse amplo processo para a formação de padrões. Trata-se da espiral logarítmica.

Como parte de uma proposta de ensino, sugere-se que com o auxílio do computador e através de conhecimentos em geometria plana, seja criada uma ferramenta

ou comando que gere uma espiral logarítmica aproximada a partir de apenas alguns parâmetros de entrada. Primeiramente é feita uma análise da construção, expondo-se os procedimentos necessários para se construir a espiral. Depois são escolhidos dois programas de geometria dinâmica, nos quais são implementadas as diferentes soluções para a criação do modelo: Cabri Géomètre II (LABORDE & BELLEMAIN, 1994) e The Geometer's Sketchpad 3 (JACKIN, 1995).

## Características da Espiral Logarítmica

Para classificar uma espiral como equiangular ou logarítmica, é necessário que qualquer um de seus vetores radiais determine um ângulo constante com a sua tangente num ponto de interseção da curva (Figura 1). Entretanto, a propriedade mais surpreendente de uma espiral desse gênero se refere ao crescimento roto-homotético de seus estágios (GHYKA, 1977). Para quaisquer dois arcos escolhidos em porções aleatórias, verifica-se a semelhança de seus formatos, pois varia-se em tamanho, mas mantém-se a forma.

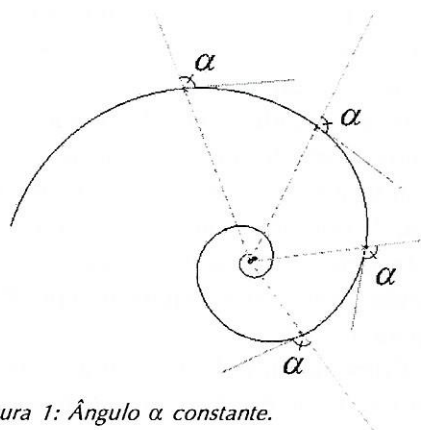


Figura 1: Ângulo  $\alpha$  constante.

Dois métodos fundamentais são apresentados para se construir uma espiral logarítmica aproximada. O primeiro depen-

de da determinação de triângulos áureos<sup>1</sup> e o segundo depende da determinação de retângulos áureos<sup>2</sup>.

### Método 1 - Triângulos Áureos

Em uma estrela de cinco pontas, várias relações indicam o valor da seção áurea. Na Figura 2 pode-se verificar que:

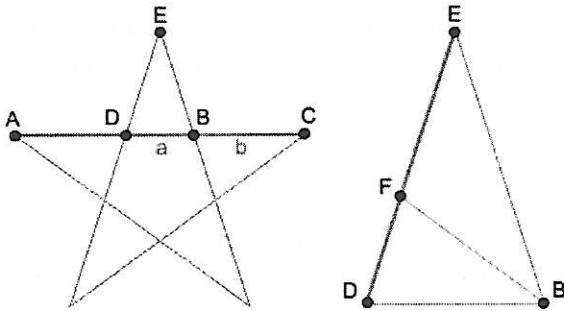


Figura 2: Estrela de cinco pontas e triângulo áureo.

Quando o triângulo áureo BDE é isolado, e a bissetriz do ângulo DBE é determinada, encontra-se novamente o valor de  $\phi$  pela razão  $EF/DF$ . Nesse caso, pode-se afirmar que tanto o triângulo original BDE, quanto o recém descoberto BDF são equivalentes. A Figura 3 mostra que, quando este procedimento de divisão é repetido várias vezes, obtém-se uma seqüência de triângulos em roto-homotetia. Nessa configuração, podem ser traçados arcos de círculo, cujos ângulos de abertura correspondem a  $108^\circ$ . Define-se, assim, o primeiro método de construção da espiral, no qual os trechos da curva são determinados por estes arcos em recursão (centro em  $P_n$  e passando por  $P_{n-3}$  e  $P_{n-2}$  para  $n \geq 3$ ).

