Matemática Experimental

Esta secção é aberta a todo o leitor que deseje contribuir com problemas, algoritmos, programas, observações, técnicas, etc., para o desenvolvimento da actividade experimental em matemática. Por *experiência* em matemática encaramos o acto de recorrer a uma simulação computacional, mas que pode também ser física ou de outro tipo permitindo desenvolver uma ideia e encontrar os possíveis resultados. Os leitores são convidados a contribuir com sugestões, temas, experiências, que traduzam resultados interessantes de matemática, ou problemas que, embora simples, tenham *complexidade* e denotem igualmente a capacidade de estarem transformando a maneira de trabalhar em ciência, reintroduzindo o método experimental em matemática.

O Caos

por *J. Sousa Ramos*

1 – O Conceito de Caos

Neste primeiro número abordaremos as teorias do caos e a sua simulação no computador. Em próximos artigos trataremos de temas como fractais, atractores estranhos, autómatos celulares, redes neuronais, máquinas que aprendem, prova automática de teoremas em geometria e aritmética, etc.. Recorrendo ao computador a teoria da iteração das aplicações do intervalo e do plano apresenta conceitos e resultados simples de compreender e usar experimentalmente. O nosso objectivo é dar e explicar os algoritmos e programas necessários para o seu uso no computador.

A ciência do *caos* estende-se hoje a toda a actividade científica. O conceito de *caos* de um modo mais ou menos consciente, com mais ou menos rigor, é vulgarmente utilizado e exprime propriedades dos fenómenos sobre os quais recai cada vez mais a nossa atenção. Em ciências físicas, químicas, biológicas, sociais, políticas, etc., os fenómenos caóticos são os que mais atraem os novos investigadores. Um matemático não pode ficar indiferente a este entusiasmo crescente pelo estudo do caos, sobretudo, porque esse estudo surgiu e avança predominantemente em resultado do trabalho dos matemáticos. O principal motor desse trabalho e o que permite avançar é, sem dúvida, o uso crescente dos microcomputadores. A facilidade de utilização, capacidade gráfica e eficácia na obtenção de resultados e sua compreensão permitem trazer um número crescente de amadores e profissionais a estes novos tópicos da matemática.

Seguidamente estudamos o exemplo mais importante, que é igualmente o mais simples, de um sistema dinâmico caótico. Por sistema dinâmico entendemos um par (X, f), onde X representa o sistema ou espaço e f a dinâmica ou aplicação, que uma vez iterada, $f^k = f \circ f \circ ... \circ f$, e partindo de uma posição inicial x_0, define órbitas $(x_0, x_1, ..., x_k, ...)$, onde $x_i = f^i(x_0)$, $i \in \mathbf{N}$. O sistema dinâmico mais simples que apresenta caos é o par (I, f), onde I é o intervalo $[0,1]$ e $f : I \rightarrow I$ é a aplicação quadrática f_b:

$$f_b(x) = 4\,b\,x\,(1-x)\ ,$$

onde o parâmetro real b pertence igualmente a $[0,1]$. Neste caso, a caracterização mais simples de uma situação de caos corresponde à propriedade de o sistema, para valores de $b > 0.8925\ldots$, apresentar um número de órbitas periódicas N_k com crescimento exponencial em k (sendo k a ordem de iteração). Uma órbita é periódica de período p se $x_p = x_0$, e a função N_k é de crescimento exponencial em k se como função de k crescer mais depressa que qualquer polinómio em k, $P(k) = a_0 + a_1\,k + a_2\,k^2 + ... + a_m\,k^m$.

2 – Estudo Experimental do Caos

Consideremos a aplicação f_b do intervalo nele próprio. Para cada valor do parâmetro b o gráfico da aplicação $f_b(x)$ é uma parábola de altura b (ver fig. 1).

Exercício 1: *Construa um programa de computador que desenhe a parábola e que permita variar a sua altura (solução: ver procedimento* `iteral` *no quadro 1 relativo ao programa da* `parábola`*).*

Os programas estão escritos em *Pascal*, versão 4 ou 5 do *Turbo-Pascal* da Borland International.

