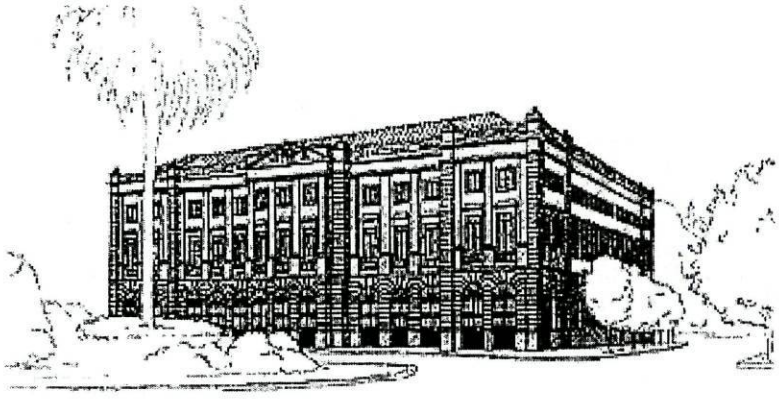
Daniela Alexandra Carvalho de Almeida

O Maple em Sistemas Dinâmicos

uma abordagem para o Ensino Secundário



Departamento de Matemática Pura
Faculdade de Ciências da Universidade do Porto

Daniela Alexandra Carvalho de Almeida

O Maple em Sistemas Dinâmicos

uma abordagem para o Ensino Secundário

Dissertação apresentada à Faculdade de Ciências da Universidade do Porto, em Dezembro de 2003, para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Ensino da Matemática, realizada sob a orientação do Professor Doutor Fernando Jorge Soares Moreira, Professor Auxiliar do Departamento de Matemática Pura da Faculdade de Ciências da Universidade do Porto.

Conteúdo

1 Algumas noções de Maple **19**

1.1 A linguagem do Maple 19

1.2 Variáveis e expressões 23

1.3 Atribuições 24

1.4 Sequências, conjuntos e listas 26

1.5 Funções 27

1.5.1 Definição de funções 27

1.5.2 Composição de funções 28

1.5.3 Derivada e primitiva de uma função 30

1.6 Esboços Gráficos 30

1.7 Animações gráficas 37

1.8 Resolução de Equações 38

1.9 Programação em Maple 42

1.9.1 Comando **if** 42

1.9.2 Comandos de repetição 43

1.9.3 Procedimentos 45

1.10 Tabelas 46

2 Procedimentos Maple para Sistemas Dinâmicos **49**

2.1 Órbita de um ponto 49

2.1.1 Procedimento listaorbita 50

2.1.2 Tabelas 51

2.2 Método gráfico 53
2.3 Comportamento periódico 56
2.3.1 Determinação de pontos fixos 56
2.3.2 Período dois 62
2.3.3 Período três 69
2.3.4 Outros pontos periódicos 73
2.4 Análise estatística de órbitas 75
2.4.1 Diagrama de frequências 76
2.4.2 Funções teste 79
2.5 Codificação de órbitas 81

3 A aplicação tenda **89**
3.1 Esboço gráfico 90
3.2 Análise de órbitas 93
3.2.1 Procedimento listaorbita 93
3.2.2 Recurso às tabelas 94
3.2.3 Análise gráfica 96
3.2.4 Variação com o parâmetro a 97
3.3 Pontos fixos 100
3.3.1 Métodos analíticos 100
3.3.2 Método gráfico 103
3.4 Período dois 108
3.4.1 Métodos analíticos 109
3.4.2 Método gráfico 115
3.5 Período três 119
3.5.1 Métodos analíticos 120
3.5.2 Método gráfico 124
3.6 Período arbitrário 128
3.7 Análise estatística de órbitas 130
3.7.1 Diagrama de frequências 131

3.7.2 Funções teste 134
3.8 Codificação de órbitas 136

4 A aplicação logística **143**
4.1 Esboço gráfico 144
4.2 Análise de órbitas 147
4.2.1 Procedimento listaorbita 147
4.2.2 Recurso às tabelas 149
4.2.3 Análise gráfica 150
4.2.4 Variação com o parâmetro a 151
4.3 Pontos fixos 159
4.3.1 Métodos analíticos 159
4.3.2 Método gráfico 161
4.4 Período dois 163
4.4.1 Métodos analíticos 164
4.4.2 Método gráfico 167
4.5 Período três 170
4.5.1 Métodos analíticos 170
4.5.2 Método gráfico 175
4.6 Período arbitrário 177
4.7 Bifurcação das órbitas periódicas 179
4.8 Constante de Feigenbaum 186
4.9 Análise estatística de órbitas 197
4.9.1 Diagrama de frequências 197
4.9.2 Expoentes de Lyapunov 200
4.10 Codificação 203

5 Maplets **209**
5.1 Pontos fixos de f^n 209
5.2 Plotorbita 211

5.3 Bifurcação 212
5.4 Aplicações bidimensionais 215

Bibliografia **221**

Introdução

Os computadores constituem um dos meios de comunicação, expressão e investigação dos nossos tempos e, cada vez mais, alteram o nosso dia-a-dia assim como a nossa relação com o saber.

Com o passar dos anos, a tecnologia computacional tem vindo a influenciar decisivamente todos os ramos da Ciência e do próprio Ensino. Consequentemente, os conteúdos dos programas de Matemática, e em especial os métodos pelos quais ela é ensinada, têm vindo a mudar. Assim, parece importante incentivar a utilização dos computadores de uma forma cada vez mais generalizada, por todos os professores e também pelos alunos que por eles se venham a interessar, com vista à desmistificação e desdramatização da utilização das novas tecnologias dentro e fora da sala de aula. Além disso, frequentemente, os assuntos tratados nas aulas não despertam grande interesse, para a maioria dos alunos que frequenta a Escola, não pelos assuntos em si, mas pela forma como são apresentados. Este afastamento dos aluno ao que se pretende ensinar leva à procura de novas soluções.

A análise de tais questões permite-nos depreender que a utilização do computador, de maneira "inteligente", pode ser de extrema importância no sentido de promover o pensamento matemático, integrando aspectos geométricos, numéricos e analíticos. Porém, os professores devem saber como e quando usar este potencial da tecnologia: as actividades de programação de computadores na aula de Matemática devem ser utilizadas para apoiar o ensino desta área científica e não devem ser o objectivo do ensino.

Parece ser crucial que o professor tenha ao seu dispor programas de Matemática que lhe permitam transmitir os conteúdos da disciplina de uma forma agradável e divertida. Actualmente, existem à disposição do professor, programas de computação simbólica,

numérica e gráfica que libertam o aluno dos cálculos de manipulação fastidiosa. Desta forma, com as mesmas noções elementares introduzidas nas várias unidades didácticas do Ensino Secundário, consegue-se conduzir o estudo dos alunos para problemas que à priori seriam mais difíceis, mas que as ferramentas computacionais permitem simplificar, na medida em que os cálculos e manipulações complexas são transferidas para o programa. Entre os programas mais utilizados na actualidade estão, sem sombra de dúvida, o Maple, produzido pela *Maplesoft* [12] e o Mathematica produzido pela *Wolfram Research* [11]. Estes programas têm, pelo menos a um nível elementar, ao qual este trabalho se enquadra, características similares, tornando a opção entre um e outro uma mera questão de gosto pessoal e a disponibilidade de acesso a um e a outro.

Neste trabalho, vamos mostrar como em unidades programáticas do Ensino Secundário se consegue, em ambiente Maple, abordar novos e mais complexos exemplos dos conceitos estudados. Veremos explicitamente, como abordar a Unidade Didáctica, (do 12º ano de Escolaridade) "Estudo de Funções", de modo a integrar o estudo da "dinâmica" de aplicações definidas em intervalos de números reais.

Partindo, então de uma função f real de variável real, podemos questionar o aluno sobre o tipo de propriedades que ficam invariantes por f^n para algum valor natural n, ou mesmo aquelas que são assimptoticamente invariantes por f (isto é, aquelas que são invariantes quando n é muito grande). Após o conhecimento das noções de sucessão, limite, continuidade e diferenciabilidade para funções reais de variável real, o aluno pode ser conduzido ao tratamento dessas aplicações do ponto de vista de um "sistema dinâmico".

Ao longo deste trabalho, os sistemas dinâmicos são, por definição, pares (X, f) em que $X \subseteq \mathbb{R}$ e f é uma função real de variável real, sendo X um intervalo na generalidade dos casos. Assim, aqui estudaremos o que habitualmente se designa por sistemas dinâmicos discretos e unidimensionais. Seguindo a nomenclatura tradicional desta área, designaremos por espaço das fases ou de estados iniciais o conjunto X. Para $x_0 \in X$ e $n \in \mathbb{N}$ usamos a notação x_n para designar o valor de $f^n(x_0)$. A *órbita* de x é a sequência $(x_n)_{n\in\mathbb{N}}$. O ponto x diz-se um *ponto periódico* se existir p tal que $f^p(x) = x$ e, nestas condições, ao menor inteiro q tal que $f^q(x) = x$, chama-se *período* de x. No caso de $q = 1$, ou

seja, $f(x) = x$, o ponto x diz-se *ponto fixo* de f. Também usaremos a notação $\omega(x)$, que designaremos por ω-limite de x, para representar o conjunto dos pontos de acumulação da órbita de x.

Neste contexto de sistemas unidimensionais discretos, a existência de pontos fixos, pontos periódicos e características do ω-limite de x para diferentes valores de x no espaço das fases são questões que podem ser colocadas ao aluno de modo que só intervenham os conceitos elementares já seus conhecidos. A determinação de pontos periódicos corresponderá à resolução de equações do tipo $f^p(x) = x$ e a caracterização de ω-limite pode ser posta como uma redefinição de convergência de sucessões.

A possibilidade de experimentação de cálculos e manipulações gráficas quase instantâneas para as funções elementares suas conhecidas (polinomiais e trigonométricas), leva a que o aluno possa abordar estes problemas dinâmicos de um modo interactivo, ultrapassando as dificuldades de cálculo que normalmente são impossíveis de se levar a cabo numa unidade lectiva.

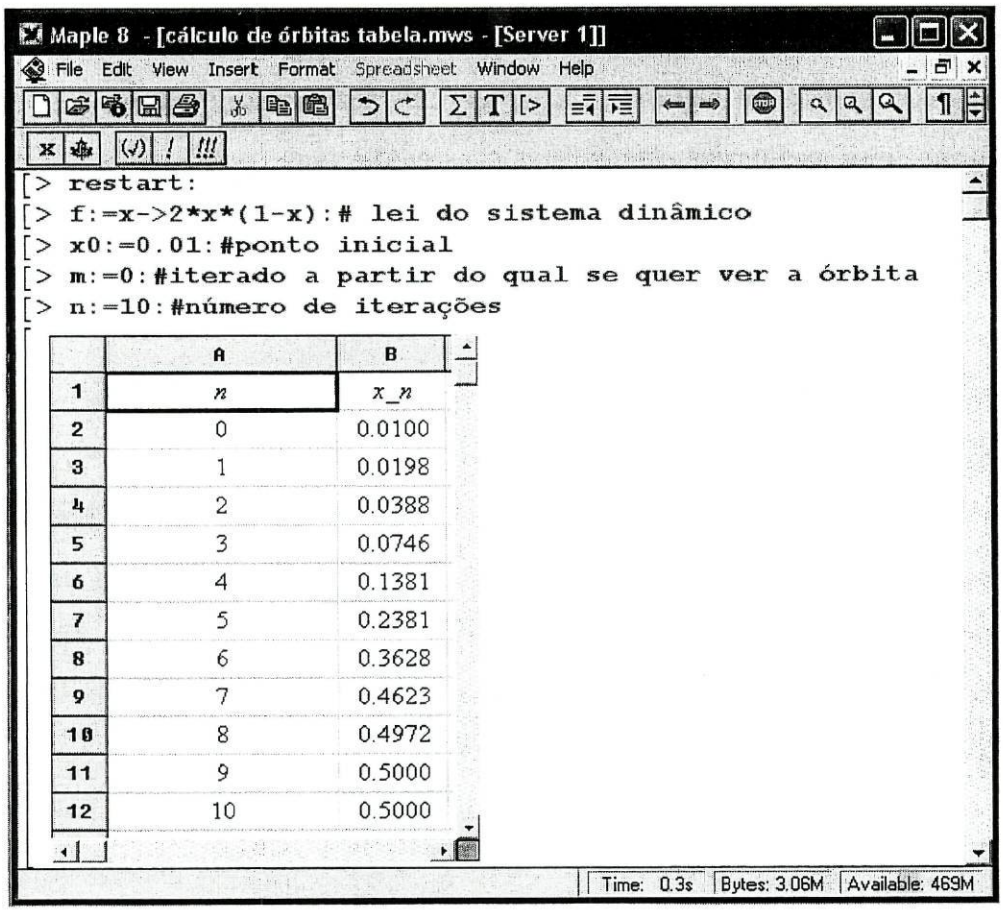
Com base nestas considerações, neste texto, ir-se-á abordar, na óptica do professor que pretende uma ferramenta poderosa, mas de uso simples, o programa Maple como utensílio no estudo dinâmico de funções reais de variável real. O nosso principal objectivo é usar as potencialidades de que o programa é portador para abordar algumas das características dinâmicas mais relevantes das famílias de aplicações tenda e logística. Assumem-se alguns conhecimentos prévios de computação e os conhecimentos de Matemática estão ao nível do 12º ano de escolaridade. Não se faz um tratamento rigoroso, não se fazem provas ou demonstrações de resultados de Sistemas Dinâmicos. A abordagem será mais no sentido de obter, com programas simples em Maple, módulos que nos permitam exprimir e conjecturar acerca dos fenómenos mais ou menos clássicos e já largamente conhecidos destas famílias de aplicações.

Para um acompanhamento deste estudo experimental, seguimos as referências [1] e [2], para as quais os resultados que os módulos evidenciam serão remetidos.

No primeiro capítulo apresentamos, de forma abreviada e na perspectiva da sua utilidade para um professor de Matemática, o ambiente de desenvolvimento do programa

Maple. Neste capítulo, é feita uma apresentação do programa, é dada uma pequena explicação acerca dos principais botões utilizados e são explicadOs, com alguns exemplos práticos, os comandos básicos que consideramos serem indispensáveis a uma boa compreensão do que vai ser tratado nos capítulos seguintes.

O segundo capítulo é todo ele dedicado à elaboração de módulos, chamados de procedimentos e que podem ser utilizados no estudo concreto de uma dada aplicação. Aí, encontraremos procedimentos para analisar a órbita de um ponto inicial, desde um procedimento que nos dá em forma de tabela segmentos de órbita (figura 1),



	A	B
1	n	x_n
2	0	0.0100
3	1	0.0198
4	2	0.0388
5	3	0.0746
6	4	0.1381
7	5	0.2381
8	6	0.3628
9	7	0.4623
10	8	0.4972
11	9	0.5000
12	10	0.5000

Figura 1: órbita de um ponto inicial x_0 em forma de tabela

até um procedimento **plotorbita** que permite obter a clássica representação gráfica da órbita de um ponto x_0 (figura 2).

Ainda no mesmo capítulo, encontraremos procedimentos para determinar pontos periódicos,

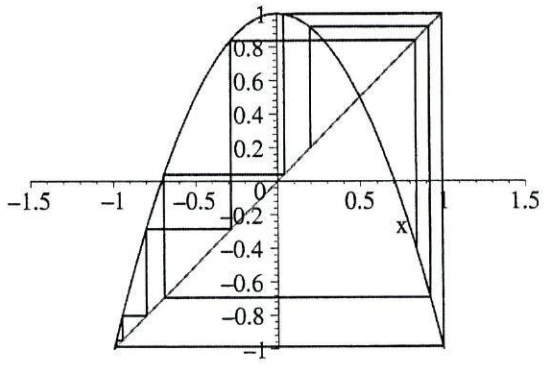


Figura 2: Representação gráfica da órbita obtida com **plotorbita**

que utilizam as potencialidades algébricas e numéricas do Maple na resolução de equações. São também apresentados exemplos que visam alertar o professor para o facto de que os resultados obtidos com procedimentos algébrico-numéricos nem sempre são os inicialmente esperados e que o uso indiscriminado de procedimentos pode conduzir a complicações no ensino das matérias. De seguida, são apresentados os procedimentos de manipulação gráfica que permitem inferir a existência de pontos periódicos. Por exemplo, o procedimento **pernplot** permite inferir a existência de pontos periódicos de período n (para períodos não muito elevados), através da representação gráfica simultânea dos gráficos de f^k, para os divisores k do inteiro n.

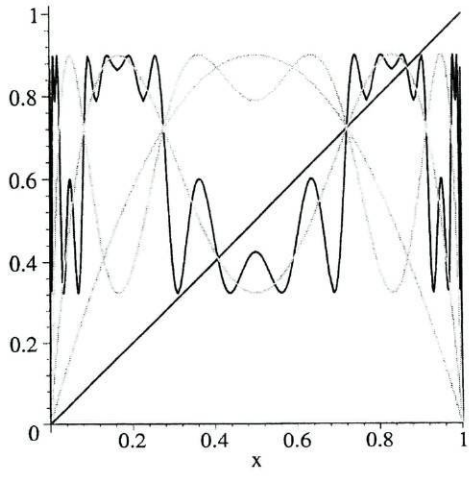


Figura 3: Representação gráfica dos pontos periódicos obtida com **pernplot**

A evidência da existência de órbitas que aparentemente não são periódicas, leva a que se possam introduzir outros elementos na caracterização das mesmas. Um dos procedimentos que se pode revelar mais útil, uma vez que permite, ao professor, relacionar o estudo de funções com a Estatística, é o **diagramafreqs** da secção 2.4.1. Este procedimento serve para identificar numericamente pontos do intervalo, onde se estuda o sistema, que sejam mais significativos na evolução do mesmo. Este procedimento permite dividir o intervalo inicial em N subintervalos de igual amplitude e estudar a frequência de visitas a cada um deles pelos elementos de um conjunto finito de iterados do ponto inicial arbitrário. Com isto, pode-se obter informação acerca da "distribuição" dos elementos de uma órbita pelo espaço de fases. Este procedimento também possibilita relacionar conteúdos e solidificar conhecimentos. Um exemplo ilustrativo desta mais valia é a possibilidade de o aluno relacionar a região do intervalo que corresponde a uma maior afluência da órbita com o conceito de moda dado no capítulo de Probabilidade e Estatística.