Como um dos objetivos deste trabalho é mostrar a possibilidade de construção de uma ferramenta automatizada para uso em

programas de Geometria dinâmica, são propostos dois algoritmos baseados nesta linha de raciocínio (Figuras 4 e 5):

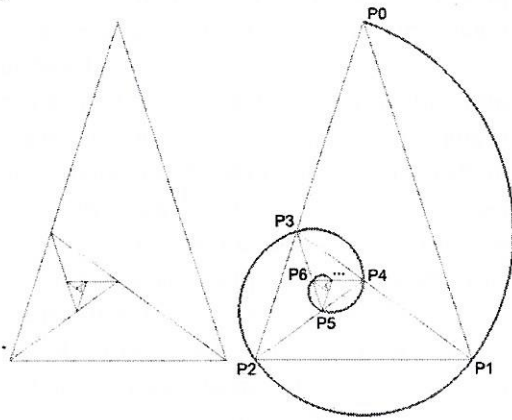


Figura 3: Conjunto de triângulos em roto-homotetia.

### Algoritmo 1A

1. Criar o ponto  $A_0$ .
2. Criar o ponto  $B_0$ .
3. Traçar o segmento  $s_0$ : Ligação dos pontos  $A_0$  e  $B_0$ .
4. Criar o ponto  $C_0$ : Rotação de  $A_0$  (centro  $B_0$ ) por um ângulo de  $36^\circ$ .
5. Traçar a bissetriz  $r_{01}$ : Definida pelo ângulo  $A_0C_0B_0$ .
6. Criar o ponto  $D_0$ : Interseção de  $s_0$  com  $r_{01}$ .
7. Traçar a bissetriz  $r_{02}$ : Definida pelo ângulo  $B_0D_0C_0$ .
8. Traçar o círculo  $c_0$ : Centro em  $D_0$  e raio = comprimento de  $B_0D_0$ .
9. Criar o ponto  $E_0$ : Interseção de  $c_0$  com  $r_{02}$ .
10. Traçar o arco  $a_0$ : Definido pelos pontos  $B_0$ ,  $E_0$  e  $C_0$ .
11. Considerar os pontos  $D_0$  e  $C_0$  como  $A_1$  e  $B_1$  e repetir os passos de 3 a 11.

<sup>1</sup> Triângulo isósceles cujos ângulos valem  $72^\circ$ ,  $72^\circ$  e  $36^\circ$ .

<sup>2</sup> Retângulo cuja razão de seus lados adjacentes corresponde ao valor da seção áurea.

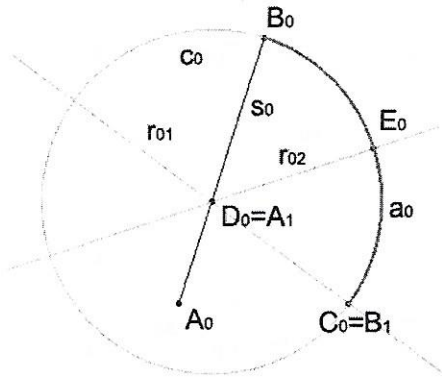


Figura 4: Imagem gerada pela primeira passagem do algoritmo 1A.

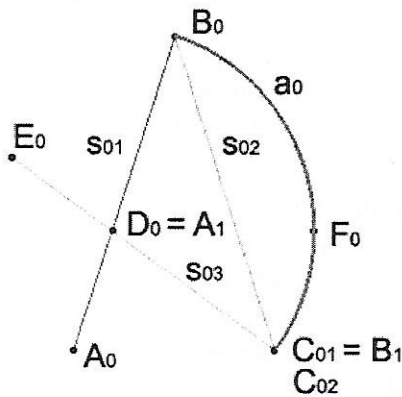


Figura 5: Imagem gerada pela primeira passagem do algoritmo 1B.