Definimos *iteração* como a repetição da acção de aplicar f. Assim a iterada de ordem k de $f = f_b$ quando aplicada a um ponto inicial $x_0 = 1/2$ é dada por:

$$\begin{aligned} x_0 = 1/2 &\xrightarrow{f} x_1 = 4\,b\,1/2\,(1-1/2) = b \\ &\xrightarrow{f} x_2 = 4\,b^2\,(1-b) \\ &\xrightarrow{f} \ldots \\ &\xrightarrow{f} x_k = f^k(x_0)\ . \end{aligned}$$

Exercício 2: *Usando o computador ou uma calculadora fixe o valor de $b \in [0,1]$ e determine várias órbitas partindo de diferentes valores iniciais $x_0 \in [0,1]$. Repita o exercício para outros valores de $b \in [0,1]$. Observe os resultados. Tente encontrar órbitas periódicas de períodos pequenos (menores que 7) (solução: ver procedimento* `itera2` *e* `itera3`*).*

As órbitas periódicas que se obtêm por iteração no computador são as órbitas periódicas estáveis ou atractivas, isto é, que atraem as órbitas com condições iniciais vizinhas. As repulsivas ou instáveis, embora existam em maior número, só são possíveis de determinar recorrendo a programas mais elaborados, que poderemos tratar numa outra ocasião.

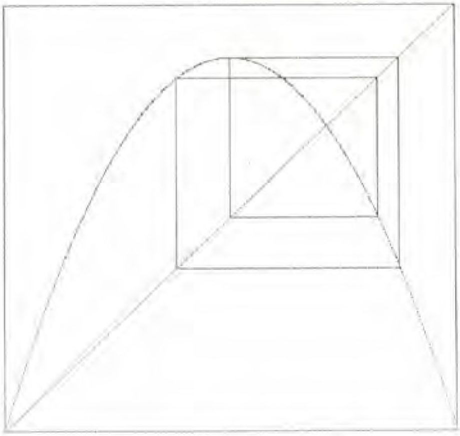

Fig. 1: Iteradas da parábola.

Exercício 3: *Construa um programa de computador que visualize a iteração geometricamente, de acordo com a figura 1 (solução: ver procedimento* `itera3`*).*

```
program parabola;
uses graph;
var gd, gm, k, I : integer;
    b, x, x1 :  real;
procedure quadro;
    begin
    cleardevice;
    initgraph(gd,gm,'');
    line(0,0,199,0);
    line(199,0,199,199);
    line(199,199,0,199);
    line(0,199,0,0);
    line(0,199,199,0);
end;
procedure itera1;
    begin
      for k:=1 to 4 do
      begin
      b:=0.8+k/20;
      for I:=1 to 400 do
      begin
      x:=i/400; x1:=4*b*x*(1-x);
      putpixel(i div 2,round(199-199*x1),7);
      end;
    end;
end;
procedure itera2;
var j :  integer;
begin
    b := 0.95;
    for I:=1 to 400 do
    begin
    x := i/400;
    for j:=1 to 2 do x:=4*b*x*(1-x);
    putpixel(i div 2,round(199-199*x),7);
    end;
    line(0,199,199,0);
    end;
procedure itera3;
   begin
   for k:=1 to 8 do
   begin
   b := 0.8+k/40;
   for I:=1 to 400 do
   begin
   x:=i/400;
   x1:=4*b*x*(1-x);
   putpixel(i div 2,round(199-199*x1),7);
   end;
   line(0,199,199,0);
   x:=0.5;
   for I:=1 to 35 do
   begin
   x1:=4*b*x*(1-x);
line(round(199*x),199-round(199*x),
round(199*x),199-round(199*x1));
line(round(199*x),199-round(199*x1),
round(199*x1),199-round(199*x1));
   x := x1;
   end; readln;
   quadro ;
   end;
end;
begin
   gd:=detect; gm:=1;
   initgraph(gd,gm,'');
   quadro; itera1;
   readln; quadro; itera2;
   readln; quadro; itera3;
   readln;
   closegraph;
end.
```

Exercício 4: *Construa um programa de computador que coloque em abcissas os valores de $b \in [1/2, 1]$ e em ordenadas os valores de $x \in [0, 1]$.*

Varie o parâmetro b de 1/2 a 1, por acréscimos de 0.01. Itere 100 vezes, para cada valor de b, partindo de um valor inicial $x_0 \in]0,1[$ arbitrário. Registe apenas as 50 últimas iteradas de acordo com o gráfico da figura 2, o qual deve obter no seu computador (solução: ver programa `bifurcação`*).*

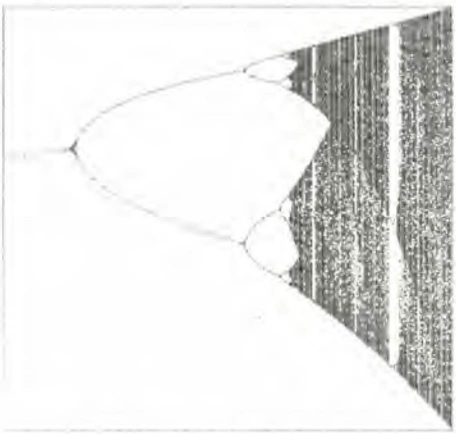

Fig. 2: Diagrama de bifurcação.