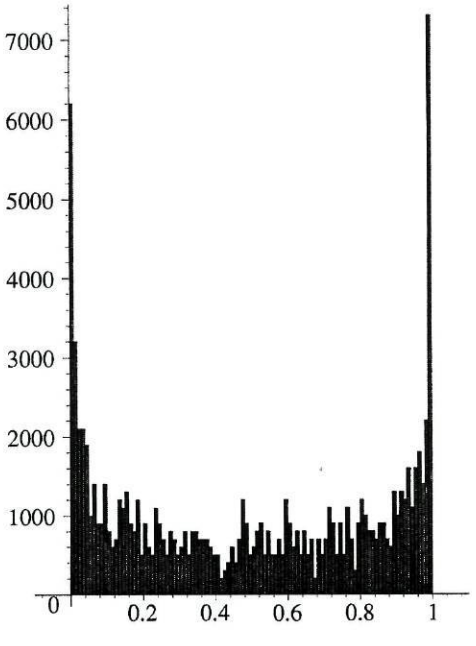


Figura 4: Histograma com obtido com **diagramafreqs**

Outro contributo para o estudo de órbitas não periódicas é o que se faz na secção 2.5, que corresponde à codificação de órbitas e ao estudo dos itinerários correspondentes. Far-

se-á uma partição $\mathcal{P}$ do espaço das fases e associa-se a cada elemento de $\mathcal{P}$ um símbolo. Para um ponto inicial x_0 a órbita determina uma sequência de símbolos, dita de itinerário. A codificação de um sistema tem como finalidade a tentativa de simplificar o sistema e, assim, permitir a obtenção de algumas propriedades de uma maneira mais simples. Nesta secção também fazemos uma análise estatística das órbitas, recorrendo ao procedimento **diagramafreq**. Este procedimento fornece-nos um diagrama da frequência de cada um dos símbolos que ocorre no itinerário de uma órbita.

Nos capítulos 3 e 4 faz-se a aplicação dos procedimentos do capítulo 2 às aplicações designadas do tipo tenda e logística, respectivamente. A família de aplicações do tipo tenda é definida por

$$T_a(x) = \begin{cases} ax & \text{se } 0 \leq x < \frac{1}{2} \\ a(1-x) & \text{se } \frac{1}{2} \leq x \leq 1 \end{cases}$$

e as aplicações da família logística são definidas por

$$f_a(x) = ax(1-x), \quad 0 \leq x \leq 1, \quad 0 \leq a \leq 4.$$

Poder-se-á motivar o aluno para o estudo da família logística através do exemplo clássico que permite modelar o crescimento de populações de colónias de bactérias e referir que a família da aplicação tenda é aquela cujo gráfico é linear por bocados e que mais se assemelha ao gráfico da logística (ver [1]). A escolha destes dois tipos de famílias de aplicações, neste trabalho, deve-se não só ao facto de as fórmulas que as definem serem simples, mas também ao facto de elas já evidenciarem as vantagens e desvantagens da utilização do Maple no estudo de sistemas dinâmicos unidimensionais discretos.

Por fim, o capítulo 5 é reservado à apresentação de uma inovação do Maple - as Maplets. Esta nova tecnologia, presente desde a versão 8 do programa permite criar janelas interactivas com caixas de texto, menus e botões, semelhantes a qualquer outro programa do sistema operativo. Essas janelas interactivas permitem ao utilizador final manipular em Maple sem conhecer o seu código. De facto, as caixas de texto podem conter parâmetros necessários para um dado procedimento e o simples clique num botão

chama (num processo invisível para o utilizador) o Maple e executa o procedimento, apresentando na mesma janela os resultados finais. Assim, o programador conhecendo a linguagem Maple pode construir módulos interactivos para as aulas e assim contribuir para uma participação mais activa do aluno.

Em Sistemas Dinâmicos a visualização gráfica e a interactividade têm um papel muito importante na assimilação, por parte do aluno, das características de um sistema. Assim, esta nova potencialidade do Maple não podia ser, de modo algum, esquecida neste trabalho. Com estas janelas, o aluno para além de visualizar gráficos e/ou animações pode também experimentar alterar os valores das caixas de texto e observar os vários resultados obtidos.

As Maplets apresentadas neste capítulo foram construídas por Avelino Sousa em [10], utilizando aí os procedimentos que se construíram no capítulo 2 deste trabalho. Como mais exemplificativo, temos a Maplet **Bifurcação** que permite esboçar, para um intervalo

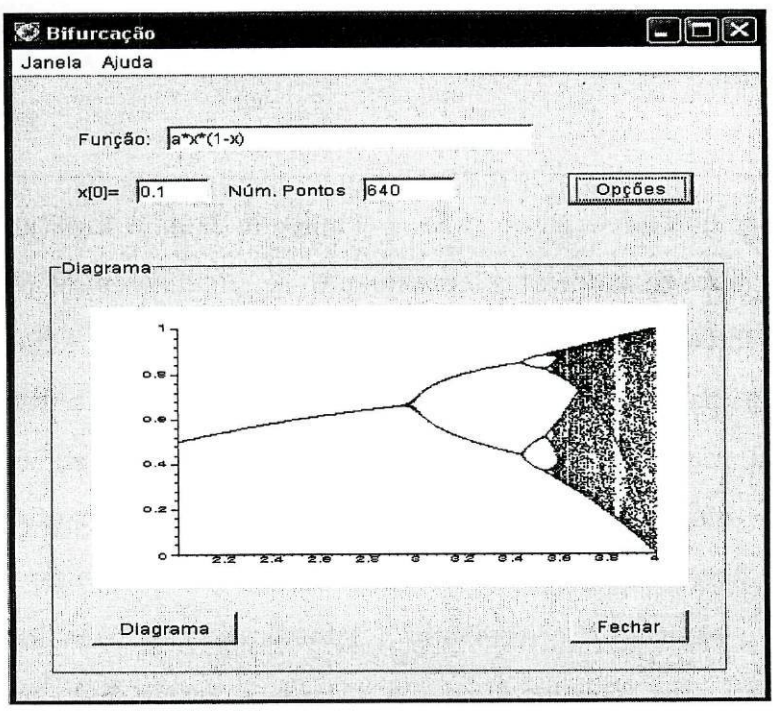


Figura 5: Maplet da bifurcação

$[amin, amax]$, $f_a^k(x_0)$ com k entre m e n. Com esta Maplet podemos obter o "clássico"

diagrama de bifurcação para a aplicação logística, discutido na secção 4.7.

Por fim, na secção 5.4, inserimos neste trabalho Maplets que tratam um sistema dinâmico quando $f : X \to X$ e $X \subseteq \mathbb{R}^2$, os chamados sistemas dinâmicos bi-dimensionais. Aí a Maplet da figura 6 tenta dar a ideia da imagem de $f^n(R)$ em que R define um rectângulo. Através da coloração de faixas do rectângulo R e da visualização de como esta região colorida é transformada, o aluno poderá iniciar a abordagem de como é que diferentes aplicações f actuam em regiões do plano, ao mesmo tempo que lhes são abertas portas para um futuro interesse pela área de sistemas dinâmicos.

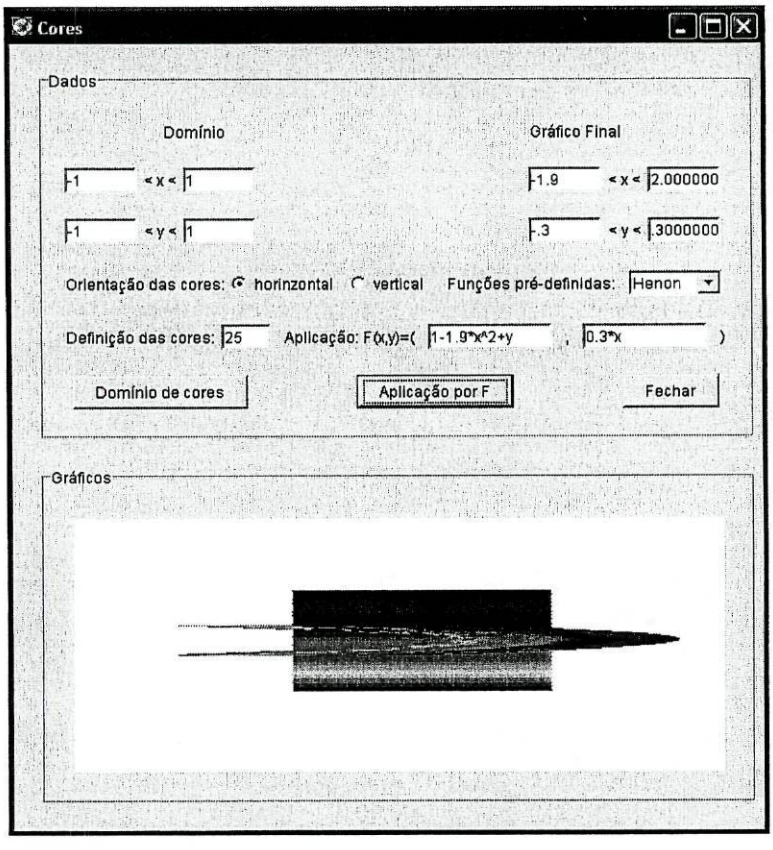


Figura 6: Maplet **Cores**

Capítulo 1

Algumas noções de Maple

Neste capítulo são indicados os comandos Maple básicos que irão ser usados na construção dos procedimentos do capítulo 2.

1.1 A linguagem do Maple

No sistema Windows, a acessibilidade ao Maple é igual à maior parte dos programas. Basta fazer um duplo-clique no símbolo do Maple 8, que por defeito aparece no ambiente de trabalho, ou executá-lo a partir do menu programas. Quando o Maple é iniciado aparece uma janela como a da figura 1.1, que é composta por alguns menus, por botões e pela folha de trabalho, que é o local privilegiado para editar os comandos do Maple. A designação mais usual para folha de trabalho é *worksheet*, pelo que ao longo deste trabalho é natural que seja usado o termo inglês em detrimento de uma tradução para Português.

Como podemos ver pela figura 1.1, na parte superior da janela aparece o menu principal e a barra de ferramentas.

O menu principal corresponde à primeira linha da parte superior da janela e é formado por vários sub-menus: **File**, **Edit**, **View**, **Insert**, **Format**, **Spreadsheet**, **Options**, **Window** e **Help**.

A barra de ferramentas é constituída por duas linhas de botões. A primeira contém botões relacionados com os menus **File**, **Edit**,**View** e **Insert** e a segunda linha contém

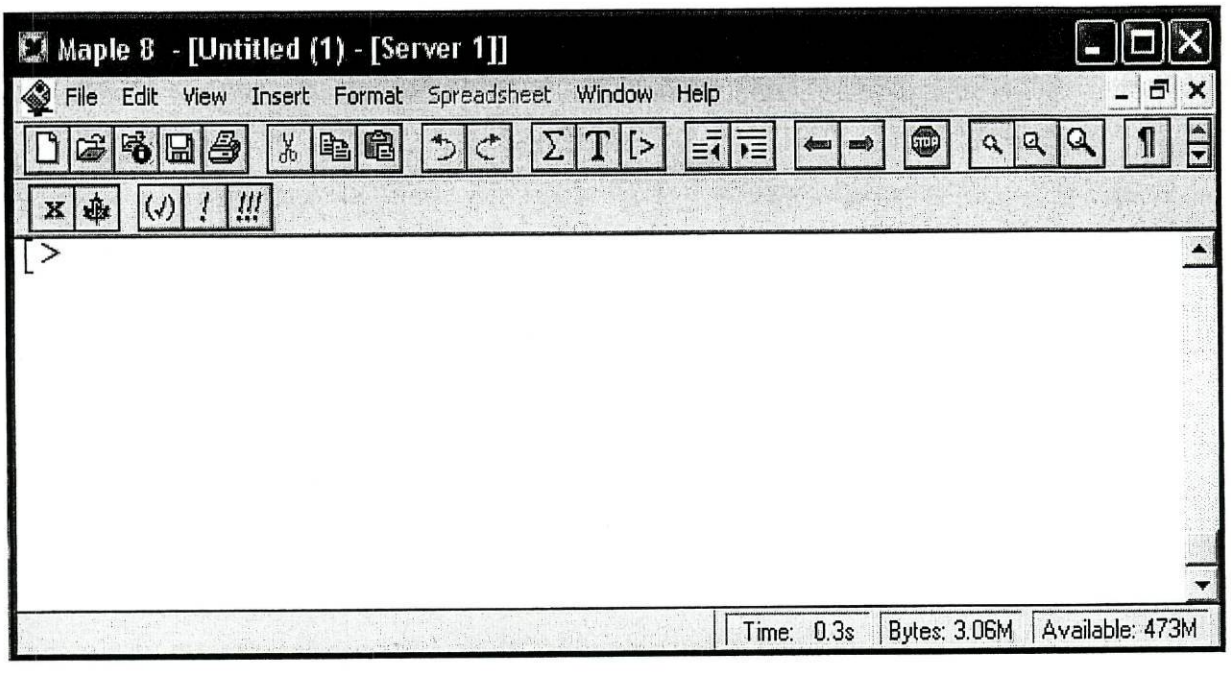


Figura 1.1: Exemplo de uma janela "em branco" do Maple

botões relacionados com comandos de formatação (quando o cursor está sobre texto) ou com comandos de execução (quando o cursor está sobre um bloco de execução).

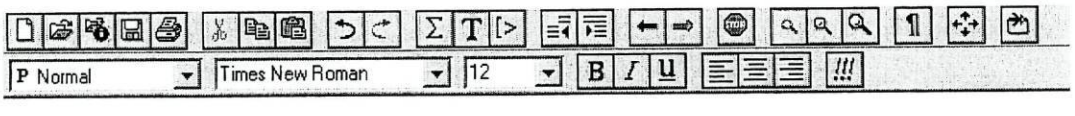


Figura 1.2: Barra de ferramentas quando o cursor está sobre o texto

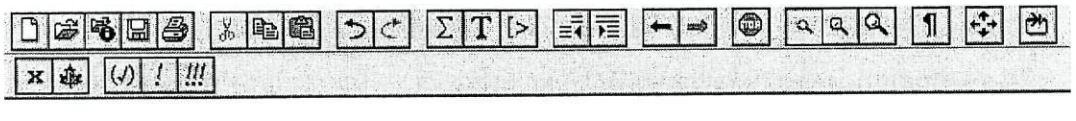
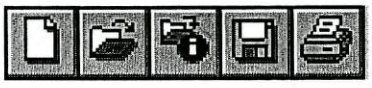
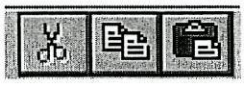
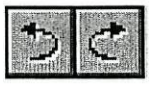


Figura 1.3: Barra de ferramentas quando o cursor está sobre um bloco de execução

De seguida, vamos explicar alguns dos botões mais usados. Os quatro primeiros botões da barra de ferramentas correspondem às operações: folha de trabalho em branco, abrir uma folha já existente, guardar e imprimir a folha actual.



A seguir há um conjunto de três botões que correspondem às operações de cortar, copiar e colar e um outro bloco constituído por dois botões que corresponde a desfazer e refazer a última operação realizada. Os próximos três botões correspondem ao modo de execução do Maple: insere fórmulas matemáticas sem executá-las, insere texto e insere comandos executáveis.


O botão **STOP** é o botão que interrompe o que estiver a ser executado. Por último temos o botão com três pontos de exclamação que serve para executar tudo o que estiver na folha de trabalho.




As linhas de entrada de comando são sempre iniciadas pelo sinal > e, por defeito, são escritas a cor vermelha. Na figura 1.4 exibimos exemplos simples de alguns comandos que é possível utilizar.

A indicação de final de um comando é expressa através do caracter ";" ou ":" seguido da tecla [Enter]. O operador : não exibe os resultados do comando após a execução do mesmo, enquanto que o operador ; imprime os resultados do comando. Ao observar a figura 1.4 podemos ver que ao escrevermos o comando com ; no fim, o resultado é mostrado.

```
> 3*6/2;
```

9

Enquanto que se digitarmos :, o resultado obtido não é mostrado de imediato.

```
> 2+3-5:
```

As linhas de saída dos comandos executados aparecem centradas horizontalmente logo a seguir a uma linha de entrada terminada com ;. Por defeito estas linhas são escritas a

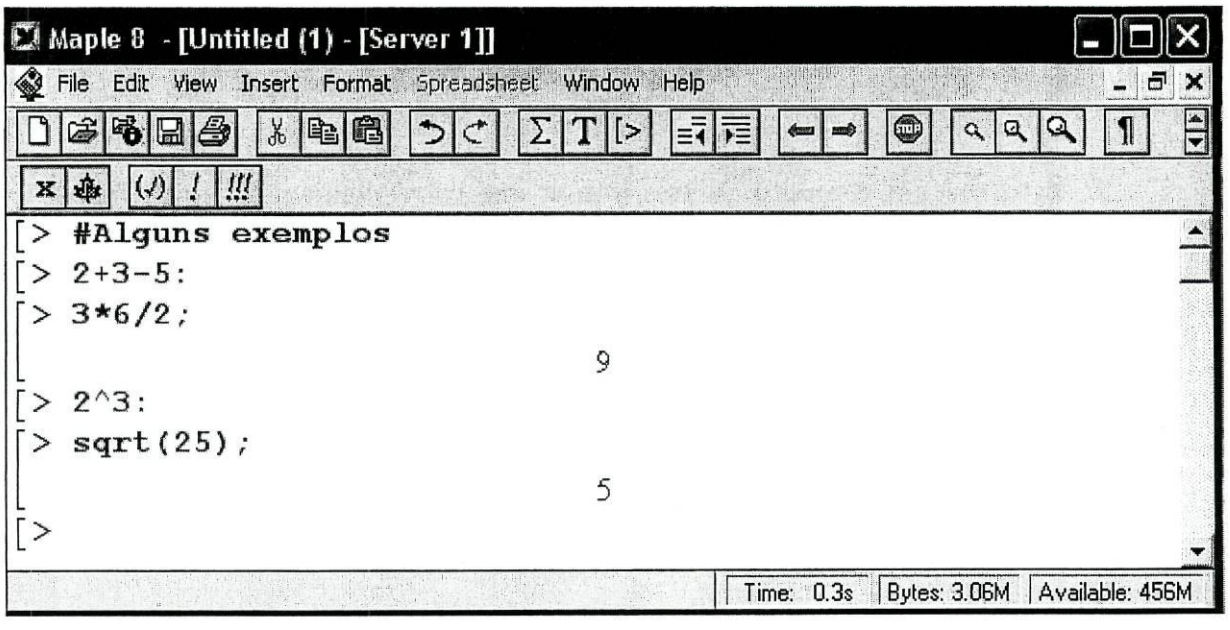


Figura 1.4: Imagem de uma folha de trabalho do Maple com alguns exemplos

azul. Ao longo deste trabalho, também os comandos Maple são transcritos a vermelho e os resultados desses comandos a azul.