### Método 2: Retângulos Áureos

Cada retângulo áureo pode ser dividido em um quadrado e em outro retângulo semelhante. Se para cada novo retângulo encontrado, o mesmo processo de divisão for

### Algoritmo 1B

1. Criar o ponto  $A_0$ .
2. Criar o ponto  $B_0$ .
3. Criar o ponto  $C_{01}$ : Rotação de  $A_0$  (centro  $B_0$ ) por um ângulo de  $36^\circ$ .
4. Traçar o segmento  $s_{01}$ : Ligação dos pontos  $A_0$  e  $B_0$ .
5. Traçar o segmento  $s_{02}$ : Ligação dos pontos  $B_0$  e  $C_{01}$ .
6. Traçar o segmento  $s_{03}$ : Rotação de  $s_{02}$  (centro  $C_{01}$ ) por um ângulo de  $36^\circ$ .
7. Criar o ponto  $D_0$ : Interseção de  $s_{01}$  com  $s_{03}$ .
8. Criar o ponto  $F_0$ : Rotação de  $C_0$  (centro  $D_0$ ) por um ângulo de  $36^\circ$ .
9. Traçar o arco  $a_0$ : Definido pelos pontos  $B_0$ ,  $F_0$  e  $C_{01}$ .
10. Considerar os pontos  $D_0$  e  $C_{01}$  como  $A_1$  e  $B_1$  e repetir os passos de 3 a 9.

repetido, deve-se formar uma nova seqüência de polígonos em roto-homotetia. Entretanto, esta última difere da seqüência do método anterior pelo fato destes polígonos serem retângulos, não mais triângulos. Como mostra a Figura 6, a construção da espiral também é feita de forma recursiva. Cada passo da curva é formado por um arco circular de  $90^\circ$ .

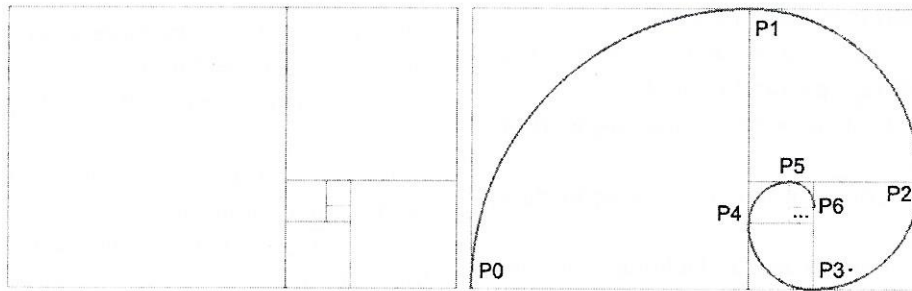


Figura 6: Conjunto de quadrados e retângulos em roto-homotetia

Outros dois algoritmos de construção são propostos (Figuras 7 e 8):

#### Algoritmo 2A

1. Criar o ponto  $A_0$ .
2. Criar o ponto  $B_0$ .
3. Traçar o segmento  $s_{01}$ : Ligação dos pontos  $A_0$  e  $B_0$ .
4. Traçar o círculo  $c_{01}$ : Centro em  $A_0$  e raio = comprimento de  $A_0B_0$ .
5. Traçar a reta  $r_{01}$ : Perpendicular ao segmento  $s_{01}$ , passando por  $B_0$ .
6. Criar o ponto  $C_0$ : Interseção de  $r_{01}$  com  $c_{01}$ .
7. Traçar a reta  $r_{02}$ : Paralela ao segmento  $s_{01}$ , passando por  $C_0$ .
8. Traçar a reta  $r_{03}$ : Perpendicular ao segmento  $s_{01}$ , passando por  $A_0$ .
9. Criar o ponto  $D_0$ : Interseção de  $r_{02}$  com  $r_{03}$ .
10. Criar o ponto  $E_0$ : Ponto-médio entre  $C_0$  e  $D_0$ .
11. Traçar o círculo  $c_{02}$ : Centro em  $D_0$  e raio = comprimento de  $D_0E_0$ .
12. Traçar o segmento  $s_{02}$ : Ligação dos pontos  $A_0$  e  $E_0$ .
13. Criar o ponto  $F_0$ : Interseção de  $c_{02}$  com  $s_{02}$ .
14. Traçar a circunferência  $c_{03}$ : Centro em  $A_0$  e raio = comprimento de  $A_0F_0$ .
15. Criar o ponto  $G_0$ : Interseção de  $s_{01}$  com  $c_{03}$ .
16. Traçar a reta  $r_{04}$ : Perpendicular ao segmento  $s_{01}$ , passando por  $G_0$ .
17. Traçar a reta  $r_{05}$ : Paralela ao segmento  $s_{01}$ , passando por  $H_0$ .
18. Criar o ponto  $I_0$ : Interseção de  $r_{04}$  com  $r_{05}$ .
19. Criar o ponto  $J_0$ : Interseção de  $s_{02}$  com  $c_{02}$ .
20. Traçar o arco  $a_0$ : Definido pelos pontos  $C_0$ ,  $J_0$  e  $A_0$ .
21. Considerar os pontos  $I_0$  e  $G_0$  como

