Exercícios sobre o Iterador-Pulga

Eduardo Colli, Guilherme de Azevedo Silveira

A idéia do trabalho é começar simples, com pequenas tarefas para aprender a usar o programa. Nas tarefas finais há menos especificações sobre como atingir seus objetivos.

Para começar, baixamos o programa no seu site oficial:

http://www.linux.ime.usp.br/~gas/iterador/

O ideal é que o estudante baixe o programa no computador que utiliza normalmente. É bem fácil: basta seguir as instruções indicadas no site acima.

O programa foi concebido para ser um facilitador para aquele que quer visualizar iterações, cuidando da geração das imagens (pois isso costuma ser torturante para aqueles que não gostam de programação). Ele também permite que o próprio usuário gere o *núcleo* do código (o algoritmo propriamente dito), e isto requer apenas conhecimento básico da *linguagem* Java, não de suas tecnologias. Ou seja, só é preciso conhecer comandos de fluxo (if, else, while, for, case) e comandos da biblioteca matemática padrão do Java, embora usuários avançados possam usar suas próprias bibliotecas.

1. Carregue a iteração

$$(x, y) \mapsto (1 - ax^2 + by, x) \ ,$$

conhecida como *aplicação de Hénon*. Para isso, vá em "File" → "New" e comece a preencher os dados em "Edit":

*"Basic Information": título da planilha, autores, descrição

*"Dimension": é o número de variáveis da iteração; no caso são duas (x e y). Observe que se forem mais variáveis, o programa visualizará sempre duas, que podem ser escolhidas em outra área do programa. Além disso, o programa por default dá o nome de x1, x2, etc a essas variáveis.

*"Initial values": são os valores iniciais das variáveis. Escolha valores não muito grandes.

*"Fixed parameters": são os parâmetros fixos, no caso "a" e "b". Ponha a=1.4 e b=0.3

*"Intermediate expressions": são expressões intermediárias, quando a fórmula de iteração é mais complicada. Veremos mais adiante o uso disso. Se houver alguma expressão escrita, removê-la.

*"Final expressions": para cada variável, a fórmula correspondente. Preencha no campo correspondente a x1 a expressão: 1-a*x1*x1+b*x2 e no campo correspondente a x2 a expressão: x1

*"Image options": é a quantidade de pixels da visualização. Se for muito grande pode dar problemas de memória. Se ultrapassar a tela, aparece o scroll bar para ajudar.

*"Proportional plot": obriga a que as dimensões da plotagem sigam a proporção do campo "real" de visualização, definido abaixo.

*"Grid, colors, axis, and scales": Pode ser criado um grid com valores para auxiliar na localização, escolhida sua cor, a cor do fundo. A cor dos pontos é escolhida em outro lugar do programa. Ali também se escolhe quais variáveis ou expressões intermediárias serão plotadas, e em que faixa de valores. Define-se também o número total de iterações e aquelas que não serão plotadas (trash).

Para ver o resultado, você tem duas opções: vá em "Redraw" → "Redraw", que itera a partir da condição inicial preencida acima ou clique na tela: a posição do mouse determina a condição inicial. Atenção que o novo valor é atualizado em "Initial values".

Observações:

1. em "Grid, colors, axis, and scales" você pode pedir que a tela seja limpa quando uma nova iteração for solicitada, ou não.

2. o clique do mouse pode ser desativado. Vá em "Plugins" → "Config" e desative, se quiser.

Para mudar cores dos pontos, vá em "Plugin" → "Advanced Color". Lá você pode fazer uma lista de cores e dizer de quantos em quantos iterados você quer que seja trocada a cor dos pontos (uma cor só, trocando de 1 em 1 iterado é a maneira de escolher a cor dos pontos).

Faça a seguinte experiência: ponha em $a = 1.2$ e $b = 0.2$, com duas cores na lista, trocando de 1 em 1 iterado. O que acontece?