As operações aritméticas adição, subtracção, multiplicação, divisão e potenciação são representadas por $+$, $-$, $*$, $/$ e $\hat{}$, respectivamente. Para a raiz quadrada de um número usa-se o comando **sqrt**.

Quando queremos fazer algum comentário usamos o sinal de #. Na linha de comando do Maple, tudo o que for digitado à direita deste símbolo será ignorado na execução do programa, servindo só para orientação do utilizador.

```
> 2+5; #soma de dois com cinco
```

7

Podemos colocar vários comandos na mesma linha. Estes comandos podem ser separados tanto por ; como por :.

```
> 2+3; sqrt(36): 64/2+5;
```

5

37

Os resultados de até três cálculos anteriores podem ser acedidos através dos comandos % (último), %%(penúltimo) e %%% (antepenúltimo). Assim, se quisermos calcular o valor de $\text{sen}^2(\pi/4) + \cos^2(\pi/4)$. Somar 3 ao resultado obtido e calcular a sua raiz quadrada, podemos escrever:

```
> sin(Pi/4);
```

$$\frac{\sqrt{2}}{2}$$

```
> cos(Pi/4);
```

$$\frac{\sqrt{2}}{2}$$

```
> (%)^2+(%%)^2;
```

$$1$$

```
> %+3;
```

$$4$$

```
> sqrt(%);
```

$$2$$

É preciso ter em atenção que se após um cálculo, movermos o cursor para uma posição diferente na folha de trabalho, o símbolo de percentagem não se irá referir ao comando exactamente acima mas sim ao último cálculo executado.

1.2 Variáveis e expressões

Uma variável é um lugar na memória identificado por um nome e serve para "guardar valores". Esse nome pode ser formado por letras minúsculas ou maiúsculas e algarismos. O nome da variável não pode começar com um algarismo. São exemplos de nomes de variáveis válidos: *x*, *y2*, teste, *valor inicial*. São exemplos de nomes inválidos: *2y*, *var:11* e *v-inicial*.

Uma expressão designa qualquer termo que inclua variáveis e operações válidas em Maple. São exemplos de expressões: $\text{sen}\,(x) + \cos(y)$ e x^y.

1.3 Atribuições

Com este programa, podemos atribuir a uma variável números inteiros ou até polinómios e equações. O operador utilizado para fazer a atribuição é :=.

```
> a:=5:
```

```
> b:=x^2+2*x+1:
```

O Maple diferencia letras minúsculas de letras maiúsculas. Por exemplo, "x" é considerado diferente de "X".

```
> x:=5:  X:=3:
```

```
> x+1;
```

$$6$$

```
> X+1;
```

$$4$$

A atribuição de um valor numérico a uma variável pode ser representada internamente de três tipos: inteiro, racional (quociente de inteiros) e do tipo "float" (representação interna de números reais na sua expansão decimal, em que se trunca o número de casas decimais). Um número formado por uma sequência de algarismos sem espaços em branco e sem ponto decimal entre eles é considerado do tipo inteiro. Um número com ponto decimal é considerado do tipo real. Por exemplo, 136 é considerado inteiro, enquanto que 136.00 é considerado real.

É importante observar a diferença entre reais e inteiros porque o programa calcula imediatamente as expressões que contiverem números reais, mas mantém sem calcular algumas expressões que envolvam exclusivamente inteiros. Por exemplo, $\frac{2}{3}$ (divisão de inteiros) é mantido em forma de fracção, mas $\frac{2.0}{3.0}$ é aproximado imediatamente para 0.6666666667, onde ele aparecer.

```
> 2/3;
```

$$\frac{2}{3}$$

```
> 2.0/3.0;
```

$$0.6666666667$$

A presença da variável **Digits** permite estabelecer uma maior ou menor precisão no resultado final. Por defeito, o Maple mostra os resultados com 10 algarismos significativos. Para alterar para outro número, devemos usar o comando **Digits:=n**, em que n representa o número de algarismos significativos. Se no exemplo anterior acrescentarmos a linha:

```
> Digits:=20:
```

Obteremos o resultado com vinte algarismos significativos.

```
> 2.0/3.0;
```

$$0.66666666666666666667$$

Quando o Maple é iniciado é permitido realizar as operações aritméticas básicas, interpretar os comandos inseridos e mostrar resultados. Contudo, o Maple contém comandos que não ficam imediatamente acessíveis ao utilizador. Esses comandos estão agrupados em módulos (*packages*), correspondendo à finalidade a que se destinam. Por exemplo, existe um módulo para Álgebra Linear, outro para Cálculo Vectorial, etc.

Para que um comando de um determinado módulo possa ser utilizado, precisamos fornecer o nome do módulo que o contém. Uma maneira prática de se ter acesso aos comandos de um módulo é carregar esse módulo antes de usar o comando, escrevendo **with(nome do módulo)**. Assim, todos os comandos do módulo ficam disponíveis.

Por exemplo, se quisermos usar comandos gráficos devemos digitar:

```
> with(plots):
```

Um comando muito útil é o de ajuda. Para obter informações específicas sobre algum comando, basta escrever um ponto de interrogação seguido do nome do comando ou pressionar a tecla F1 em cima do nome do comando. Outra maneira de obter ajuda, consiste em seleccionar ajuda com o botão direito do rato, ou ainda, acedendo ao menu do Help do Maple.

Para restaurar a memória do Maple, ou seja, para inicializar o valor de todas as variáveis, funções e pacotes carregados anteriormente, usamos o comando **restart** no início da *worksheet*.

```
> restart:
```

1.4 Sequências, conjuntos e listas

As expressões do tipo *expressão1, expressão2,....* são chamadas de listas. O Maple faz atribuições do tipo *variável:= lista.* Por exemplo, pode-se escrever

```
> X:=2,2,3,5,7; #lista
```

$$X := 2, 2, 3, 5, 7$$

A sequência $Y = (2, 2, 3, 5, 7)$ será representada por $[X]$ e o conjunto $Z = \{2, 3, 5, 7\}$ por $\{X\}$.

```
> Y:=[X]; #Sequência
```

$$Y := [2, 2, 3, 5, 7]$$

```
> Z:={X}; #Conjunto
```

$$Z := \{2, 3, 5, 7\}$$

Na representação de conjuntos no Maple, como habitual, não importa a ordem dos elementos, ao contrário das sequências em que a ordem dos elementos é importante. As listas e as sequências podem ter elementos repetidos. Nos conjuntos, a repetição de elementos é automaticamente eliminada.

Algumas operações entre conjuntos, tais como união, intersecção e diferença podem ser realizadas. Se tivermos, por exemplo, os conjuntos $A = \{1, 2, 3, 4\}$ e $B = \{3, 4, 5, 6\}$, para calcularmos os conjuntos $X = A \cup B$, $Y = A \cap B$ e $Z = A \backslash B$, devemos escrever:

```
> A:={1,2,3,4}:B:={3,4,5,6}:
```

```
> X:=A union B;Y:=A intersect B;Z:=A minus B;
```

e obtemos:

$$X := \{1, 2, 3, 4, 5, 6\}$$

$$Y := \{3, 4\}$$

$$Z := \{1, 2\}$$

O número de elementos de um conjunto ou sequência X é dado por **nops(X)** (do inglês "number of operands"). Para os conjuntos X e Z do exemplo anterior, temos que:

```
> nops(X);
```

6

```
> nops(Z);
```

2

O i-ésimo elemento de uma sequência (ou conjunto) X, que habitualmente denotamos por x_i, é dado por **op(i;X)** ou por **X[i]**. Para a sequência X do exemplo anterior, para determinar o elemento que ocupa a terceira posição escrevemos

```
> op(3,X);
```

e obtemos

3

Em Maple, um intervalo $[a, b]$ é representado por $a..b$. No lugar do valor i na expressão **op(i;X)**, podemos ter um intervalo de valores inteiros de i a j, que representamos por **op(i..j;X)**. Assim, para exibir os elementos da sequência X do exemplo anterior, entre as posições dois e quatro, escrevemos:

```
> op(2..4,X);
```

e obtemos

2, 3, 4

1.5 Funções

1.5.1 Definição de funções

O Maple tem uma grande quantidade de funções pré-definidas. Além disso, também podemos criar as nossas próprias funções. As funções podem ser definidas directamente. Por exemplo, se quisermos definir a função $f(x) = 2x(1 - x)$ devemos escrever:

```
> f:=x->2*x*(1-x);
```

e obtemos

$$f := x \to 2\,x\,(1 - x)$$

Se a função a definir está descrita por ramos, devemos recorrer ao comando **piecewise**.

Por exemplo, para definir $f(x) = \begin{cases} x, & \text{se } x > 0 \\ -x, & \text{se } x < 0 \end{cases}$, escrevemos:

```
> f:=x->piecewise(x<0,-x,x>0,x):
```

Com este comando também podemos usar os operadores lógicos **and** e **or**. Por exemplo, para definir $f(x) = \begin{cases} 2x+1, & \text{se } x^2 > 4 \text{ e } x < 8 \\ x+2, & \text{se } x > 10 \text{ ou } x < -3 \end{cases}$, escrevemos:

```
> f:=x->piecewise(x^2>4 and x<8,2*x+1,x>10 or x<-3,x+2):
```

Para definir funções de várias variáveis procede-se de modo idêntico ao caso de funções com uma só variável. Assim, para $f(x,y) = xy^2 + x - y^2$, digitamos:

```
> f:=(x,y)->x*y^2+x-y^2:
```

Também se pode definir uma função através do comando **unapply** que transforma uma expressão numa função. Se tivermos a expressão $q := x^2 + y^3 + 1$ e quisermos defini-la como uma função em x, podemos fazê-lo do seguinte modo:

```
> q := x^2 + y^3 + 1:
> f := unapply(q,x);
```

$$f := x \to x^2 + y^3 + 1$$

Agora que a função f está definida pode-se determinar a imagem por f de qualquer objecto. Assim, para determinar a imagem por f de 3, basta escrever:

```
> f(3);
```

para alcançar a resposta

$$10 + y^3$$

1.5.2 Composição de funções

A composição de funções é feita com o operador @. $f@g$ representa $f \circ g$. Por exemplo, para determinar $(f \circ g)(x)$, sendo $f(x) = x^2$ e $g(x) = \frac{1}{x}$, devemos escrever

```
> f:=x->x^2:

> g:=x->1/x:

> (f@g)(x);
```

obtemos a resposta

$$\frac{1}{x^2}$$

A composta de f consigo mesma, n vezes, pode ser obtida com $f@@n$. Por exemplo, ao escrevermos:

```
> f:=x->2*x*(1-x):

> (f@@3)(x);
```

a resposta dada é a expressão de $f^3(x)$ com $f(x) = 2x(1-x)$.

$$8\,x\,(1-x)\,(1-2\,x\,(1-x))\,(1-4\,x\,(1-x)\,(1-2\,x\,(1-x)))$$

Para um n elevado (acima de 30, dependendo do grau de complexidade da função) o Maple deixa de responder em tempo útil a este comando. No caso do exemplo anterior, se quisermos calcular $f^{30}(x)$ demorará mais do que quatro minutos [1].

Se quisermos ver o resultado anterior na forma expandida, podemos usar o comando

```
> expand(%);
```

$$8\,x - 56\,x^2 + 224\,x^3 - 560\,x^4 + 896\,x^5 - 896\,x^6 + 512\,x^7 - 128\,x^8$$

[1] Num Pentium IV, 1.5 Ghz

1.5.3 Derivada e primitiva de uma função

A derivada de uma função definida por uma expressão algébrica $f(x)$ na variável x é calculada com o comando **diff(f(x), x)**. Para determinar a derivada de $f(x) = 2x(1-x)$, devemos escrever

```
> f:=x->2*x*(1-x):
> diff(f(x),x);# derivada de f(x) em relação a x
```

e obtemos $f'(x)$.

$$2 - 4\,x$$

Pode-se obter uma primitiva de uma função, definida por uma expressão algébrica $f(x)$ na variável x, através do comando **int(f(x), x)**. Por exemplo, para determinar a primitiva de $f(x) = 4x^2 - 3x + 2$, escrevemos

```
> f:=x->4*x^2-3*x+2;
> int(f(x),x);
```

Como resposta obtemos uma primitiva de f.

$$\frac{4}{3}\,x^3 - \frac{3}{2}\,x^2 + 2\,x$$

1.6 Esboços Gráficos

Para as funções polinomiais e as que contêm funções trigonométricas, exponenciais, logarítmicas que são estudadas no Secundário, a resposta do Maple, a comandos gráficos, é quase instantânea. O Maple possui vários comandos e muitas opções para manipulação gráfica. Para desenharmos o gráfico de $f(x) = x^2$ com x a variar no intervalo $[-3, 3]$, escrevemos:

```
> with(plots):
> plot(x^2,x=-3..3);
```

Apesar de neste caso não ser necessário escrever a instrução, **with(plots)** (pois o gráfico pretendido é bastante simples), esta instrução faz com que o Maple reaja rapidamente ao comando de esboço gráfico.

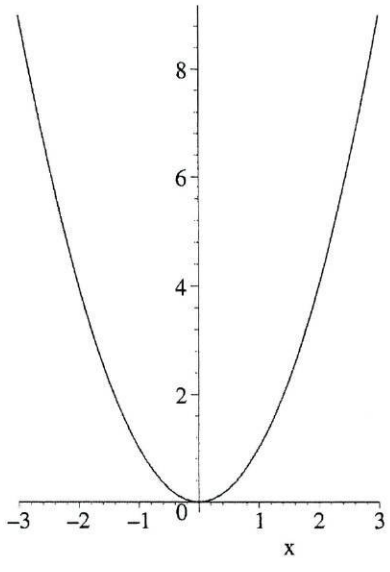


Figura 1.5: Gráfico da função $f(x) = x^2$

O comando **plot** pode ter a seguinte estrutura:

plot($f(x)$, x=a..b, opção1, opção2,...)

Existem várias opções, que devem ser separadas por vírgulas e das quais destacamos:

- A amplitude dos eixos: podemos definir entre que valores queremos ver o eixo dos xx e dos yy. Se quisermos ver o eixo dos xx entre -2 e 2 e o eixo dos yy entre 1,8 e 5 devemos escrever:

```
> plot(f(x),x=-2..2,view=[-2..2,1.8..5]);
```

- A cor utilizada para desenhar o gráfico: para que o gráfico da função fique, por exemplo, a vermelho devemos escrever:

```
> plot(f(x),x=-2..2,color=red);
```

- A fonte utilizada para o texto que aparece no gráfico: para que texto apareça escrito, por exemplo, no tamanho 14, no estilo *Roman* e na família *Times*, devemos escrever

```
> plot(f(x),x=-2..2,font=[TIMES,ROMAN,14]);
```

- A escala utilizada nos eixos: por defeito a escala usada é a que melhor aproveita as dimensões da janela do Maple. Se quisermos a mesma escala em ambos os eixos xx e yy:

```
> plot(f(x),x=-2..2,scaling=constrained);
```

Como já referimos existem outras opções que podem ser consultadas no menu Ajuda do programa.

Para obter o gráfico da figura 1.6, com todas as opções atrás referidas devemos escrever:

```
> with(plots):

> f:=x->x^2+2:
> plot(f(x),x=-2..2,color=red,view=[-2..2,1.8..5],font=[TIMES,ROMAN,12]
  ,scaling=constrained);
```

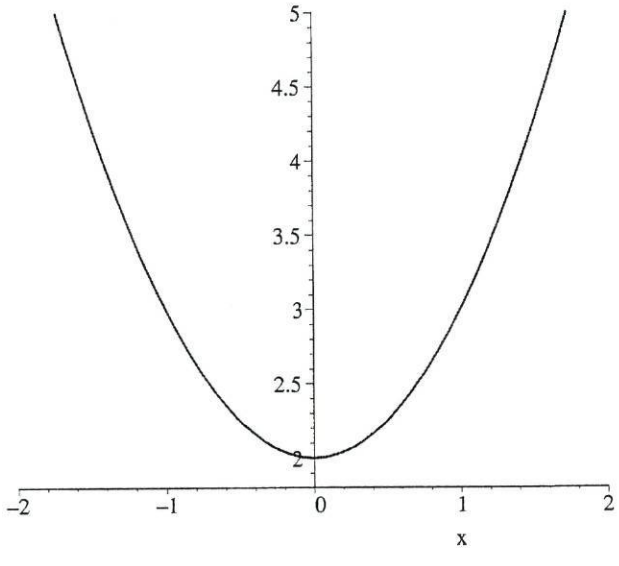


Figura 1.6: Gráfico de $f(x) = x^2 + 2$ exibindo as várias opções do comando **plot**

Há a hipótese de construir vários gráficos num mesmo sistema de eixos. A maneira mais simples de o fazer é fornecendo uma lista de funções $[f_1, f_2, ...]$, que será usada num só comando **plot**. Nesses casos as funções devem ter um domínio comum e as outras opções, tal como as funções, também podem ser dadas na forma de lista (opção=[opção1,opção2,...]).

Se quisermos representar no mesmo sistema de eixos, por exemplo, os gráficos das funções *seno* e *cosseno* no intervalo $[-\pi, \pi]$, em que o gráfico da função *seno* é construído com um traçado fino (*thickness = 0*) e a cor azul (*color = blue*), e o gráfico da função *cosseno* é construído com um traçado grosso (*thickness = 3*) e a cor vermelha (*color = red*), devemos escrever:

```
> with(plots):
> plot([sin(t),cos(t)],t=-Pi..Pi,thickness=[0,3],color=[blue,red]);
```

Com as instruções acima referidas, obtemos o gráfico da figura 1.7.