$A_1$  e  $B_1$  e repetir os passos de 3 a 21.

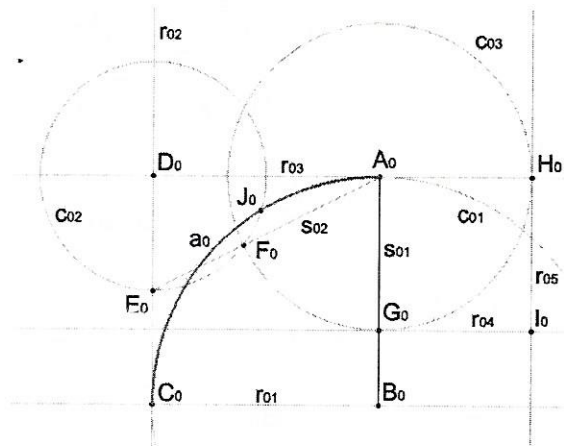


Figura 7: Imagem gerada pela primeira passagem do algoritmo 2A.

#### Algoritmo 2B

1. Criar o ponto  $A_0$ .
  2. Criar o ponto  $B_0$ .
  3. Traçar o segmento  $s_0$ : Ligação dos pontos  $A_0$  e  $B_0$ .
  4. Traçar a reta  $r_{01}$ : Perpendicular com o segmento  $s_0$ , passando por  $B_0$ .
  5. Criação do ponto  $C_0$ : Homotetia do ponto  $B_0$  por  $A_0$  com fator = 0.61803.
  6. Traçar o círculo  $c_0$ : Centro em  $B_0$  e raio = comprimento de  $A_0B_0$ .
  7. Criar o ponto  $D_0$ : Interseção de  $c_0$  com  $r_{01}$  (pela esquerda).
  8. Criar o ponto  $E_0$ : Interseção de  $c_0$  com  $r_{01}$  (pela direita).
  9. Criar o ponto  $F_0$ : Homotetia do ponto  $D_0$  por  $A_0$  com fator = 0.61803.
  10. Traçar a reta  $r_{02}$ : Passando por  $C_0$  e  $F_0$ .
  11. Criar o ponto  $G_0$ : Interseção de  $c_0$  com  $r_{02}$  (pela esquerda).
  12. Traçar o arco  $a_0$ : Definido pelos pontos  $A_0$ ,  $G_0$  e  $D_0$ .
- Considerar os pontos  $F_0$  e  $C_0$  como  $A_1$  e  $B_1$  e repetir os passos de 3 a 12.

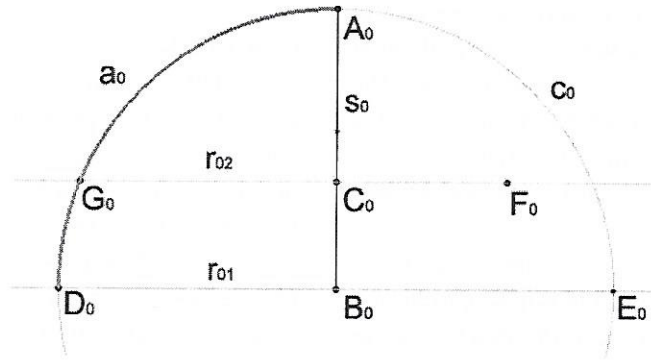


Figura 8: Imagem gerada pela primeira passagem do algoritmo 2B.