Estes gráficos chamam-se *diagramas de bifurcação* e a palavra *bifurcação* esconde um conceito matemático muito interessante. Para valores de b menores que um dado valor b_{i_1} (dito de bifurcação) o gráfico mostra que por iteração, os valores de $x_i = f^i(x_0)$ caem num ponto fixo, caracterizado por $x_{k+1} = f(x_k) = x_k$, $\forall k \in \mathbf{N}$. Para valores de $b \in]b_{i_1}, b_{i_2}[$ o gráfico mostra que por iteração os valores de x_i caem numa órbita de período 2, isto é, (x_k, x_{k+1}) pois tem-se $x_{k+2} = f(x_{k+1}) = f^2(x_k) = x_k$, $x_{k+3} = f(x_{k+2}) = f^2(x_{k+1}) = x_{k+1}$ e $x_k \neq x_{k+1}$, $\forall k \in \mathbf{N}$. Dizemos então que os valores de b_{i_1} e b_{i_2} são valores de bifurcação pois a órbita periódica atractiva de período 1 (ou ponto fixo), que existia para $b < b_{i_1}$, desaparece em $b = b_{i_1}$ (na verdade torna-se repulsiva) e dá origem para $b \in]b_{i_1}, b_{i_2}[$ a uma órbita atractiva de período 2. Para $b = b_{i_2}$ dá-se uma nova bifurcação criando-se uma órbita atractiva de período 4, e assim sucessivamente. São os fenómenos de duplicação de período — um número infinito de bifurcações.

Exercício 5: *Usando o programa realizado no exercício 4 identifique experimentalmente outros pontos de bifurcação. Descubra as órbitas periódicas de períodos 3 e 5.*

Exercício 6: *Construa um programa que coloque em abcissas o número de iteradas e em ordenadas o valor de x. Observe a variação $x(k)$ com o número k de iterações, para diversos valores do parâmetro b.*

Exercício 7: *Faça "correr" o programa construído em 4, mas colocando em abcissas outros valores de b pertencentes a um subintervalo $[b_1, b_2]$ de modo a ampliar pormenores do diagrama de bifurcação. Procure obter por ampliações deste tipo (mudanças de escala) figuras semelhantes à inicial; poderá observar fenómenos de auto-semelhança — a parte é idêntica ao todo.*

Finalmente o que é o caos? Existem várias definições de caos. Neste contexto elementar e experimental entendemos como caóticas todas as aplicações definidas para $b > b_\infty$, onde o valor de $b = b_\infty = 0.8925\ldots$ é obtido da 1ª acumulação de um número infinito de duplicações de período.

Um outro conceito interessante é o de órbita ergódica; uma tal órbita tem a propriedade de ser aperiódica.

Exercício 8: *Com o programa construído no exercício 2, procure um valor de* b, *para o qual partindo de qualquer* $x_0 \in (0,1)$, *a órbita* $f^k(x_0)$ *é aperiódica.*

Exercício 9: *Repetir os exercícios anteriores para outras famílias a um parâmetro de aplicações com apenas um máximo ou um mínimo, por exemplo,* $f_a(x) = x^2 + a$, $f_a(x) = a\,\mathrm{sen}(x)$ e $f_a(x) = a\exp(x)$.

```
program bifurcacao;
uses crt, graph;
var gd, gm, k, i :  integer;
    b, x :  real;
begin
    gd := detect; gm:=1;
    initgraph(gd,gm,'');
    line(0,0,639,0);
    line(639,0,639,199);
    line(639,199,0,199);
    line(0,199,0,0);
    for k:=1 to 639 do
    begin
    b := 0.7+0.3*k/639;
    x := 0.5;
    for I:=1 to 100 do
        begin
        x := 4 * b * x * (1 - x);
        if i> 50 then
        putpixel(k,round(200-199*x),15);
    end;
end;
repeat until keypressed;
closegraph;
end.
```

3 – Expoentes de Liapunov

O expoente de Liapunov λ mede a taxa de divergência de órbitas que em dado momento são vizinhas. Podemos no caso unidimensional defini-lo a partir da média geométrica das derivadas $A_i = \frac{df_b}{dx}(x_i)$ ao longo dos pontos da órbita $(x_0, x_1, ..., x_k)$ ou mais precisamente a partir do limite

$$\lim_{k\to\infty} \sqrt[k]{A_0 \ldots A_k}\ ;$$

ao logaritmo deste número chama-se expoente de Liapunov, tendo-se portanto:

$$\lambda = \lim_{k\to\infty} \frac{1}{k}\sum_{i=0}^{k} \log \frac{df_b}{dx}(x_i)$$