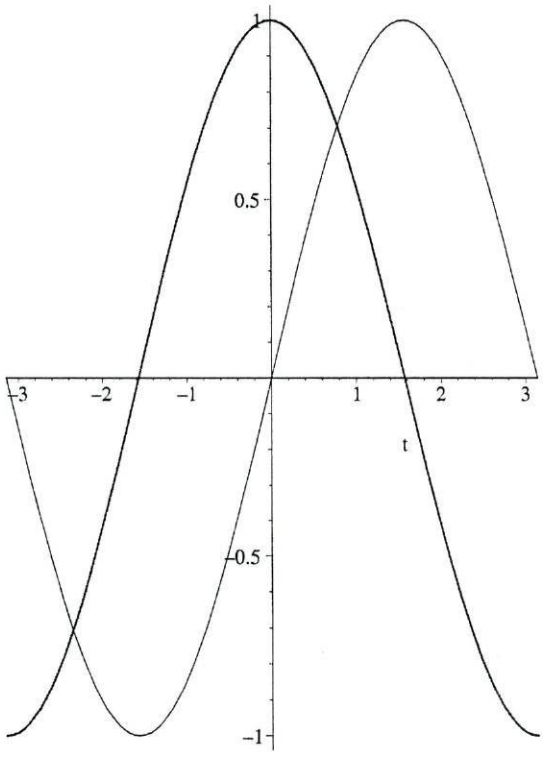


Figura 1.7: Gráfico de duas funções representadas no mesmo referencial

Existe outra possibilidade para obter vários gráficos simultaneamente: construindo cada um deles separadamente e mostrando-os usando o comando **display** que faz parte da package **with(plots)**. Para que os pontos obtidos sejam guardados numa variável, cada gráfico é obtido com comandos do tipo

variável:=**plot**(...):

É conveniente terminar com **:** no final desse comando para que a extensa relação dos pontos do gráfico não seja mostrada. No final, é só usar o comando

display(variável1, variável2, ...)

para que os gráficos apareçam todos no mesmo referencial.

Suponhamos que queríamos mostrar no mesmo sistema de eixos os gráficos das funções $f(x) = x$, $g(x) = ln(x)$ e $h(x) = x^2$. Para tal, vamos construir os gráficos destas três funções separadamente e no fim usar o comando **display**.

```
> with(plots):
> f:=plot(x,x=-4..4,y=-4..4,scaling=constrained,color=red,thickness=2):
  g:=plot(ln(x),x=0..4,y=-4..4,scaling=constrained,color=blue,thickness=2):
  h:=plot(x^2,x=-4..4,y=-4..4,scaling=constrained,color=black,thickness=0):
```

Ao usarmos o comando

```
> display(f,g,h);
```

Obtemos o gráfico da figura 1.8.

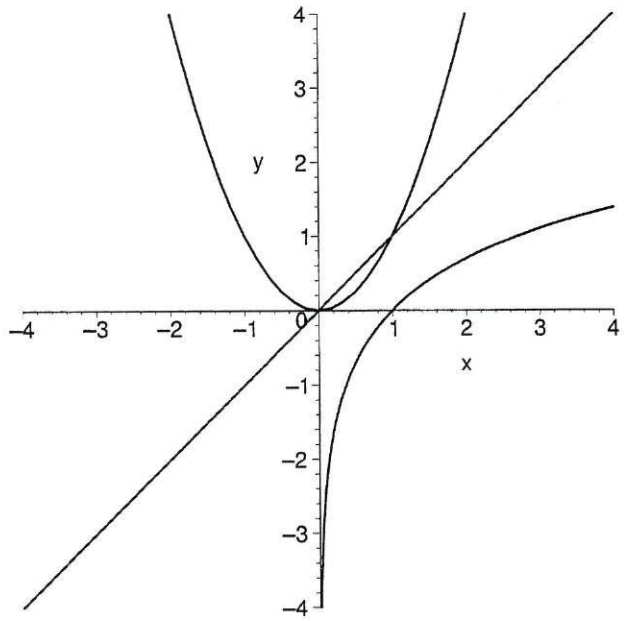


Figura 1.8: Representação gráfica obtida com o comando **display**

Também é possível esboçar o gráfico de funções com duas ou mais variáveis. Uma função de duas variáveis reais x e y pode ser construído com o comando **plot3d**, que tem a seguinte sintaxe:

plot3d, (x=a..b, y=c..d, opção1, opção2, ...)

Se quisermos, por exemplo, construir o gráfico do parabolóide $z = -x^2 - y^2$ e definir o intervalo de variação do x e do y como sendo o intervalo $[-2; 2]$, devemos escrever o seguinte:

```
> with(plots):
> plot3d(-x^2- y^2,x=-2..2,y=-2..2);
```

Ao escrevermos estas instruções obtemos o gráfico da figura 1.9.

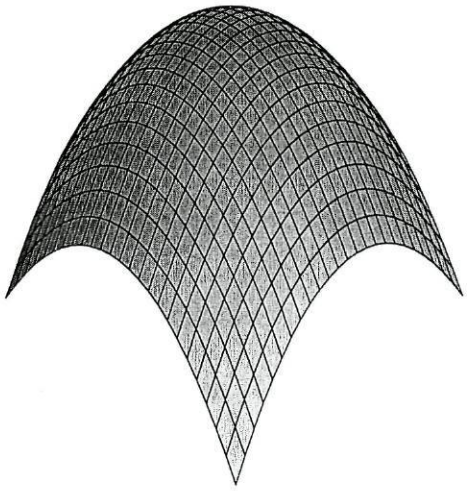


Figura 1.9: Gráfico do parabolóide $z = -x^2 - y^2$

Após a construção do gráfico, ele pode ser rodado com o auxílio do rato: basta dar um simples clique com o botão esquerdo e, mantendo-o preso, rodar o gráfico para uma nova posição. Pode-se também modificar o gráfico construído com os botões que aparecem no écran (abaixo da barra de ferramentas). As opções deste comando podem ser consultadas no menu de Ajuda.

Há uma potencialidade do Maple que é o comando **pointplot** que é usado para criar um esboço bidimensional de pontos. Os pontos em questão têm que estar em forma de conjunto ou de lista. Suponhamos que temos o conjunto de pontos $A = \{(1, 0.5)(-1, 1)(1, -1)(1, 1)\}$ e queremos marcar estes pontos num sistema de eixos. Para isso, escrevemos:

```
> with(plots):
```

```
> A:={[1,0.5],[-1,1],[1,-1],[1,1]}:
> pointplot(A);
```

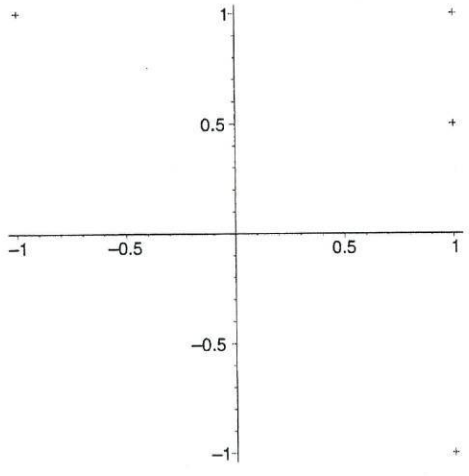


Figura 1.10: Representação bidimensional de pontos

Se os quisermos unir por segmentos de recta, devemos usar a opção ***connect=true***, que faz com que os pontos sejam unidos pela ordem que são inseridos.

```
> with(plots):
> A:={[1,0.5],[-1,1],[1,-1],[1,1]}:
> pointplot(A,connect=true);
```

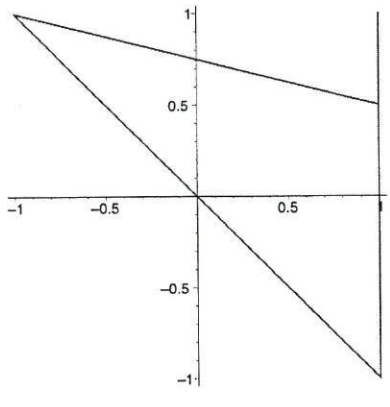


Figura 1.11: Representação de segmentos que unem pontos de uma lista

As opções deste comando são semelhantes às do comando **plot**.

Existe um outro comando no Maple, o comando **PLOT(CURVES)**, que serve para definir uma sequência de curvas que une os pontos de uma determinada lista e esboçar essa sequência. A diferença entre este comando e o anterior é que este desenha de imediato os segmentos, sem ser necessário usar a opção ***connect=true***.

Suponhamos que queríamos desenhar um quadrado de vértices $(0,0),(1,0),(1,1)$ e $(0,1)$. Assim, devemos escrever:

```
> PLOT(CURVES([[0,0],[0,1],[1,1],[1,0],[0,0]]));
```

Para obter o gráfico da figura 1.12.

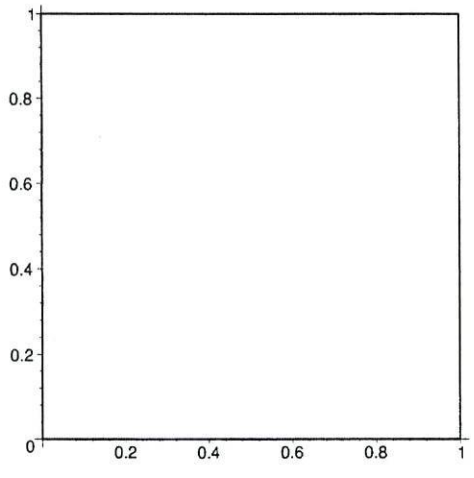


Figura 1.12: Quadrado obtido com o comando **PLOT(CURVES)**

1.7 Animações gráficas

Uma animação consiste numa sequência de gráficos, mostrados consecutivamente, um após o outro, o que gera uma ilusão de movimento pois parece que os gráficos se movem no écran.

A sintaxe do **animate** é

$$\textbf{animate}(f(x,t),\ x = a..b,\ t = c..d,\ \text{opções})$$

onde $f(x,t)$ é uma função de duas variáveis, que consideramos aqui como sendo a expressão de uma família de funções $f_t(x)$, onde x é uma variável e t é um parâmetro (real).

A expressão $f(x,t)$ deve ser seguida pela variação do x e do t. A variação do $x = a..b$ corresponde ao domínio das funções f_t envolvidas na animação, enquanto que a variação do $t = c..d$ corresponde às posições intermédias. O valor $t = c$ corresponde ao gráfico inicial e $t = d$ corresponde ao gráfico final. O número total de gráficos construídos é controlado com a opção frames = n. Por defeito, são apresentados 16 gráficos. Os parâmetros do comando **animate** são a função, o intervalo do domínio, o intervalo de valores do parâmetro. Por exemplo, se escrevermos

```
> animate(sin(2*Pi*(x+t)),x=0..1,t=0..11/12);
```

surge uma animação do gráfico de $f_t(x) = sin(2\pi(x+t))$, para o domínio $[0;1]$, com t no intervalo $[0;\frac{11}{12}]$. Após a execução do comando de animação, aparecerá um gráfico estático. Para animá-lo, devemos clicar com o botão direito do rato sobre o gráfico e utilizar a opção *play* do menu *animation*. Existem outras opções tais como: *next* (para ver o próximo gráfico da animação), *faster* ou *slower* (para configurar a velocidade da animação em rápida ou lenta) e *continuous* ou *syngle circle* (para tornar a animação contínua ou não).

1.8 Resolução de Equações

Uma equação é identificada pelo sinal de igual entre duas expressões. O comando para a resolução exacta de uma equação é o **solve(equac,x)**. A resposta fornecida por este comando é uma sequência das raízes encontradas para a equação **equac** em ordem à variável **x**.

Se quisermos resolver a equação $4x - 1 = 7x + 9$, em ordem a x, basta escrever:

```
> solve(4*x-1=7*x+9,x);
```

para obtermos a solução:

$$\frac{-10}{3}$$

Analogamente, para equações polinomiais do 2° grau, a resposta é quase imediata, mesmo quando as raízes são imaginárias.

```
> solve(x^2+100=0,x);
```

$$10\,I,\ -10\,I$$

Note-se que este comando não resolve todos os problemas. O Maple segue um algoritmo e portanto não tem a habilidade de usar atalhos que podem ser usados quando se fazem os cálculos manualmente. Em geral, para equações polinomiais de grau superior a quatro não é possível exprimir explicitamente as suas soluções. Nestes casos, as soluções são dadas implicitamente através da expressão **RootOf**, que indica que a equação tem raízes, mas não consegue exprimi-las por funções elementares em relação aos coeficientes introduzidos. Se quisermos, por exemplo, resolver a equação $x = \cos(x)$, com o comando **solve**, não obteremos um valor algébrico para a solução. De facto, ao escrevermos o comando:

```
> solve(cos(x)=x,x);
```

o que nos apresenta é:

$$\text{RootOf}(_Z - \cos(_Z))$$

Por vezes, para funções não polinomiais, ele consegue reconhecer se os resultados obtidos são reais ou não. Com o comando **type**(expressão, **realcons**) é possível determinar se essa raiz é real. No caso do exemplo anterior, podemos escrever:

```
> solve(cos(x)=x,x):
> type(%, realcons);
```

Obtemos a resposta

$$true$$

que, neste caso, indica que a solução da equação $\cos(x) = x$ é real. Caso não o fosse, o resultado do comando **type**(expressão, **realcons**) seria **false**.

O comando **solve** permite resolver equações que têm raízes não reais. Por exemplo, as duas raízes da equação $x^2 + \frac{1}{2} = x$ são imaginárias e de facto, conseguimos obtê-las, no Maple, com o comando **solve**.

```
> solve(x^2+1/2=x,x);
```

$$\frac{1}{2} + \frac{1}{2}I,\ \frac{1}{2} - \frac{1}{2}I$$

Quando algum dos coeficientes da equação é do tipo **float**, o Maple reage como se

tratasse de valores aproximados, dando um resultado numérico. Se na equação anterior, em vez de escrevermos $\frac{1}{2}$, escrevermos 0.5, o resultado obtido será diferente.

```
> solve(x^2+0.5=x,x);
```

$$0.5000000000 + 0.5000000000\,I,\ 0.5000000000 - 0.5000000000\,I$$

Para obter soluções reais aproximadas de uma equação, o Maple possui o comando **fsolve(equac,x)**. A resposta fornecida por este comando é uma sequência das raízes encontradas para a equação **equac** em ordem à variável **x**.

Para determinar numericamente as raízes da equação $3x^4 - 16x^3 - 3x^2 + 13x + 16 = 0$, podemos escrever:

```
> fsolve(3*x^4-16*x^3-3*x^2+13*x+16=0,x);
```

e obtemos as raízes

$$1.324717957,\ 5.333333333$$

Com uma pequena alteração na sintaxe do comando, podemos indicar o intervalo no qual queremos procurar a raiz da equação. Assim, o comando **fsolve(equac,x,a..b)** dá-nos a sequência das raízes da equação **equac** em ordem à variável **x**, no intervalo **a..b**.

Para o exemplo anterior, se quiséssemos só a solução que está no intervalo $[3, 6]$, devíamos escrever o seguinte:

```
> fsolve(3*x^4-16*x^3-3*x^2+13*x+16=0,x=3..6);
```

e obtemos a raíz

$$5.333333333$$

O comando **fsolve** apresenta um valor numérico aproximado para as soluções de uma equação. Por esse facto, quando uma equação tem raízes imaginárias e reais, só estas últimas é que são apresentadas. Por exemplo, a equação $1 + 1.5x^3 = x$ tem duas soluções reais e uma imaginária e ao resolver a equação com o comando **fsolve** só é apresentada a raiz real.

```
> fsolve(1+1.5*x^3=x,x);
```

$$-1.122706479$$

Quando a equação não tem soluções reais, não é apresentado qualquer resultado. Por

exemplo, a equação $1 + 0.5x^2 = x$ tem duas raízes no corpo dos complexos e ao resolver a equação com o comando **fsolve** não é mostrada nenhuma delas.

```
> fsolve(1+0.5*x^2=x,x);
```

Para que sejam mostradas todas as soluções em $\mathbb{C}$, devemos usar a opção **complex**. Assim, no caso do exemplo anterior, da equação $1+0.5x^2 = x$, devemos escrever o seguinte:

```
> f:=x->1+1.5*x^2:

> fsolve(f(x)=x,x,complex);
```

e obtemos todas as raízes da equação.

$$0.3333333333 - 0.7453559925\,I,\ 0.3333333333 + 0.7453559925\,I$$

Esta opção pode ser usada mesmo quando a equação só tem raízes reais. A equação $1 - 0.5x^2 = x$ só tem raízes reais e o uso desta opção não interfere em nada com o resultado apresentado.

```
> fsolve(1-0.5*x^2=x,x,complex);
```

$$-2.732050808,\ 0.7320508076$$

Em geral, para funções polinomiais o Maple apresenta todas as soluções reais, nos restantes casos dá como resposta só uma das soluções. Como exemplo, temos a equação $\sin(x) - x^2 = 0$ que tem duas soluções no intervalo $[0, 1]$ e com o comando **fsolve**, só obtemos uma.

```
> fsolve(sin(x)- x^2=0,x);
```

$$0.$$

Ao esboçarmos o gráfico de sen (x) e de x^2 no mesmo sistema de eixos (figura 1.13), observamos que os dois gráficos se intersectam em dois pontos distintos.

A opção **avoid** do comando **fsolve** permite verificar a existência de outras soluções para além das já conhecidas. No nosso exemplo, podemos escrever

```
> fsolve(sin(x)- x^2=0,x,avoid={x=0});
```

e assim obtemos a outra solução da equação.

$$0.8767262154$$

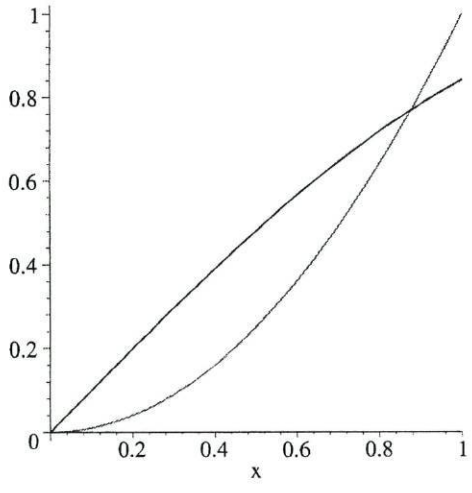


Figura 1.13: Representação gráfica das funções $sin(x)$ e x^2

1.9 Programação em Maple

1.9.1 Comando if

Ao tentar codificar determinado algoritmo, muitas vezes precisamos de definir que determinado grupo de comandos só será executado sob determinadas condições. Para isso, o Maple possui o comando **if** cuja forma mais simplificada de usar é:

if condição **then** comando **end if**

onde a condição é uma expressão lógica que deve ser avaliada de verdadeira ou falsa e o comando deve ser escrito depois da palavra **then** e antes de **end if** e só será executado se a condição for verdadeira.

Uma forma mais completa da declaração do comando **if** é:

if condição **then** comando1 **else** comando2 **end if**

Neste caso, se a condição for verdadeira será executado o comando1; caso contrário, será executado o comando2. No lugar de comando1 ou comando2 podemos ter vários comandos.

No exemplo seguinte temos que $y = x$ se $x \geq 0$ e $y = -x$ se $x < 0$.

```
> if x >= 0 then y:=x else y:=-x end if:
```

Para testar o programa com, por exemplo $x = 6$, devemos escrever:

```
> x:=6:

> if x >= 0 then y:=x else y:=-x end if;
```

e obtemos a resposta

$$y := 6$$

1.9.2 Comandos de repetição

A repetição na execução de comandos é muito utilizada em quase toda a computação. Para o efeito, o Maple possui os comandos **for** e **while**.