### Macros e scripts

Os programas de geometria dinâmica utilizados são o *Cabri Géomètre II* e o *The Geometer's Sketchpad 3*. Os dois possuem recursos para a construção de ferramentas personalizadas. Os procedimentos criados pelo usuário no *Cabri* e no *Sketchpad* são respectivamente denominados *macros* e *scripts*. Em termos conceituais, ambos têm um significado comum. Um *script* corresponde à gravação de uma seqüência de construções geométricas de objetos e das relações entre os mesmos. Quando ele é disparado pelo usuário, as figuras são desenhadas automaticamente na tela.

### Fatores de escolha dos algoritmos

*A atividade da programação estruturada da solução informática (ou de uma das soluções possíveis) para um problema dado é a única capaz de explorar a dimensão nova (o método da resolução ordenada) aberta pelo computador na paisagem escolar. Não há razão nenhuma para que essa programação seja de nível ótimo, do ponto*

*de vista da eficácia, da elegância, da rapidez ou da utilização integral das possibilidades dos computadores que hoje ingressam nas escolas. (VITALE, 1991:20)*

A escolha do algoritmo "ideal" para a construção de um procedimento realizado por um aluno, em um ambiente de ensino, é uma questão polêmica. Quando VITALE (1991) comentou sobre o uso da programação estruturada para a resolução de problemas no âmbito educacional, tornou pública a sua idéia de que o aluno deve privilegiar a construção de procedimentos explícitos, claros e transparentes, ao invés de se preocupar com a escolha de uma solução "ótima" para o computador. As justificativas que fundamentam essa afirmação estão ligadas aos resultados de seus experimentos realizados com o LOGO de PAPERT (VITALE, 1991).

Os programas de geometria dinâmica, ao contrário do LOGO, utilizam um estilo de interação baseado em WIMP<sup>5</sup>. Em vez de digitar uma série específica de comandos, o usuário é guiado, com a ajuda da

<sup>5</sup> WIMP é um acrônimo da língua inglesa que conjuga os conceitos de Janelas, Ícones, Menus e Apontadores (NIELSEN, 1993).

manipulação direta, por ícones e menus até as ferramentas do sistema. Seja qual for o micromundo<sup>6</sup> (*Cabri Géomètre II* ou *The Geometer's Sketchpad 3*) em uso, são estas últimas que proporcionam a criação e a modificação dos elementos geométricos em cada modelo.

O tempo de resposta é importante porque está sendo usada a manipulação direta. Afinal, se a geometria é dinâmica, não se deve descartar uma solução intermediária, conjugando a transparência com a rapidez. As *N* passagens do algoritmo precisam se adequar ao limite da memória e ao tempo de processamento da aplicação no computador. Ao clicar e arrastar os pontos de controle de uma espiral logarítmica, julga-se desejável que toda a estrutura possa ser recalculada e reposicionada num tempo curto o suficiente para manter a impressão de movimento contínuo<sup>7</sup>.

De acordo com SHNEIDERMAN (1998), quando é procurada a solução para um problema complexo, e existem várias possibilidades para resolvê-lo, os usuários passam a adaptar a forma com que eles trabalham de modo a diminuir o tempo de resposta que o sistema oferece. Será que a produtividade num ambiente de ensino deve ser a mesma procurada num local de trabalho? Não se trata de pressionar os alunos para que eles apresentem soluções em tempo recorde. Quando lidamos com a criação e o uso de ferramentas (que disparam a execução de algoritmos estruturados), existem duas etapas com propósitos diferentes:

1-Criação da ferramenta - o usuário gasta o tempo necessário para raciocinar em cima de um problema que deseja resolver. Ele interage com o sistema, reflete, faz

questionamentos e descobre uma solução com a ajuda dos recursos que lhe são oferecidos. Uma vez que ele tenha descoberto e compreendido todo o processo de construção, é possível então criar uma ferramenta que esconde e engloba a seqüência completa dos passos para a resolução do modelo. No caso da espiral logarítmica, a solução representa um arco isolado que será repetido quantas vezes o usuário desejar.