TAREFA. Mude os parâmetros a e b e verifique o que acontece. Descreva resumida mas claramente os comportamentos observados, com figuras ilustrativas.

2. Crie a iteração a dois parâmetros

$$(x, y) \mapsto ((a - b/3)x - 5by/3, 2bx/3 + (a + b/3)y) \ .$$

TAREFA. O que acontece com a dinâmica em função de a e b? Existem valores de a e b para os quais as seqüências de iterados convergem para conjuntos distintos?
Exercício: defina o que é "convergir para um conjunto".

3. Agora veremos como usar as expressões intermediárias e o plugin de bacias de atração com as aplicações de Hénon.

Neste exercício, não plotaremos iterações da f mas da composição de dois iterados: f^2. Há duas maneiras de fazer isso: uma é calcular explicitamente a função f^2.

Exercício. Calcule f^2, explicitamente.

Outra maneira é a seguinte: (i) crie expressões intermediárias r e s, onde são atribuídas a r e s as expressões $1-a*x1*x1+b*x2$ e $x1$, respectivamente. (ii) vá nas expressões finais e atribua às variáveis $x1$ e $x2$ as expressões $1 - a * r * r + b * s$ e r, respectivamente. O que aconteceu? As expressões r e s correspondem à iteração pela aplicação de Hénon original de $x1$ e $x2$. Os novos valores de $x1$ e $x2$ correspondem à iteração pela aplicação de Hénon de r e s. Então os novos valores de $x1$ e $x2$ correspondem à composição de duas iterações de Hénon.

Agora coloque parâmetros $a = 1.2$ e $b = 0.2$ e veja o que acontece com várias condições iniciais. Quantos atratores você identifica?

O plugin de Bacia de Atração é o instrumento adequado para se visualizar o destino final de uma condição inicial. Ele vai escolher uma cor para cada pixel, onde o pixel define uma condição inicial e a cor é escolhida segundo o

atrator para o qual a órbita subseqüente converge (no sentido definido pelo exercício acima). Para isso, siga os passos abaixo.

O ponto crucial nesse plugin é que ele *não é automático*. É preciso que o usuário defina um critério para que o programa decida se uma órbita converge para determinado atrator ou não. Esse critério é definido no "Average Sampler".

Para entender como ele funciona, é preciso antes ler estes parágrafos. Em "Edit average functions" você tem, por default, duas funções ϕ_1 e ϕ_2, que dependem das variáveis. Se forem só duas variáveis, serão funções de duas variáveis. Você pode mudar essas funções, mas não precisa fazer isso agora.

Vamos pensar em duas variáveis para facilitar. Durante as iterações, os valores de $x1$ e $x2$ mudam, e em geral mudam também os valores de ϕ_1 e ϕ_2 calculados para esses pontos. O programa supõe que, se uma órbita se aproxima muito de um atrator, então *as médias* dos valores de ϕ_1 e ϕ_2 convergem para um valor. É sabido que isso não é sempre verdade, mas é raro encontrar órbitas em que isso não acontece. Então, espera-se que para cada atrator corresponda um par de médias $(\overline{\phi_1}, \overline{\phi_2})$.

O "Average Sampler" faz o seguinte: ele sorteia condições aleatórias dentro do espaço definido em "Grid, colors, etc", e para cada uma calcula o par de médias, usando o número de iterações também ali definido. As médias obtidas são plotadas no espaço de médias (não é o espaço das variáveis x1 e x2!!!).

O passo seguinte é cercar as "nuvens" de médias com polígonos e definir cores diferentes para cada nuvem. Isso associará a cada atrator a sua cor correspondente. Você pode também criar um polígono e editar para pôr em modo "reverse". Em geral, é usado assim: faça um polígono que englobe todos os outros e ponha em modo reverse. Ele serve para identificar as condições iniciais cujas órbitas vão a infinito.