- **for**

  A forma mais utilizada do comando **for** é

  for variável **from** valor1 to valor2 **do** comando **end do**

  e funciona da seguinte maneira: o comando deve ser escrito depois do **do** e antes do **end do** para ser executado para cada valor que a variável assumir no intervalo [valor1; valor2]. A variável tem valor inicial valor1 e vai sendo aumentada sucessivamente de 1 em 1 e, enquanto o seu valor for menor ou igual a valor2, o comando vai sendo executado. Assim que a variável assumir um valor maior do que valor2, o ciclo **for** termina.

  Para determinar a soma de todos os números inteiros entre 1 e 100, podemos iniciar com o valor 0 para uma variável *soma* e, para cada valor de n entre 1 e 100, adicionar o valor de n à variável *soma*.

```
> soma:= 0:

  for n from 1 to 100 do

  soma:=soma+n:

  end do;
```

  O comando **for** admite que seja definido um incremento para a variável:

for variável **from** valor1 to valor2 **by** incremento **do** comando **end do**

O valor do incremento fornecido é usado para ir alterando o valor da variável, até que ela ultrapasse o valor2. A sequência de valores da variável é crescente se o incremento for positivo e é decrescente se ele for negativo.

Para calcular a soma de todos os números ímpares menores do que 100, podemos iniciar com o valor 0 para uma variável *soma*, aumentar o n de 2 em 2 enquanto for menor ou igual a 100 e adicionar o valor de n à variável *soma*.

```
> soma:=0:
> for n from 1 to 100 by 2 do

  soma:=soma+n:

  end do;
```

- **while**

A forma mais utilizada deste comando de repetição é:

while condição **do** comando **end do**.

A condição é avaliada. Se for verdadeira, o comando escrito após o **do** e antes do **end do** é executado. Se a condição for falsa, o comando será ignorado. Sempre que o comando for executado, a condição será avaliada e, enquanto ela for verdadeira, o comando será executado novamente. O comando deve ter instruções que possam em algum momento modificar o valor da condição, senão o ciclo **while** repetir-se-á indefinidamente. A vantagem de usar este comando é que não temos de saber quantas vezes é que o ciclo será repetido.

Se quisermos determinar os quadrados dos cinco primeiros números naturais, podemos escrever:

```
> i:=1:
> while i<6 do print(i^2);

  i:=i+1:

  end do:
```

1.9.3 Procedimentos

Um procedimento não é mais do que uma função que depende de vários parâmetros e que como output dá um cálculo, um gráfico, etc. Para chamarmos um procedimento é preciso escrever o nome do procedimento seguido de parênteses.

Como em toda a linguagem de programação, existem dois tipos de variáveis: as locais e as globais. Como o nome já diz tudo, a diferença é só na região do programa onde elas são conhecidas. As globais são conhecidas em toda parte e as locais são restritas aos procedimentos ou às funções onde elas estiverem definidas. No Maple, uma variável x é definida como sendo local, se for declarada na forma **local x**; no início do procedimento. Analogamente, x será global se for definida na forma **global x**. Se num procedimento, não for especificado nada a respeito de uma variável, o Maple tentará defini-la como sendo local.

Um procedimento deve obedecer à seguinte sintaxe:

```
> nome:=proc(parâmetro1,parâmetro2,...)
  declaração de variáveis locais;
  bloco de comandos;
  end proc:
```

Com o procedimento que a seguir apresentamos obtemos a soma de dois números.

```
> soma:=proc(a,b)
  local c; # variável usada para guardar a soma
  c:=a+b:
  end proc:
```

Ao executarmos o procedimento para os números 6 e 3:

```
> soma(6,3);
```

obtemos:

1.10 Tabelas

O Maple tem a possibilidade de criar tabelas com folhas de cálculo similares às do Excel, mas efectuando também cálculos algébricos. Para que uma tabela apareça numa folha de trabalho do Maple temos de ir ao menu **Insert** e escolher a opção **Spreadsheet**. Obtendo-se a tabela da figura 1.14.

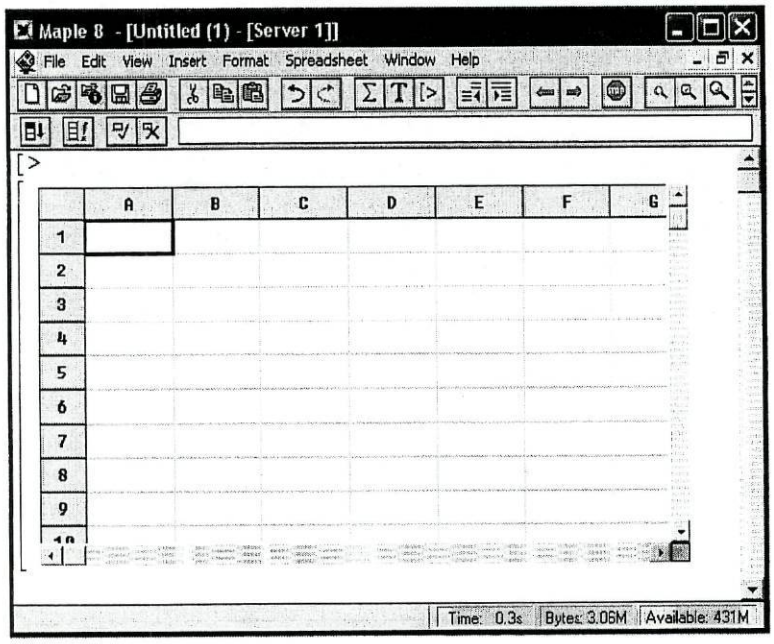


Figura 1.14: Janela do Maple com uma tabela

Se quisermos construir uma tabela de duas colunas, em que na primeira coluna estejam os primeiros dez números naturais e na segunda os respectivos quadrados desses números, procedemos como a seguir se indica.

(i) Na célula A1 escrevemos n;

(ii) Na célula B1 escrevemos n^2;

(iii) Na célula A2 escrevemos 1;

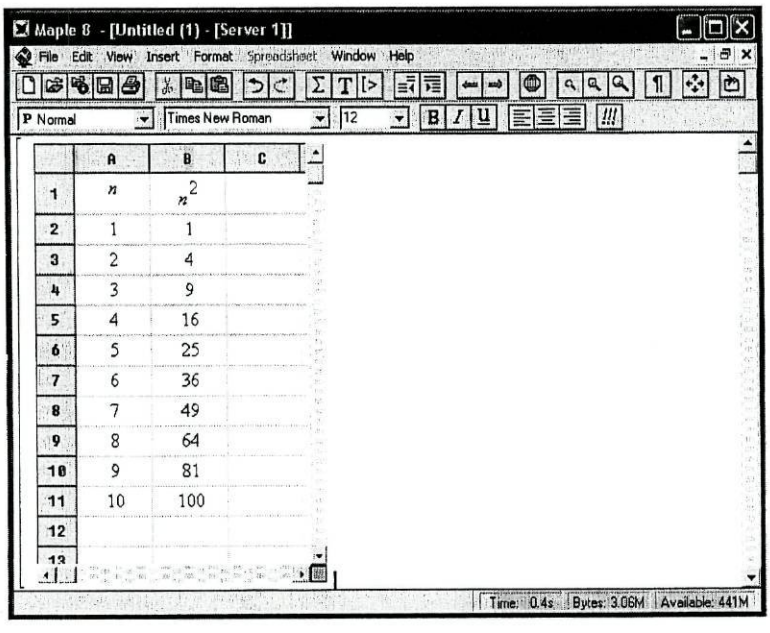
(iv) Para que apareçam automaticamente os números de 1 a 10 na primeira coluna, seleccionamos a célula A2 e algumas mais para baixo. Vamos ao menu **Spreadsheet**,

escolhemos a opção *Fill* e depois *Detailed.* Aparecerá uma janela onde temos de escolher o *Step size* (1), o *stop value* (10) e *Direction* (*down*).

(v) Para que na segunda coluna apareçam os quadrados dos números que estão na primeira coluna, na célula B2 escrevemos $(\sim A2)^2$. O símbolo $\sim$ é usado para que A2 seja entendido como a localização da célula e não como uma palavra.

De seguida seleccionamos a célula B2 e algumas mais para baixo. Vamos ao menu **Spreadsheet**, escolhemos a opção *Fill* e depois *down.*

Aparecerá uma tabela do género à que apresentamos na figura 1.15.



	A	B	C
1	n	n^2	
2	1	1	
3	2	4	
4	3	9	
5	4	16	
6	5	25	
7	6	36	
8	7	49	
9	8	64	
10	9	81	
11	10	100	
12			
13			

Figura 1.15: Exemplo de uma tabela feita no Maple

Existem várias propriedades da tabela que podem ser alteradas no menu **Spreadsheet**. Destacamos:

- O número de algarismos significativos com que o Maple executa internamente os cálculos (por defeito é 10);
- O número de casas decimais com que os resultados são apresentados (por defeito são 4);

Para saber mais sobre o Maple, para além do Help interactivo disponível no próprio programa, pode também consultar-se informação nas referências [6], [7], [8]. Também há muita bibliografia sobre Computação Algébrica e sobre o Maple na Internet, conforme podemos ver em [12], [13] e [14].

Capítulo 2

Procedimentos Maple para Sistemas Dinâmicos

Depois de introduzida a linguagem Maple, apresentamos aqui os procedimentos que permitem explorar junto do aluno os conceitos elementares de sistemas discretos unidimensionais. Assumimos que f é sempre uma aplicação real de variável real, isto é, tem domínio $A \subset \mathbb{R}$ e conjunto de chegada $\mathbb{R}$.

2.1 Órbita de um ponto

Tendo uma aplicação f, a órbita de um ponto $a \in \mathbb{R}$, é dada por $Orb_f(a) := (a, f(a), f^2(a), \ldots, f^n(a), \ldots)$. Sendo $Orb_f(a)$ uma sequência infinita de elementos, torna-se impossível representá-la, no Maple, na sua totalidade. Para procedermos à análise das órbitas escolhemos então "pedaços" de órbita entre instantes m e n, que por definição é a sequência $(f^m(a), \ldots, f^n(a))$. Temos essencialmente três opções para a exibição de pedaços de órbita:

- O uso da linha de comandos para obter a sequência $(f^m(a), \ldots, f^n(a))$, que veremos de seguida;
- O recurso às formalidades do uso de uma tabela, que veremos na secção 2.1.2;

- O método gráfico, que veremos na secção 2.2.

2.1.1 Procedimento listaorbita

Este procedimento pode ser usado para determinar um pedaço da órbita de um ponto x_0 sob a forma de lista. Tem como parâmetros f, x_0, m e n, onde:

- f é a aplicação para a qual se quer determinar a órbita;
- x_0 é o ponto inicial da órbita;
- m o instante a partir do qual a órbita vai ser calculada;
- n o instante final que se pretende.

Este procedimento fornece-nos a lista $[f^m(x_0), \ldots, f^n(x_0)]$. As duas primeiras linhas do procedimento

```
> listaorbita:=proc(f,x0,m,n)
  local L,k,x;
```

apresentam, como de costume, os parâmetros (f, x_0, m, n) e as variáveis locais utilizadas (L, k, x). A variável L é a lista onde o pedaço da órbita é armazenado; k é um contador e x representa cada um dos elementos $f^k(x)$. Na definição do procedimento **listaorbita** seguem-se as linhas do código.

```
> x:=x0:
  for k from 1 to m do;
  x:=f(x);
  od:
  L:=[x]:#L contém f^m(x)
```

onde, as primeiras $m-1$ iterações não são armazenadas em L, uma vez que só interessa o valor de $f^m(x_0)$, pois é a partir de m que queremos visualizar a órbita. Segue-se:

```
> for k from m+1 to n do
  x:=f(x);
  L:=[op(L),x];#L=(L,f^k(x))
  od:
  end proc:
```

Neste ciclo, determinamos $f^k(x)$ para $k = m+1, \ldots, n$. No início do ciclo **for** $L = f^m(x)$ e no fim $L = (f^m(x), f^{m+1}(x), \ldots, f^n(x))$.

Para determinar, por exemplo, $(f^k(0.01))_{k=0,\ldots,10}$ para a função $f(x) = x^2+1$, devemos escrever:

```
> f:=x->x^2+1:
> listaorbita(f,0.01,0,10);
```

respondendo o Maple com:

[0.01, 1.0001, 2.00020001, 5.000800080, 26.00800144, 677.4161389, 458893.6252, $0.2105833592\,10^{12}$, $0.4434535117\,10^{23}$, $0.1966510170\,10^{46}$, $0.3867162249\,10^{91}$]

Para visualizar o pedaço da órbita a partir do terceiro até ao décimo iterado, para a mesma função $f(x) = x^2 + 1$ e valor inicial $x_0 = 0.01$, devemos usar:

```
> f:=x->x^2+1:
> listaorbita(f,0.01,3,10);
```

[5.000800080, 26.00800144, 677.4161389, 458893.6252, $0.2105833592\,10^{12}$, $0.4434535117\,10^{23}$, $0.1966510170\,10^{46}$, $0.3867162249\,10^{91}$]

2.1.2 Tabelas

Para visualizar pedaços da órbita podemos usar a potencialidade do Maple em criar folhas de cálculo mergulhadas na sua área de trabalho. Seguindo o roteiro indicado na secção 1.10 do capítulo anterior, vamos indicar o procedimento para a obtenção da tabela correspondente a $x_0, \ldots, x_{10}$, onde $x_k = f^k(x)$, para $x_0 = 0.01$ e $f(x) = 2x(1-x)$.

Em primeiro lugar temos de definir a função com que estamos a trabalhar, o ponto inicial x_0 e o número de iterações n.

```
> f:=x->2*x*(1-x):# lei do sistema dinâmico
> x0:=0.01:#ponto inicial
> n:=10:#número de iterações
```

Como já foi referido, para construirmos uma tabela na área de trabalho do Maple temos de ir ao menu **Insert** e escolher a opção **Spreadsheet**.

Neste caso, para pôr em funcionamento a tabela, procedemos do seguinte modo:

(i) Na célula A1 escrevemos n. (Na primeira coluna vai aparecer o valor do n);

(ii) Na célula B1 escrevemos x_n. (Na segunda coluna vai aparecer a órbita);

(iii) Na célula A2 escrevemos 0 e na célula B2 escrevemos x_0;

(iv) Para que apareçam automaticamente os números de 0 a 10 na primeira coluna, seleccionamos a célula A2 e algumas mais para baixo (as suficientes para os 10 iterados), vamos ao menu **Spreadsheet**, escolhemos a opção **Fill** e depois **Detailed**. Aparecerá uma janela onde temos de escolher o **Step size** (1), o **stop value** (10) e **Direction** (**down**).

(v) Para gerar a órbita na segunda coluna, na célula B3 escrevemos $f(\sim B2)$. De seguida seleccionamos a célula B3 e algumas mais para baixo. No menu **Spreadsheet**, escolhemos a opção **Fill** e depois **down**.

E o resultado que se obtém é semelhante ao da figura 2.1.

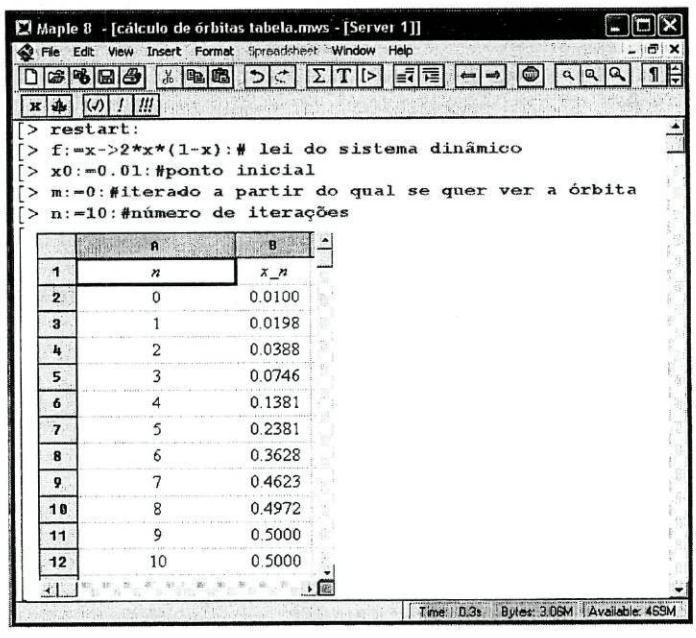


Figura 2.1: Folha de trabalho com uma tabela da órbita de um ponto

2.2 Método gráfico

Tendo os gráficos de uma função f e da recta $y = x$, representados no mesmo sistema de eixos e procedendo à identificação $a \leftrightarrow (a, a)$, podemos analisar a dinâmica de $a \mapsto f(a)$ pela dinâmica de $(a, a) \mapsto (f(a), f(a))$. Com base nesta identificação, se partirmos de um ponto (a, a), traçarmos a recta vertical que passa por (a, a), essa recta intersecta o gráfico de f no ponto $(a, f(a))$. De seguida, traça-se uma recta horizontal que passa por $(a, f(a))$. Esta recta intersecta a diagonal no ponto $(f(a), f(a))$. Repetindo o processo, a recta vertical que passa no ponto $(f(a), f(a))$ vai intersectar o gráfico de f no ponto $(f(a), f^2(a))$. A recta horizontal que passa por $(f(a), f^2(a))$ intersecta a diagonal no ponto $(f^2(a), f^2(a))$ e assim sucessivamente, podemos obter o conjunto de pontos $\{a, f(a), f^2(a), \ldots\}$.

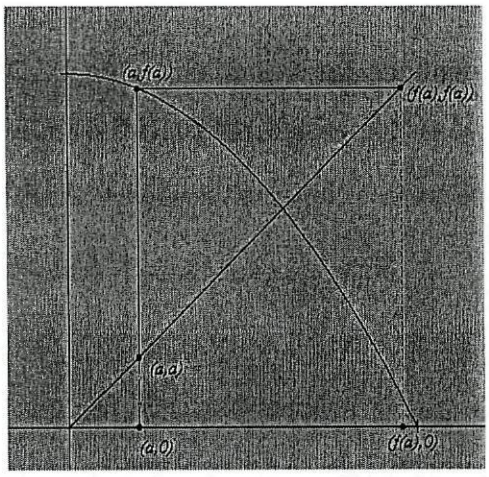


Figura 2.2: Método gráfico

Com base nesta ideia simples concebemos o procedimento plotorbita, que nos dá a representação gráfica da órbita de um ponto, conforme descrito. Os parâmetros de entrada são f, n, x_0, $xmin$, $xmax$, $ymin$ e $ymax$ onde:

- f é a aplicação para a qual se quer determinar a órbita;
- x_0 é o ponto inicial da órbita;
- n é o número de iterações pretendido;

- *xmin* e *xmax* são os valores entre os quais se pretende visualizar o eixo dos xx;
- *ymin* e *ymax* são os valores entre os quais se pretende visualizar o eixo dos yy.