2- Uso da ferramenta - o seu processo de criação foi elaborado na etapa anterior, e o usuário não está mais envolvido com o raciocínio que o levou à construção do algoritmo inicial. Sua intenção, ao selecionar a nova ferramenta, é obter o traçado automaticamente pela indicação de poucos parâmetros iniciais, não deixando vestígios da maneira pela qual foi elaborado. Ele pode até mesmo construir novos elementos a partir do modelo obtido. Entretanto, se o algoritmo inicial for ineficaz, à medida que novos elementos forem surgindo poderá haver uma queda na qualidade de serviço da aplicação. O próprio usuário deverá pensar em uma solução mais adequada.

Como poderão ser avaliados os fatores que contribuem para a eficácia do algoritmo? Qual deles aparenta ser o mais adequado? As estatísticas trazem as pistas (Tabelas 1, 2, 3, e 4).

O algoritmo 2A possui 9 passos por etapa, o 1B tem 20 passos. O primeiro é mais otimizado pois, além de ser menor, cria poucas instâncias de objetos. Entretanto, o algoritmo 1B só precisa de dois pontos iniciais, o 2A requer mais um parâmetro (ângulo de 36°). Se é possível construir uma espiral

<sup>6</sup> Micromundo é um ambiente de aprendizado. Ele se baseia em regras e permite a exploração criativa dos conceitos trabalhados pelos aprendizes (MADDUX et al., 1997).

<sup>7</sup> Para manter a ilusão de movimento, deve haver uma taxa mínima de 24 quadros por segundo. Este é o padrão usado pela indústria cinematográfica.

logarítmica com apenas dois pontos de entrada, que sentido faz acrescentar mais um valor numérico? A resposta é óbvia. Eventualmente, as soluções que dependem de

mais parâmetros iniciais também possuem um menor número de passos. A comparação do algoritmo 2B com o 1B é mais um exemplo que se enquadra neste caso.

Objetos	Ocorrências	Quantidade
Ponto	$A_0, B_0, C_0, D_0, E_0$	5
Reta	$r_{01}, r_{02}$	2
Segmento	$s_0$	1
Círculo	$c_0$	1
Arco	$a_0$	1
Dado Numérico	$36^\circ$	1

Dados Iniciais	$A_0, B_0$ e ângulo = $36^\circ$
Número de passos por etapa	10

Tabela 1: Estatísticas do algoritmo 1A.

Objetos	Ocorrências	Quantidade
Ponto	$A_0, B_0, C_{01}, C_{02}, D_0, E_0, F_0$	6
Reta	-	0
Segmento	$s_{01}, s_{02}, s_{03}$	3
Círculo	-	0
Arco	$a_0$	1
Dado Numérico	$36^\circ$	1

Dados Iniciais	$A_0, B_0$ e ângulo = $36^\circ$
Número de passos por etapa	9

Tabela 2: Estatísticas do algoritmo 2A.

Objetos	Ocorrências	Quantidade
Ponto	$A_0, B_0, C_0, \dots, H_0, I_0, J_0$	10
Reta	$r_{01}, r_{02}, r_{03}, r_{04}, r_{05}$	5
Segmento	$s_{01}, s_{02}$	2
Círculo	$c_{01}, c_{02}, c_{03}$	3
Arco	$a_0$	1
Dado Numérico	-	0

Dados Iniciais	$A_0$ e $B_0$
Número de passos por etapa	20

Tabela 3: Estatísticas do algoritmo 1B.



Objetos	Ocorrências	Quantidade
Ponto	$A_0, B_0, C_0, D_0, E_0, F_0, G_0$	7
Reta	$r_{01}, r_{02}$	2
Segmento	$s_0$	1
Círculo	$c_0$	1
Arco	$a_0$	1
Dado Numérico	0.61803	1

Dados Iniciais	$A_0$ e $B_0$ e valor = 0.62
Número de passos por etapa	12

Tabela 4: Estatísticas do algoritmo 2B.