(ver fig. 3).

```
program liapunov;
uses graph;
var
    gd, gm, k, I : integer;
    b, x, y, z :  real;
procedure liap;
begin
    for k:=1 to 600 do
        begin
        b:=0.8+k/3000;
        x:=0.9; z:=0;
        for I:=1 to 75 do
        begin
        x:=4*b*x*(1-x);
        y:=4*b-8*b*x;
        z:=z+ln(abs(y));
        end;
    z:=z/75;
    putpixel(k,round(100-100*z),7);
    end;
end;
begin
    gd:=detect; gm:=0;
    initgraph(gd,gm,'');
    line(0,0,600,0);
    line(600,0,600,199);
    line(600,199,0,199);
    line(0,199,0,0);
    liap; readln;
    closegraph;
end.
```

Como se vê através deste exemplo o computador encarado como um laboratório de experiências onde se aprende a descobrir "coisas" e a simular todo o tipo de fenómenos. O leitor não precisa de conhecer mais do que o modelo que lhe é apresentado; os programas para o computador são simples como se vê e podem ser modificados e usados pelo leitor mesmo inexperiente em computação. A experimentação, a procura de compreensão, a descoberta de regularidades dentro da complexidade, deve ser encarada como um jogo do qual se tira prazer e inteligibilidade.

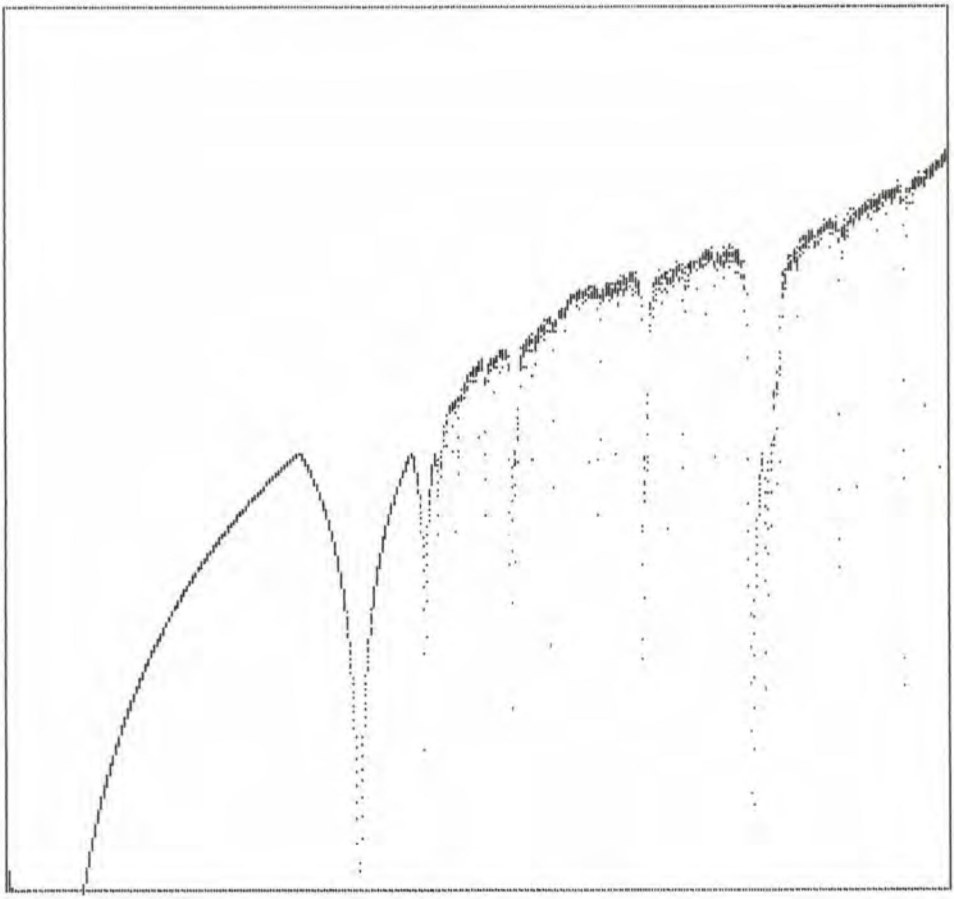

Fig. 3: Expoente de Liapunov.