Começamos com as definições dos parâmetros, variáveis locais e instrução **with(plots)**.

```
> plotorbita:=proc(f,n,x0,xmin,xmax,ymin,ymax)
  local x,p,i,diag:# variáveis usadas no procedimento
  with(plots):# Necessário porque vamos usar comandos gráficos
```

Agora, a partir do ponto inicial x_0 calculamos $x_1 = f(x_0)$ para que possamos desenhar o primeiro segmento de recta vertical: o que une (x_0, x_0) a (x_0, x_1).

```
> x[0]:=x0:
  x[1]:=f(x[0]):
```

Esse segmento é desenhado através do comando **PLOT(CURVES)** referido na secção 1.6. A seguir desenhamos no mesmo referencial o gráfico da função f e esse segmento de recta. A uma variável p à qual atribuímos o gráfico de f, juntamos este primeiro segmento.

```
> p:=plot(f(x),x=xmin..xmax):
  p:=p, PLOT(CURVES([[x[0],[x[0]],[x[0],x[1]]]),COLOR(RGB,0,0,1)):
```

Para completar o esquema usamos um ciclo **for** que permite que em cada instante i sejam desenhados o segmento de recta que une os pontos (x_{i-2}, x_{i-1}) e (x_{i-1}, x_{i-1}) e o segmento de recta que os pontos (x_{i-1}, x_{i-1}) e (x_{i-1}, x_i), sendo armazenados na variável p. Como já temos os dois primeiros iterados já só nos faltam $n-2$ e por isso a variável i começa a contagem a partir de 2, no ciclo **for**.

```
> for i from 2 to n do x[i]:=f(x[i-1]):
  p:=p,PLOT(CURVES([[x[i-2],x[i-1]],[x[i-1],x[i-1]],[x[i-1],x[i]]]),
  COLOR(RGB,0,0,1)):
  od:
```

Depois de termos todos os segmentos é preciso colocá-los todos no mesmo referencial juntamente com o gráfico da diagonal. Para isso, primeiro guardamos o gráfico da diagonal numa variável (*diag*) e depois fazemos **display** das variáveis p e *diag*.

```
> diag:=plot(x,x=xmin..xmax,color=green):
  display({p,diag},view=[xmin..xmax,ymin..ymax],font=[TIMES,ROMAN,14]);
  end proc:
```

A tabela 2.1 mostra alguns dos resultados obtidos com este procedimento, **plotorbita**, com a função $f(x) = 1 - 2x^2$, $x_0 = 0.2$ e para $n = 1, 4, 6, 7$ e 10.

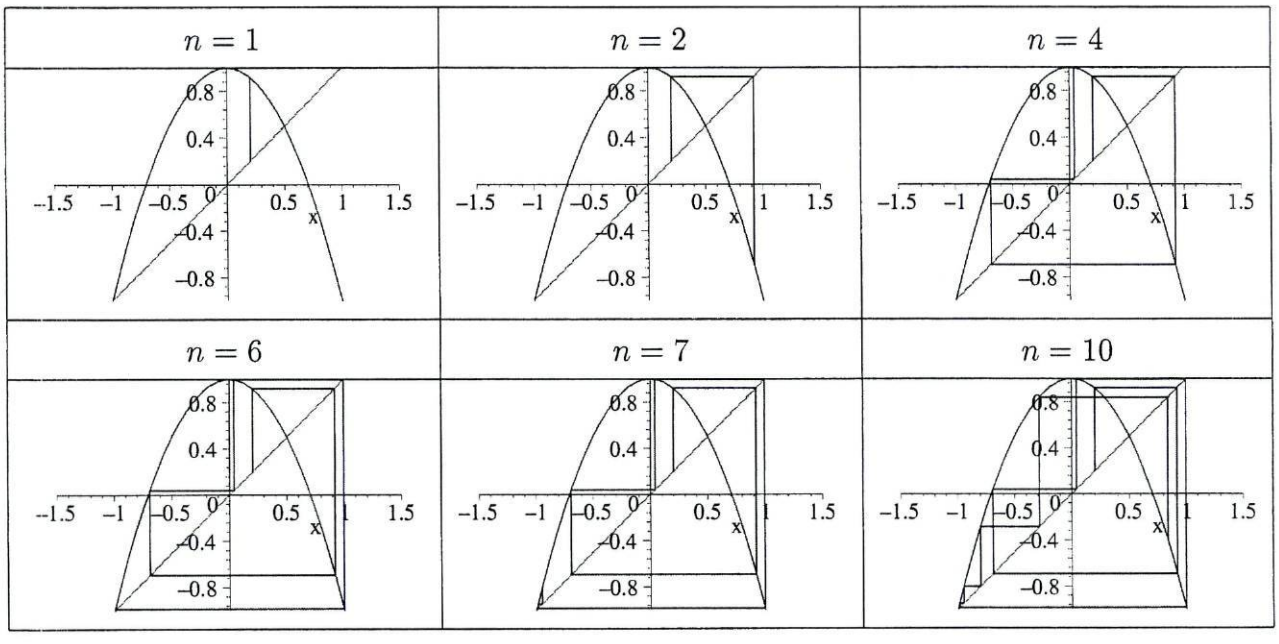


Tabela 2.1: Representação gráfica da órbita para diferentes iterações

Para obtermos os gráficos, em primeiro lugar temos de definir a função f, com que estamos a trabalhar.

```
> f:=x->1-2*x^2;
```

$$f := x \to 1 - 2\,x^2$$

A seguir é necessário definir os restantes parâmetros de entrada. Nos exemplos que exibimos na tabela 2.1, escolhemos $xmin = -1.5$, $xmax = 1.5$, $ymin = -1$ e $ymax = 1$. Assim, dependendo do valor de n considerado devemos escrever

```
> plotorbita:=proc(f,n,0.2,-1.5,1.5,-1,1);
```

No lugar do parâmetro n devemos escrever 1, 2, 4, 6, 7 e 10, respectivamente.

2.3 Comportamento periódico

Na secção anterior vimos como, dada uma função f, proceder para determinar pedaços da órbita de um ponto x, $(x_m, x_{m+1}, \ldots, x_n)$. Utilizando as técnicas aí descritas juntamente com as potencialidades do cálculo algébrico e numérico mostraremos agora como se consegue identificar que uma aplicação f possua pontos fixos e outros pontos periódicos.

2.3.1 Determinação de pontos fixos

Um ponto fixo de f é, por definição, um ponto x que satisfaz a condição $f(x) = x$. Para a determinação dos pontos fixos de uma função, existem várias maneiras de proceder. As estratégias aqui apresentadas dividem-se em três grupos: resolução algébrica, resolução numérica e esboço gráfico.

Resolução algébrica

Podemos determinar pontos fixos, resolvendo algebricamente a equação $f(x) = x$, usando o comando **solve(f(x)=x,x)**. Para determinarmos algebricamente, os pontos fixos de $f(x) = 1 - 2x^2$, devemos então escrever as seguintes instruções:

```
> f:=x->1-2*x^2:
> solve(f(x)=x,x);
```

$$-1, \frac{1}{2}$$

Neste caso, obtemos as duas soluções da equação $f(x) = x$, que são -1 e $-\frac{1}{2}$.

Como já referimos no capítulo 1, existem casos em que o Maple é incapaz de encontrar um valor para as soluções de uma equação, indicando apenas que existe solução. Por exemplo, não é possível determinar os pontos fixos de $f(x) = 1 - \exp(-\frac{1}{x^2})$. A resposta dada neste caso só nos permite concluir que a equação tem solução, mas não explicita o seu valor.

```
> solve(1-exp(-1/x^2)=x,x);
```

$$1 - e^{\mathrm{RootOf}(_Z - 2\,_Z\,e^{-_Z} + _Z\,(e^{-_Z})^2 + 1)}$$

Contudo, se quisermos saber se esta solução é um número real devemos escrever:

```
> type(solve(1-exp(-1/x^2)=x,x), realcons);
```

true

A resposta dada neste caso indica-nos que $1-\exp(-\frac{1}{x^2})$ tem um ponto fixo. Se a resposta dada fosse *false*, significava que a equação $f(x) = x$ tem raízes em $\mathbb{C}$, mas não em $\mathbb{R}$ e, portanto, no nosso contexto, f não tem pontos fixos (f é uma aplicação real de variável real).

Resolução numérica

Para determinar numericamente os pontos fixos de uma função usamos o comando **fsolve(f(x)=x,x,a..b)**, que tenta resolver numericamente a equação **f(x)=x** em ordem à variável **x**, no intervalo $[a, b]$. Por exemplo, se quisermos verificar se existem pontos fixos de $f(x) = 1 - 0.5x^2$ entre -1 e 1, devemos escrever:

```
> f:=x->1-0.5*x^2:
> fsolve(f(x)=x,x,-1..1);
```

0.7320508076

Como já referimos no capítulo 1, este comando tal como o comando **solve** nem sempre é eficaz. Existem casos em que a resposta dada por este comando não abrange todas as soluções da equação, como é o caso da equação $sin(x) - x^2 + x = x$, já discutido anteriormente.

Esboço gráfico

Pela simples observação de que uma função f tem um ponto fixo em a se e só se o ponto (a, a) pertence ao gráfico de f, ou seja, a função f tem um ponto fixo em a se e só se o seu gráfico intersecta a recta $y = x$ no ponto (a, a), podemos usar as potencialidades gráficas do Maple, para identificar os pontos de intersecção do gráfico de f com a diagonal $y = x$.

Com base nesta ideia, criamos o procedimento pontosfixosplot que tem como parâmetros f, $xmin$ e $xmax$, em que:

- f é a aplicação em estudo;

- *xmin* é o extremo esquerdo do intervalo;
- *xmax* é o extremo direito do intervalo.

Com estes parâmetros o procedimento dá-nos, no mesmo sistema de eixos, a representação gráfica da função f e da identidade.

```
> pontosfixosplot:=proc(f,xmin,xmax)
  local x,id,p,q;
  with(plots):# Vamos usar comandos gráficos
  id:=x->x:# Expressão analítica da identidade
  q:=plot(id(x),x=xmin..xmax,color=green):# Gráfico da identidade
  p:=plot(f(x),x=xmin..xmax):# Gráfico da função f
  display({p,q});
  end proc:
```

Para testar o procedimento **pontosfixosplot** para a função $f(x) = 1-2x^2$, $x \in [-1,1]$, devemos escrever:

```
> f:=x->1-2*x^2:

> pontosfixosplot(f,-1,1);
```

Com estas instruções, obtemos o gráfico da figura 2.3.

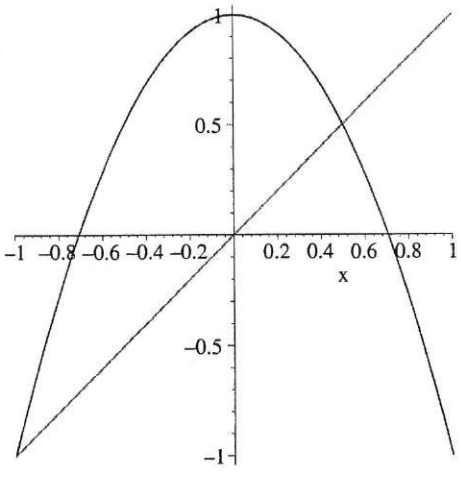


Figura 2.3: Pontos fixos de $f(x) = 1 - 2x^2$ no intervalo $[-1,1]$

Para que os pontos fixos sejam bem visíveis, podemos construir um procedimento que, para além, de exibir no mesmo referencial, os gráficos de f e da identidade, assinale os pontos fixos de f.

Esta opção não foi inserida no procedimento **pontosfixosplot**, pois, uma vez que se utiliza o comando **solve**, este procedimento só será eficaz no caso em que esse comando consegue resolver a equação $f(x) = x$.

Nos casos em que o comando **solve** consegue resolver a equação, o procedimento pontosfixosplots é constituído pelas seguintes linhas de comandos.

```
> pontosfixosplots:=proc(f,xmin,xmax)
  local x,id,p,q,r,sols,PF,i;
  with(plots):
  id:=x->x:
  q:=plot(id(x),x=xmin..xmax,color=green):
  p:=plot(f(x),x=xmin..xmax):
  sols:=[solve(f(x)=x,x)]:
  PF:=[seq([sols[i],sols[i]], i=1..nops(sols))]:
  r:=NULL:
  for i from 1 to nops(PF) do
  r:=r,PLOT(POINTS(PF[i],SYMBOL(DIAMOND)),TEXT(PF[i],''ponto
  fixo'',ALIGNBELOW,ALIGNRIGHT,FONT(TIMES,ROMAN,14))):
  end do:
  display({p,q,r},font=[TIMES,ROMAN,14],thickness=3,scaling=constrained);
  end proc:
```

Começa-se por determinar algebricamente os pontos fixos da função f, guardando essa informação na variável **sols**.

```
> sols:=[solve(f(x)=x,x)]:#pontos fixos de f
```

Em seguida, guarda-se numa lista **PF**, os pontos de coordenadas $(sols[i], sols[i])$.

```
> PF:=[seq([sols[i],sols[i]], i=1..nops(sols))]:#cria uma
  sequência de pontos constituída pelos pontos fixos de f
```

A seguir é usado um ciclo **for** que para cada i desenha os pontos da lista **PF**, guardando-os na variável **r**.

```
> r:=NULL:#garante que a lista r está vazia
  for i from 1 to nops(PF) do
  r:=r,PLOT(POINTS(PF[i],SYMBOL(DIAMOND)),TEXT(PF[i],''ponto
  fixo'',ALIGNBELOW,ALIGNRIGHT,FONT(TIMES,ROMAN,14))):
  end do:#desenha cada ponto crítico e ao seu lado escreve a
  palavra ponto crítico
```

Por fim usa-se o comando **display** para que os gráficos de f e da identidade sejam desenhados no mesmo referencial, juntamente com os pontos fixos de f.

```
> display({p,q,r});
  end proc:
```

Para testar o procedimento para a função $f(x) = 1 - 2x^2$, $x \in [-1, 1]$, devemos escrever:

```
> f:=x->1-2*x^2:
```

```
> pontosfixosplots(f,-1,1);
```

Obtemos o gráfico da figura 2.4.

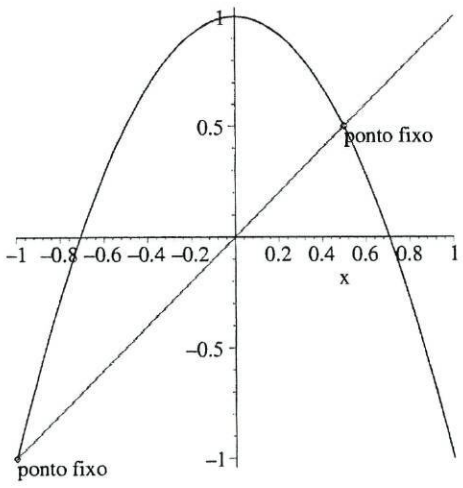


Figura 2.4: Pontos fixos de $f(x) = 1 - 2x^2$ no intervalo $[-1, 1]$

Nesta figura, o gráfico de f está representado a vermelho e o gráfico da identidade está representado a verde. Pela análise gráfica, podemos concluir que só existe intersecção do gráfico de f com a diagonal em dois pontos (os assinalados na figura 2.4), pelo que f

tem dois pontos fixos no intervalo $[-1, 1]$. Um encontra-se no intervalo $[-1, 0]$ e o outro encontra-se no intervalo $[0, 1]$. Se pretendermos uma maior precisão para o ponto fixo que se encontra em $[0, 1]$, podemos executar o comando

```
> pontosfixosplots(f,0,1);
```

e obtemos o gráfico da figura 2.5.

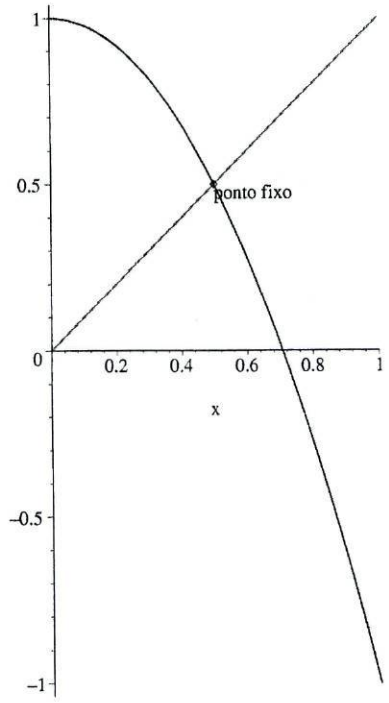


Figura 2.5: Pontos fixos de $f(x) = 1 - 2x^2$ no intervalo $[0, 1]$

Dependendo da precisão que pretendemos para o ponto fixo, podemos encurtar o intervalo no qual queremos testar a existência de pontos fixos. No caso do exemplo anterior, podemos ainda usar o procedimento **pontosfixosplots** no intervalo $[0.4, 0.6]$.

```
> pontosfixosplots(f,0.4,0.6);
```

E obtém-se o gráfico da figura 2.6.

Com este último gráfico pode-se confirmar que f tem um ponto fixo para $x = 0.5$ (o que algebricamente é imediato).

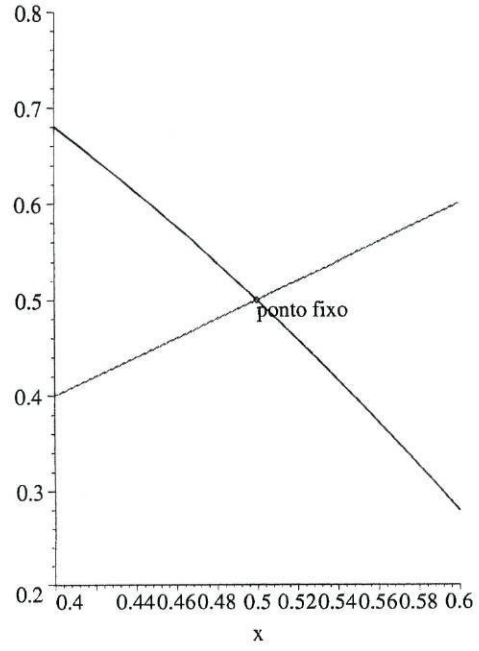


Figura 2.6: Pontos fixos de $f(x) = 1 - 2x^2$ no intervalo $[0.4, 0.6]$

2.3.2 Período dois

Por definição, um ponto x diz-se periódico se e só se satisfaz a equação $f^p(x) = x$, para algum $p \in \mathbb{N}$. Ao menor $p \in \mathbb{N}$ nestas condições, chama-se o período de x. Se x for um ponto periódico de período um, então é ponto fixo. Da definição de ponto periódico, segue que os métodos utilizados para determinar pontos fixos podem ser utilizados para determinar pontos periódicos de qualquer período.