## Cabri Géomètre II

Ao construir os quatro algoritmos da espiral logarítmica no Cabri Géomètre II, podem ser observadas algumas particularidades deste programa. As principais são:

- Utiliza-se a sintaxe pré-fixada para construir cada passo do algoritmo. Primeiro são selecionadas as ferramentas, depois os objetos.
- Para criar uma macro, o usuário deve escolher os objetos iniciais, depois os objetos finais e somente então definir a sua identidade.
- Já que são escolhidos os objetos finais, não há necessidade de esconder os objetos auxiliares da construção.
- Uma macro não mostra os passos de execução quando é utilizada, tornando-se indecifrável para o usuário que não a criou.
- Para utilizar uma macro, basta o usuário selecionar a ferramenta que a caracteriza e depois indicar os objetos iniciais. O programa desenha automaticamente o modelo na tela (Figura 9).
- Não existe o conceito de macros recursivas. Para criar N passos da espiral logarítmica, o usuário repete a sua chamada N vezes.

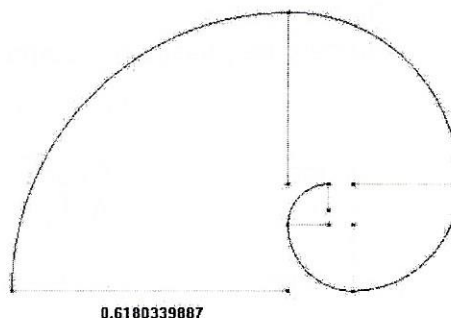


Figura 9: Resultado de uma macro (Cabri Géomètre II).

## The Geometer's Sketchpad 3

O programa The Geometer's Sketchpad 3 também possui características singulares. A construção dos quatro algoritmos propostos indicam as seguintes:

- Utiliza-se a sintaxe pós-fixada para construir cada passo do algoritmo. Primeiro são selecionados os objetos, depois as ferramentas.
- Existem duas possibilidades para gerar um script. (1) O usuário constrói o modelo, seleciona todos os seus elementos, e depois identifica o novo script. (2) O usuário dispara a gravação de seus atos, constrói o modelo e depois pára tudo, identificando o novo script.
- É preciso esconder os objetos auxi-

liares de construção, pois ao contrário do Cabri, não há uma definição de quais sejam os objetos finais (Figura 10).

- Para utilizar um script, também existem duas possibilidades diferentes: (1) o programa o executa automaticamente. (2) o programa mostra passo a passo a execução

do algoritmo associado (Figura 11).

- Existe o conceito de recursão para scripts. Ao criar a espiral logarítmica, o usuário só precisa requisitar uma única vez o script em questão. Cabe a ele decidir o grau de recursão desejado. A Figura 10 indica um grau de recursão igual a 4.

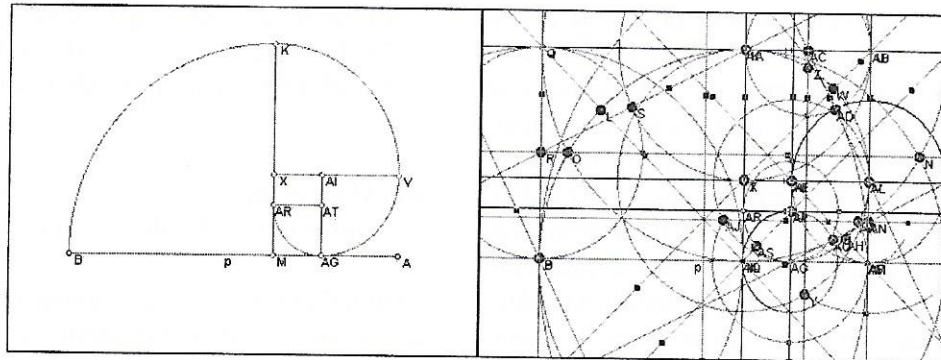


Figura 10: Objetos escondidos à esquerda e visíveis à direita (The Geometer's Sketchpad 3).