Depois de escolhidas as nuvens, vá em "Execute" e veja o resultado. Na versão atual (16/09) ainda é preciso escolher, em "Basic configuration", de quantos em quantos pixels será feita a plotagem. Veja se está tudo ok com 5 pixels e depois faça uma figura com 1 pixel, que é mais demorada. (As próximas versões farão essa execução de forma diferente, aguarde alguns dias).

TAREFA. Tente identificar todos os atratores e suas *bacias de atração*. **Exercício:** defina a bacia de atração de um atrator A. Com a figura pronta você ainda pode plotar os atratores superpostos a suas bacias.

4. Crie a iteração complexa

$$z \mapsto z^2 + c \, ,$$

onde c é parâmetro complexo. Você pode escrever $z = x1 + ix2$ e $c = c1 + ic2$ e escrever a iteração como se fosse uma iteração em $\mathbb{R}^2$. Comece investigando a bacia para $c = 0$ e depois mude aos poucos o valor de c para ver o que acontece.

5. Aplique o Método de Newton em $z^3 - 1 = 0$ e obtenha as bacias de atração das três raízes.

6. Você pode traçar o diagrama de bifurcações da família quadrática ou da família $x \mapsto a \sin(\pi x)$. Para isso, defina a como variável ($x2$) e itere com $x1 = x2 * Math.sin(Math.PI * x1)$ e $x2 = x2$, mas use o seguinte truque: vá em "Edit" $\rightarrow$ "Pieces of code", e determine o seguinte código, para ser executado de 1000 em 1000 iterações (por exemplo): x2 = x2 + 0.001

No "Advanced color", pinte com a cor do fundo e com outra cor, revezando de 500 em 500 iterações. Assim, a cada mudança de parâmetro, as 500 primeiras iterações são jogadas fora.

Atenção apenas com o número de iterações: da maneira que fizemos, se começarmos com x2=0 e definirmos um milhão de iterações, conseguiremos andar até x2=1.

Adapte o esquema para que você tenha liberdade de escolher o número de iterações por parâmetro e trash.

TAREFA. Ache um intervalo de parâmetros em que predomine um atrator de período 7 e suas bifurações subseqüentes e amplie essa janela.

7. Examine a iteração $(x, y) \mapsto (y, c*(1-x^2+y^2))$ (chame de Aplicação Hiperbolóide) e a iteração $(x, y) \mapsto (y, c*(1-x^2-y^2))$ (Aplicação Parabolóide). Ache parâmetros interessantes. Descreva os fenômenos observados. Produza um diagrama de bifurcação como o anterior. Invente outros parâmetros para modificar as funções de outra forma.

8. Implemente a seguinte iteração. Pré-defina três pontos em $\mathbb{R}^2$. A cada iteração, sorteie um dos pontos e pegue a posição média entre o ponto sorteado e a posição atual. Defina essa posição como o novo ponto.

Depois altere o número de pontos e em vez do ponto médio pegue o novo ponto numa fração pré-escolhida do segmento que une a posição antiga com o ponto sorteado.

Para sortear um número inteiro de 0 a 2 você pode usar
(int) (Math.random()*3)

9. Implemente com a idéia de sorteio o fractal da "samambaia" de Barnsley. Utilize uma condição inicial no R^2 e escolha um número aleatório entre 1 e 4. De acordo com o número i escolhido, aplique a transformação linear A_i e a translação b_i:

$$x_{k+1} = A_i x_k + b_i \ .$$

$$A_1 = \begin{pmatrix} 0.2 & -0.26 \\ 0.23 & 0.22 \end{pmatrix}, b_1 = \begin{pmatrix} 0.4 \\ 0.045 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{pmatrix}, b_2 = \begin{pmatrix} 0.075 \\ 0.18 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 0 & 0 \\ 0 & 0.16 \end{pmatrix}, b_3 = \begin{pmatrix} 0.5 \\ 0 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{pmatrix}, b_4 = \begin{pmatrix} 0.575 \\ -0.086 \end{pmatrix}$$