Os pontos periódicos de período dois de uma função são aqueles que são solução da equação $f^2(x) = x$ e que não são pontos fixos de f, ou seja, que não são solução da equação $f(x) = x$. Podemos, portanto determiná-los algebricamente, numericamente e graficamente.

Algebricamente

Para determinar algebricamente os pontos periódicos de período dois de uma função criamos o procedimento perdois que tem como único parâmetro de entrada f que representa a função que estamos a estudar e como resposta dá-nos a lista dos pontos periódicos

de período dois de f.

```
> perdois:=proc(f)
  local i,sol,ss,fixos:
```

Na variável *sols2* guardam-se os pontos fixos de f^2 e na variável *fixos* guardam-se os pontos fixos de f.

```
> sols2:=[solve((f@@2)(x)=x,x)]:
  fixos:=solve(f(x)=x,x):
```

Como os pontos periódicos de período dois são aqueles que são solução da equação $f^2(x) = x$ e não são solução da equação $f(x) = x$, executamos o comando

```
> pdois:=sols2 minus fixos:
```

que guarda na variável *pdois* os pontos $x \in \mathbb{C}$ que satisfazem $f^2(x) = x$ e não são pontos fixos de f.

O ciclo **for** serve para verificar se cada um dos pontos x em *pdois* é real. Ao usarmos o comando **Re**(expressão), o Maple indica-nos a parte real de um número. Assim, fazemos um teste a cada elemento do conjunto *pdois* para ver se a parte real de cada um coincide com ele próprio.

```
> sol:=[]:
  for i from 1 to nops(pdois) do
  if (Re(pdois[i])=pdois[i])
```

Os pontos de *pdois* que são reais são guardados na variável *sol*.

```
> then sol:=[op(sol),pdois[i]]
  end if
  end do:
  sol:
  end proc:
```

Para determinar os pontos periódicos de período dois de $f(x) = 1 - 2x^2$ com este procedimento, escrevemos:

```
> f:=x->1-2*x^2:
```

```
> perdois(f);
```

$$[\frac{1}{4} + \frac{\sqrt{5}}{4}, \frac{1}{4} - \frac{\sqrt{5}}{4}]$$

Para as funções definidas por polinómios do segundo grau, o Maple responde prontamente. No entanto, não se pode esperar, com este método, a identificação de pontos periódicos de período dois para todos os polinómios de grau superior a dois, uma vez que conduziria à resolução algébrica de equações de grau superior a seis, para as quais não existe uma fórmula resolvente e para as quais o comando **solve** responde com a expressão **RootOf**.

Por exemplo, com este procedimento, ao inserirmos a função $f(x) = x^3 + 3x^2 - x - 1$, o Maple responde com a expressão **RootOf**.

```
> f:=x->x^3+3*x^2-x-1:

> perdois(f);
```

$$[\text{RootOf}(_Z^6 + 6_Z^5 + 8_Z^4 - 2_Z^3 + 4_Z^2 + _Z - 2,\ index = 1),$$
$$\text{RootOf}(_Z^6 + 6_Z^5 + 8_Z^4 - 2_Z^3 + 4_Z^2 + _Z - 2,\ index = 4)]$$

Para polinómios de grau superior a 15, o programa demora mais do que cinco minutos para responder [1], acontecendo, por vezes, que a memória virtual do computador falha. Por exemplo, para encontrar os pontos periódicos de período dois de $f(x) = x^{15} - 1$, o programa leva aproximadamente dezoito minutos a dar um resposta!

Numericamente

Tal como para os pontos fixos, podemos usar o comando **fsolve(equac,x,a..b)** para determinar numericamente os pontos periódicos de período dois de uma função.

Procedendo a uma pequena alteração do procedimento **perdois**, ou seja, trocando as ocorrências do comando **solve** pelo comando **fsolve**, obtemos o procedimento perdoisf, que nos dá a lista dos pontos periódicos de período dois de uma função, no intervalo $[xmin, xmax]$, utilizando o comando **fsolve**. Esse procedimento é composto pelos seguintes comandos:

[1] Aqui, e em todo o trabalho a medição do tempo tem por base um pc com processador Pentium IV a 1.5 GHz, com 256 Mb de memória RAM, com sistema operativo Windows Xp e espaço livre em disco de 10 Gbytes

```
> perdoisf:=proc(f,xmin,xmax)
  local i,fixos2,sols,fixos:
  fixos2:=[fsolve((f@@2)(x)=x,x,xmin..xmax)];
  fixos:=fsolve(f(x)=x,x,xmin..xmax);
  sols:=[]:
  for i from 1 to nops(fixos2) do
  if ( not is(fixos2[i] in {fixos})) then
  sols:=[op(sols),fixos2[i]]
  end if:
  end do:
  sols:
  end proc:
```

Por exemplo, se quisermos verificar a ocorrência de pontos periódicos de período dois de $f(x) = 1 - 2x^2$, no intervalo $[-1, 1]$, devemos escrever:

```
> f:=x->1-2*x^2:
> perdoisf(f,-1,1);
```

Com estas instruções obtemos a lista dos pontos periódicos de período dois entre -1 e 1 de $f(x) = 1 - 2x^2$.

$$[-0.3090169944,\ 0.8090169944]$$

Graficamente

O procedimento que a seguir apresentamos, perdoisplot, permite determinar graficamente os pontos periódicos de período dois. Como parâmetros de entrada tem: f, x_{min} e x_{max}, onde f é a função pretendida e $[x_{min}, x_{max}]$ um intervalo do seu domínio. Como resposta, o procedimento fornece-nos no mesmo sistema de eixos o gráfico de f, de f^2 e da identidade.

```
> perdoisplot:=proc(f,xmin,xmax)
  local A,B,Diag:
  with(plots):
  A:=plot(f(x),x=xmin..xmax,color=yellow,scaling=constrained):
  B:=plot((f@@2)(x),x=xmin..xmax,color=black,scaling=constrained):
  Diag:=plot(x,x=xmin..xmax,color=black,scaling=constrained):
  display({A,B,Diag});
  end proc:
```

Para testar este procedimento com a função $f(x) = 4x(1-x)$, $x \in [0,1]$, escrevemos:

```
> f:=x->4*x*(1-x):
> perdoisplot(f,0,1);
```

O resultado obtido está expresso na figura 2.7.

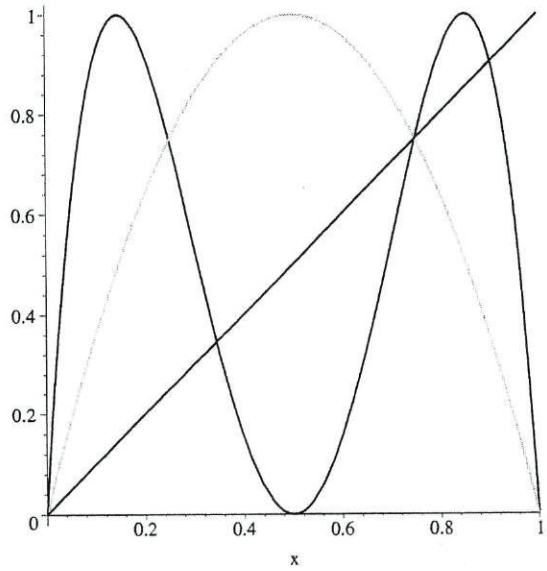


Figura 2.7: Pontos periódicos de período dois de $f(x) = 4x(1-x)$

O gráfico de f está representado a amarelo e o de f^2 e da identidade a preto. Os pontos de intersecção do gráfico de f^2 com o gráfico da identidade correspondem a pontos fixos de f^2. Para obtermos graficamente os pontos periódicos de período dois de uma função f, devemos considerar os pontos que são dados por $Graf(f^2) \cap Graf(Id) - Graf(f)$, ou seja, graficamente correspondem aos pontos onde só se intersectam os gráficos que estão representados com cor preta. No caso da função $f(x) = 4x(1-x)$, pode-se ver que esta tem dois pontos periódicos de período dois.

Tal como fizemos para os pontos fixos, também podemos modificar o procedimento **perdoisplot**, de modo que este ao ser executado assinale os pontos periódicos de período dois. Como já foi referido esta alteração só é bem sucedida no caso em que o comando **fsolve** consegue resolver as equações $f(x) = x$ e $f^2(x) = x$.

Começa-se por determinar numericamente os pontos fixos da função f e os pontos fixos

da função f^2, guardando essa informação nas variáveis **fixos** e **fixos2**, respectivamente.

```
> fixos:=fsolve(f(x)=x,x):
  fixos2:=fsolve((f@@2)(x)=x,x):
```

Depois de termos os pontos fixos de f e os pontos fixos de f^2, guarda-se na variável **sols** as raízes de f^2 que não são raízes de f [2]. Para isso, temos de usar os seguintes comandos:

```
> for i from 1 to nops(fixos2) do
  if (not is( fixos2[i] in {fixos})) then
  sols:=[op(sols),fixos2[i]]
  end if:
  end do:
```

Numa lista **PF2** guarda-se os pontos de coordenadas $(sols[i], sols[i])$.

```
> PF:=[seq([sols[i],sols[i]], i=1..nops(sols))]:#cria uma
  sequência de pontos constituída pelos pontos de período 2 de f
```

A seguir usa-se um ciclo **for** que para cada i desenha os pontos da lista **PF2**, guardando-os na variável **r**.

```
> r:=NULL:#garante que a lista r está vazia
  for i from 1 to nops(PF) do
  r:=r,PLOT(POINTS(PF[i],SYMBOL(DIAMOND)),TEXT(PF[i],
  ''período 2'',ALIGNBELOW,ALIGNRIGHT,FONT(TIMES,ROMAN,14))):
  end do:#desenha cada ponto periódico e ao seu lado escreve
  período 2
```

Por fim usa-se o comando **display** para que os gráficos de f^2, de f e da identidade sejam desenhados no mesmo referencial, juntamente com os pontos fixos de f^2.

```
> display({Diag,A,B,r});
  end proc:
```

O procedimento perdoisplots com esta funcionalidade tem o seguinte aspecto:

[2] Note-se que o comando **fsolve** já exibe só soluções que estão em $\mathbb{R}$

```
> perdoisplots:=proc(f,xmin,xmax)
  local A,B,Diag,fixos,fixos2,sols,i,PF2,r:
  with(plots):
  A:=plot(f(x),x=xmin..xmax,color=yellow,scaling=constrained):
  B:=plot((f@@2)(x),x=xmin..xmax,color=black,scaling=constrained):
  Diag:=plot(x,x=xmin..xmax,color=black,scaling=constrained):
  fixos:=fsolve(f(x)=x,x):fixos2:=[fsolve((f@@2)(x)=x,x)]:
  sols:=[]:
  for i from 1 to nops(fixos2) do
  if (not is( fixos2[i] in {fixos})) then sols:=[op(sols),fixos2[i]]
  end if:  end do:
  PF2:=[seq([sols[i],sols[i]], i=1..nops(sols))]:r:=NULL:
  for i from 1 to nops(PF2) do
  r:=r,PLOT(POINTS(PF2[i],SYMBOL(DIAMOND)),TEXT(PF2[i],''período
  2'',ALIGNBELOW,ALIGNRIGHT,FONT(TIMES,ROMAN,14))):  end do:
  display({Diag,A,B,r},font=[TIMES,ROMAN,14],thickness=3);
  end proc:
```

Para testar este procedimento com a função $f(x) = 4x(1-x)$, $x \in [0,1]$, escrevemos:

```
> f:=x->4*x*(1-x):
> perdoisplots(f,0,1);
```

O gráfico obtido é o que se mostra na figura 2.8.

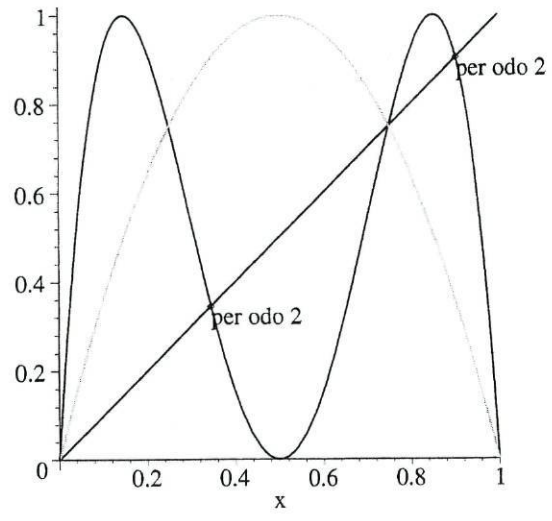


Figura 2.8: Indicação dos pontos periódicos de período dois de $f(x) = 4x(1-x)$

2.3.3 Período três

Para funções contínuas a existência de pontos periódicos de período três implica a existência de pontos periódicos de qualquer período, conforme se pode verificar em [9] e [4]. Por este motivo incluímos aqui procedimentos para o estudo da existência de período três. Os pontos periódicos de período três de uma função são aqueles que são solução da equação $f^3(x) = x$ e que não são pontos fixos de f, ou seja, que não solução da equação $f(x) = x$. Para determinar os pontos periódicos de período três podemos proceder de modo idêntico ao que fizemos para os de período um e dois.

Algebricamente

Para determinar algebricamente os pontos periódicos de período três de uma função, criamos o procedimento pertres que tem como único parâmetro de entrada f que representa a função que estamos a estudar e como resposta dá-nos a lista dos pontos periódicos de período três de f.

```
> pertres:=proc(f)
  local i,sols3,sol,ptres,fixos:
  sols3:={solve((f@@3)(x)=x,x)};
  fixos:={solve(f(x)=x,x)};
  ptres:=sols3 minus fixos:
  sol:=[]:
  for i from 1 to nops(ptres) do
  if (Re(ptres[i])=ptres[i]) then
  sol:=[op(sol),ptres[i]]
  end if:
  end do:
  sol:
  end proc:
```

Por exemplo, para obtermos os pontos de período três de $f(x) = 4x(1-x)$, devemos escrever:

```
> f:=x->4*x*(1-x):

> pertres(f);
```

e obtemos

[]

Esta resposta indica-nos que a função $f(x) = 4x(1-x)$ não tem pontos periódicos de período três, o que não é verdade! Este erro, deve-se ao uso do comando **solve** no procedimento. Quando as equações envolvidas são de grau superior a três, é natural que este comando nem sempre consiga apresentar um valor numérico para as soluções da equação, dando uma resposta que não está correcta.

Numericamente

Como já fizemos para os pontos fixos e para os pontos periódicos de período dois, podemos usar o comando **fsolve** para determinar os pontos periódicos de período três de uma função.

Procedendo a uma pequena alteração do procedimento **perdoisf**, obtemos o procedimento pertresf, que nos dá a lista dos pontos periódicos de período três de uma função, no intervalo $[xmin, xmax]$, utilizando o comando **fsolve**. Esse procedimento é composto pelos seguintes comandos:

```
> pertresf:=proc(f,xmin,xmax)
  local i,sols,fixos3,fixos:
  fixos3:=[fsolve((f@@3)(x)=x,x,x=xmin..xmax)];
  fixos:=fsolve(f(x)=x,x,x=xmin..xmax);
  sols:=[]:
  for i from 1 to nops(fixos3) do
  if ( not is(fixos3[i] in{fixos})) then
  sols:=[op(sols),fixos3[i]]
  end if:
  end do:
  sols:
  end proc:
```

Para determinar os pontos periódicos de período três de $f(x) = 4x(1-x)$, devemos escrever:

```
> f:=x->4*x*(1-x):

> pertresf(f);
```

[0.1169777784, 0.1882550991, 0.4131759112, 0.6112604670, 0.9504844340, 0.9698463104]

Graficamente

Usando um procedimento semelhante ao dos pontos periódicos de período dois, também é possível exibir graficamente os pontos periódicos de período três de uma função.

O procedimento pertresplot é em tudo semelhante ao **perdoisplot**. A única diferença reside no facto de em **pertresplot** considerarmos $f^3(x)$ em vez de $f^2(x)$. Analogamente ao que referimos para os pontos periódicos de período dois, $Graf(f^3) \cap Graf(Id) - Graf(f)$ dá-nos os pontos periódicos de período três.

```
> pertresplot:=proc(f,xmin,xmax)
  local A,B,Diag:
  with(plots):
  A:=plot(f(x),x=xmin..xmax,color=yellow):
  B:=plot((f@@3)(x),x=xmin..xmax,color=black):
  Diag:=plot(x,x=xmin..xmax,color=black):
  display({Diag,A,B},font=[TIMES,ROMAN,14],scaling=constrained);
  end proc:
```

Para executarmos o procedimento para a função $f(x) = 4x(1-x)$, $x \in [0,1]$, devemos escrever:

```
> f:=x->4*x*(1-x):

> pertresplot(f,0,1);
```

Obtemos assim, o gráfico da figura 2.9, onde f está representado a amarelo, f^3 e a identidade a preto. Deste modo, para além dos pontos fixos, o gráfico de $f^3(x)$ intersecta-se com o gráfico da identidade em seis novos pontos, ou seja, f tem seis pontos periódicos de período três, no intervalo $[0,1]$.