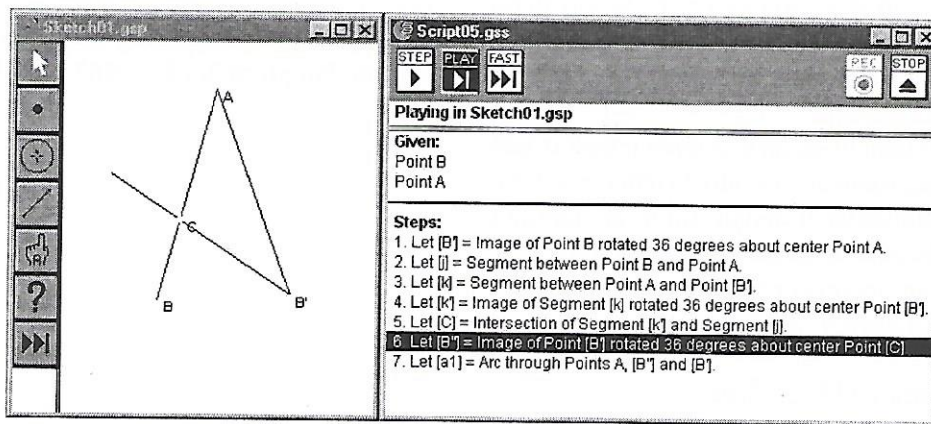


Figura 11: Script executado passo a passo (The Geometer's Sketchpad 3).

## Conclusão

As espirais logarítmicas são encontradas em vários elementos da natureza, desde galáxias até os centros de girassóis. Uma espiral deste gênero é caracterizada pelo crescimento roto-homotético de seus estágios. A partir do traçado de triângulos ou retângulos áureos, é possível obter construções aproximadas desta classe de curvas.

Neste trabalho foram propostos, com o auxílio de programas de geometria dinâmica, quatro algoritmos diferentes para se traçar uma espiral logarítmica. Estas alternativas foram buscadas para a criação de uma ferramenta capaz de automatizar a tarefa de construção. No decorrer deste processo, questionamentos sobre a qualidade dos algoritmos foram realizados. Acreditamos que, mesmo em um ambiente de aprendizado, os fatores que influenciam a velocidade do processamento de uma aplicação não devem ser descartados.

Foram utilizados os programas Cabri Géomètre II e The Geometer's Skecthpad 3. O traçado das espirais em cada um deles mostrou que ambos possuem diferentes abordagens de interface, apresentando vantagens e desvantagens.

Procuramos, através deste trabalho, não somente motivar o estudo do tema, mas também alimentar o debate sobre as questões da informática na educação, e a influência que a ergonomia de interfaces homem-computador exerce sobre as mesmas.

## Referências Bibliográficas

- DOCZI, György. O poder dos limites. São Paulo: Mercuryo, 1990.
- GHYKA, Matila. The geometry of art and life. New York: Dover, 1978.
- JACKIN, Nicholas. The Geometer's skecthpad. Berkeley: Key Curriculum Press, 1995.
- LABORDE, Jean-Marie & BELLEMAIN, Franck. Cabri-Geometry II. Dallas: Texas Instruments, 1994.
- MADDUX, Cleborne D., JOHNSON, D. LaMont, WILLIS, Jerry W. Educational Computing – Learning with Tomorrow's Technologies. Second Edition. Needham Heights, MA: Allyn & Bacon, 1997.
- NIELSEN, Jakob. Usability Engineering. San Francisco, CA: Academic Press, 1993.
- SHNEIDERMAN, Ben. Designing the User Interface. Reading, MA: Addison-Wesley, 1998.
- VITALE, Bruno. In: Ciência Hoje, vol. 13, nº 77, p.18-25, 1991.
- WELLS, David. The penguim dictionary of curious and interesting numbers. London: Penguim Books, 1987.