Tal como para os pontos fixos e para os pontos periódicos de período dois, é possível,

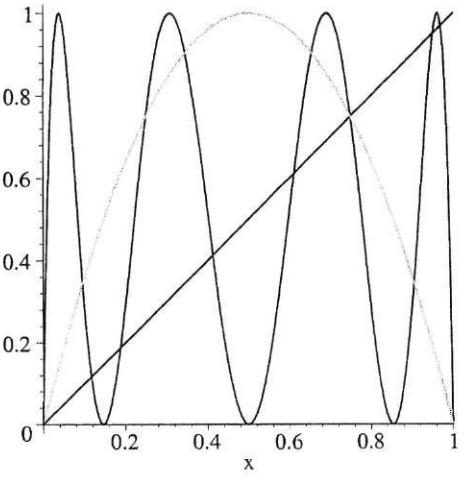


Figura 2.9: Pontos periódicos de período três de $f(x) = 4x(1-x)$

em certos casos (quando o comando **fsolve** resolve as equações $f(x) = x$ e $f^3(x) = x$), assinalar, no gráfico, os pontos periódicos de período três. Para isso, temos de juntar os comandos

```
> fixos:=fsolve(f(x)=x,x):fixos3:=[fsolve((f@@3)(x)=x,x)]:
  sols:=[]:
  for i from 1 to nops(fixos3) do
  if (type(fixos3[i], realcons) and not is( fixos3[i] in
  {fixos})) then sols:=[op(sols),fixos3[i]] end if
  end do:
  PF3:=[seq([sols[i],sols[i]], i=1..nops(sols))]:
  r:=NULL:
  for i from 1 to nops(PF2) do
  r:=r,PLOT(POINTS(PF3[i],SYMBOL(DIAMOND)),TEXT(PF3[i],''período 3'',
  ALIGNBELOW,ALIGNRIGHT,FONT(TIMES,ROMAN,14))):
  end do:
  display({Diag,A,B,r},font=[TIMES,ROMAN,14],thickness=3);
```

ao procedimento **pertresplot** antes do comando de finalização

```
> end proc:
```

Com estes novos comandos, os executar os comandos

```
> f:=x->4*x*(1-x):
```

```
> pertresplot(f,0,1);
```

Obtemos o gráfico da figura 2.10.

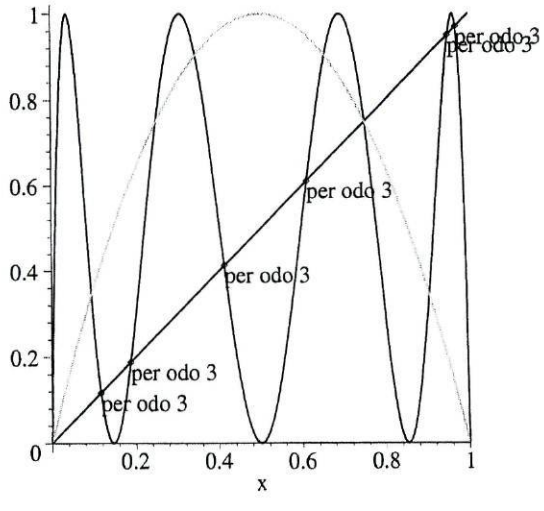


Figura 2.10: Indicação dos pontos periódicos de período três de $f(x) = 4x(1-x)$

Como se pode ver por este gráfico existem casos em que a marcação dos pontos periódicos no gráfico não ajuda a uma melhor visualização, tornando o gráfico bastante confuso. Isto pode ser evitado retirando as legendas do gráfico, ou seja, substituindo no procedimento anterior

```
> r:=r,PLOT(POINTS(PF3[i],SYMBOL(DIAMOND)),TEXT(PF3[i],''período
  3'',ALIGNBELOW,ALIGNRIGHT,FONT(TIMES,ROMAN,14))):
```

por

```
> r:=r,PLOT(POINTS(PF3[i],SYMBOL(DIAMOND))):
```

2.3.4 Outros pontos periódicos

Como já vimos, dependendo do grau de complexidade da função, o Maple deixa de conseguir responder ao comando **solve** e **fsolve**, tornando assim inviável a apresentação de procedimentos algébricos e numéricos baseados nestes dois comandos. Devido a estes impedimentos a abordagem proposta para o estudo dos pontos periódicos de período superior a três assenta no método gráfico.

Para determinar graficamente os pontos periódicos de período p, em que p é primo, basta esboçar no mesmo referencial os gráficos de f, f^p e da identidade. Como exemplos temos o caso dos pontos periódicos de período dois e três que já vimos atrás. No caso do período k, em que k não é primo, temos de desenhar no mesmo referencial os gráficos da função identidade e das funções f^j em que j divide k. O procedimento que a seguir apresentamos, pernplot, dá-nos estas representações gráficas. Como parâmetros de entrada tem f, n, $xmin$ e $xmax$, onde:

- f representa a função em estudo;
- n representa o período para o qual se quer determinar os pontos periódicos;
- $xmin$ e $xmax$ são os valores entre os quais se pretende visualizar o eixo dos xx (os pontos periódicos).

```
> pernplot:=proc(f,n,xmin,xmax)
  local i,d,pp:
  with(numtheory):# módulo que permite determinar os divisores de
  um número
  with(plots):
  d:=divisors(n):cálculo dos divisores de n
```

Para que sejam facilmente detectáveis os pontos periódicos de período n de f, os gráficos da identidade e de f^n são desenhados a preto e os restantes são desenhados a amarelo.

```
> pp:=NULL:
  for i from 1 to nops(d)-1 do
  pp:=pp,plot((f@@d[i])(x),x=xmin..xmax,color=yellow);
  end do:
  pp:=pp,plot({(f@@d[nops(d)])(x),x},x=xmin..xmax,color=black):
  display(pp);
  end proc:
```

Por exemplo, para determinar graficamente os pontos periódicos de período seis de uma função, **pernplot** desenha no mesmo referencial, para além do gráfico da identidade,

os gráficos de f, f^2, f^3 e de f^6, uma vez que os divisores de seis são: 1, 2, 3 e 6. Para $f(x) = 3.6x(1-x)$, o resultado obtido é o da figura 2.11.

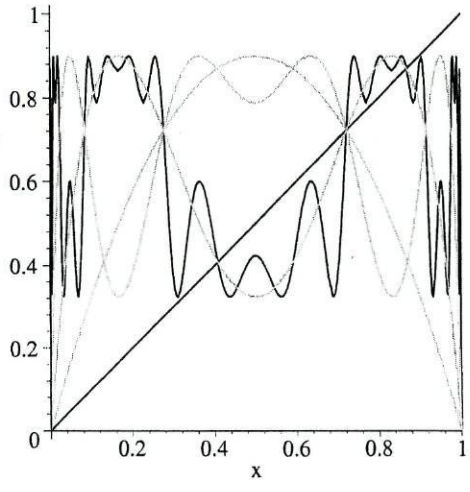


Figura 2.11: Pontos periódicos de período seis de $f(x) = 3.6x(1-x)$

2.4 Análise estatística de órbitas

Existem casos em que o esboço gráfico de uma órbita não nos fornece indicações claras acerca do seu comportamento qualitativo e que somente através da representação gráfica da órbita fica impossível determinar o estado de $x_n = f^n(x_0)$ de um ponto inicial x_0. Se o aluno usar o procedimento plotorbita para analisar os cinquenta primeiros iterados, por exemplo, da órbita de $x_0 = 0.2$ pela função $f(x) = 4x(1-x)$, escrevendo os seguintes comandos:

```
> f:=x->4*x*(1-x):

> plotorbita(f,50,0.2);
```

obtém o gráfico da figura 2.12. Observando esta representação gráfica da órbita, o aluno não poderá concluir que x_0 tem comportamento periódico.

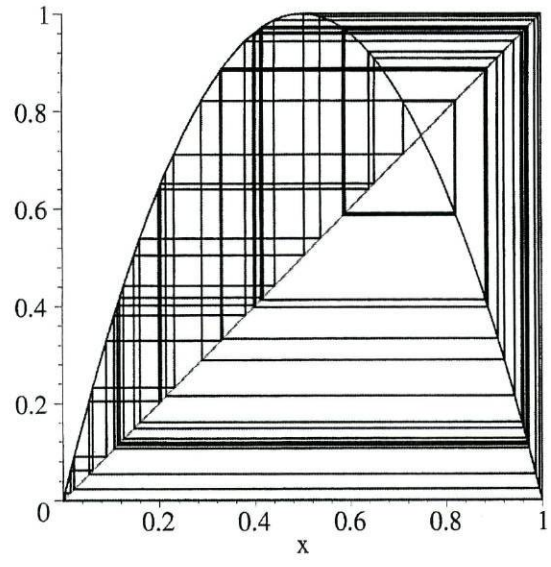


Figura 2.12: Órbita de $x_0 = 0.2$ pela função $f(x) = 4x(1-x)$

2.4.1 Diagrama de frequências

Para recolhermos informação de uma órbita podemos recorrer ao procedimento **diagramafreqs**, que não é mais do que um método de análise estatística. Este procedimento permite identificar numericamente intervalos I que sejam mais significativos na evolução do sistema, ou seja, que correspondam a passagens mais ou menos frequentes de $f^k(x)$, $k \in \mathbb{N}$ [3]. O procedimento diagramafreqs tem como parâmetros de entrada f, a, b, N, x_0 e n, em que:

- f é a função em estudo;
- a é o extremo esquerdo do intervalo;
- b é o extremo direito do intervalo;
- N é o número de subintervalos de igual amplitude de $[a, b]$ a considerar;
- x_0 é o ponto inicial;

[3] Isto significa que estudamos a percentagem de iterados $f^k(x)$ que está em I, para $k \leq n$, onde $n \in \mathbb{N}$ é um inteiro "grande" fixado previamente.

- n é o número de iterados a considerar.

Com estes parâmetros, o procedimento divide o intervalo $[a, b]$ em N subintervalos igualmente espaçados e para cada um deles vai contar o número de elementos do pedaço da órbita $\{x_0, f(x_0), \ldots, f^n(x_0)\}$ que estão nesse intervalo. Feito isto, faz uma representação gráfica desses valores em forma de histograma. Obtemos assim, um gráfico de barras contíguas, de base $I_i = [a + i\frac{b-a}{N}; a + (i+1)\frac{b-a}{N}]$ e altura h. Com isto, obtemos uma "distribuição"dos elementos da órbita pelo intervalo dado.

O procedimento **diagramafreqs** é constituído pelas seguintes linhas de comandos:

```
> diagramafreqs:=proc(f,a,b,N::integer,x0,n::integer)
  local x,i,aa,bb,k,data,V:
  with(stats[statplots]):#necessário para tratamento estatístico
  with(plots):
  x:=evalf(x0): aa:=evalf(a): bb:=evalf(b):
  for i from 1 to n do
  V[i]:=0: end do: for i from 1 to n do
  V[floor(N*(x-aa)/(bb-aa))+1]:=V[floor(N*(x-aa)/(bb-aa))+1]+1:
  x:=f(x):
  end do:
  data:=[]:
  for k from 0 to N-1 do
  data:=[op(data),Weight(k*(b-a)/N..(k+1)*(bb-aa)/N,V[k+1])]:
  end do:
  display(histogram(data,color=red));
  end proc:
```

As variáveis locais usadas neste procedimento são:

- x que representa cada um dos iterados $f^k(x_0)$;
- i representa o índice do subintervalo;
- k é um contador usado no ciclo **for**;
- aa e bb que representam respectivamente os extremos esquerdo e direito do intervalo;

- V que é um vector onde é armazenada a frequência relativa associada a cada subintervalo;

- a variável *data* que armazena o histograma

O primeiro bloco de instruções que é constituído por:

```
> x:=evalf(x0): aa:=evalf(a): bb:=evalf(b):
  for i from 1 to n do
  V[i]:=0:
  end do:
  for i from 1 to n do
  V[floor(N*(x-aa)/(bb-aa))+1]:=V[floor(N*(x-aa)/(bb-aa))+1]+1:
  x:=f(x):
  end do:
```

serve para determinar os subintervalos de $[a, b]$ e determinar a frequência associada a cada um deles. O comando **floor** do Maple serve para determinar a característica de um número. Neste procedimento **floor** é usado para indicar o subintervalo I_i a que pertence determinado elemento da órbita.

O bloco de instruções que se segue

```
> data:=[]:
  for k from 0 to N-1 do
  data:=[op(data),Weight(k*(b-a)/N..(k+1)*(bb-aa)/N,V[k+1])]:
  end do:
  display(histogram(data,color=red));
  end proc:
```

serve para construir o histograma.

Para vermos o resultado obtido com, por exemplo, $f(x) = 4x(1-x)$, $x_0 = 0.2$, considerando 1000 iterações e dividindo o intervalo $[0, 1]$ em 100 partes, devemos escrever:

```
> f:=x->4*x*(1-x):

> diagramafreqs(f,0,1,100,0.2,1000);
```

e obtemos o diagrama da figura 2.13.

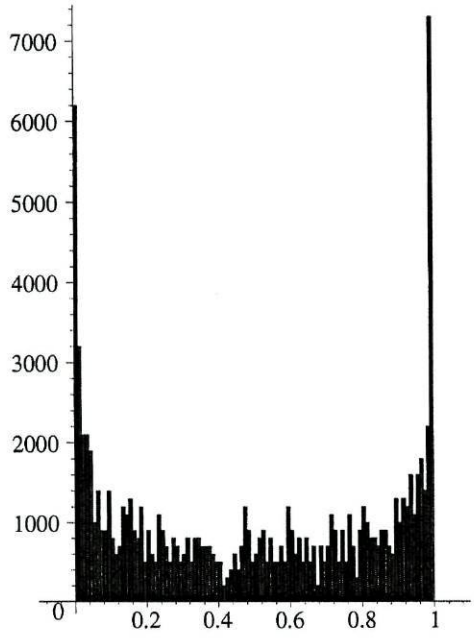


Figura 2.13: Diagrama de frequências de $x_0 = 0.2$ pela função $f(x) = 4x(1-x)$

Também podemos modificar o procedimento **diagramafreqs** de modo a que o histograma corresponda a frequências relativas, isto é, a área de cada barra é igual a $\#\{k \in \mathbb{N} : f^k(x_0) \in I\}$. Para isso, basta substituir

```
> data:=[op(data),Weight(k*(b-a)/N..(k+1)*(bb-aa)/N,V[k+1])]:
```

por

```
> data:=[op(data),Weight(k*(b-a)/N..(k+1)*(bb-aa)/N,V[k+1]/n)]:
```

2.4.2 Funções teste

Por vezes, o estudo qualitativo de um Sistema Dinâmico definido por uma função f conduz-nos à análise do comportamento das chamadas funções teste, ao longo de uma determinada órbita de f. Nomeadamente, para uma função h, estudamos as expressões do tipo $S_n = \frac{1}{n}\sum_{k=0}^{n-1} h(x_k)$, quando $n \to +\infty$ para um dado ponto inicial x_0 que são conhecidas como "Somas de Birkhoff". Em particular, quando consideramos $h(x) = \ln(|f'(x)|)$, obtemos os conhecidos "Expoentes de Lyapunov" para um ponto inicial x_0 (ver [1]). O expoente de Lyapunov (L) é então definido por $\lim_{n\to+\infty} L_n$ onde

$$L_n = \frac{1}{n}(\ln|f'(x_0)| + \ln|f'(x_1)| + \ldots + \ln|f'(x_n)|)$$

(quando o limite existe). Note-se que se na órbita existe um ponto x_i tal que $f'(x_i) = 0$, o expoente de Lyapunov não está definido.

A seguir apresentamos o procedimento Lyapunov, que tem como parâmetros de entrada f, n e x_0 e em que:

- f é a aplicação em estudo;
- n é o número de iterações pretendido;
- x_0 é o ponto inicial.

Com estes parâmetros, o procedimento determina o L_n para o valor inicial x_0, permitindo assim estudar os citados expoentes de Lyapunov.

```
> Lyapunov:=proc(f,n,x0)
  local dg,x1,i,L,h:
  dg:=unapply(diff(f(x),x),x);
  h:=x->ln(abs(df(x))):
  x1:=x0:
  L:=0:
  for i from 0 to n do
  L:=L+h(x1):
  x1:=f(x1):
  od:
  L:=L/n:
  end proc:
```

Para determinar, por exemplo, L_{500} para $x_0 = 0.1$ e $f(x) = 3.85x(1-x)$, devemos escrever:

```
> f:=x->3.85*x*(1-x):

> x0:=0.1:

> n:=500:
```

```
> Lyapunov(f,n,x0);
```

e obtemos

0.09712923488

Este procedimento pode ser facilmente modificado para tratar somas de Birkhoff associadas a uma função teste.

2.5 Codificação de órbitas

Muitas vezes fica impossível determinar exactamente a posição de $x_n = f^n(x_0)$ de um ponto inicial x_0. A codificação de um sistema tem como finalidade a tentativa de ao simplificar o sistema, permitir a obtenção de algumas das suas propriedades de um modo mais simples. Estrategicamente, podemos reduzir à precisão do cálculo de x_0, recorrendo a uma partição $P = X_0, \ldots, X_k$ do domínio de f. Em vez de visualizar x_n, indica-se o elemento X_i da partição P a que x_n pertence. Temos assim, para cada n, o valor $s(n)$ definido por $s(n) = j \Leftrightarrow x_n \in X_j$. Atendendo à definição de $s(n)$, é costume designar a sequência $(s_n)_{n\in\mathbb{N}}$ por itinerário do ponto x_0.

Para funções reais de variável real e para uma lista crescente de números reais $c_0 < \ldots < c_{p+1}$, temos determinada uma partição P de $\mathbb{R}$ através de $[c_{i-1}; c_i[_{i=0\ldots p+1}$, em que $c_{-1} = -\infty$ e $c_{p+1} = +\infty$.

Para determinar o itinerário de um ponto x_0 é necessário ter o ponto x_0, a partição L e o número de iterações que se pretende. Depois de ter a partição P, é necessário calcular os elementos da órbita e determinar o símbolo que está associado a cada um desses elementos. O procedimento que a seguir apresentamos, simbolo, tem como parâmetros de entrada x e L, em que:

- x representa o elemento da órbita;
- L é uma partição em forma de lista.

Com estes parâmetros, o procedimento fornece o símbolo associado a cada elemento x da órbita.

```
> simbolo:= proc(x,L)
  local ss,i; ss é a variável que guarda o símbolo associado a cada
  valor de x
  ss:=-1:#0 que quer dizer que o ponto está fora da partição
  for i from 1 to nops(L) while x>=L[i] do
  ss:=ss+1:
  od:
  ss;
  end proc:
```

Dada uma lista $L = [c_1, \ldots, c_k]$, definindo $X_i = [c_i, c_{i+1}]$, atribuímos ao intervalo X_i o símbolo i. Assim, se houver algum elemento ao qual seja atribuído o símbolo $k+1$, de imediato saberemos que ele está fora da partição L (para a direita). O símbolo -1 significa que o elemento está fora da partição L (à esquerda). Este procedimento utiliza um ciclo **for** que serve para obter o símbolo associado a cada elemento da órbita. Inicializa-se $ss = -1$ e, enquanto x for maior do que cada um dos elementos de L, aumenta-se uma unidade à variável ss; caso contrário, mantém-se o valor de ss. NO final do ciclo **for**, ss contém o símbolo associado a cada x_i.

Se quisermos, por exemplo, usar este procedimento para determinar o símbolo associado ao número 5 na partição $L = [0, 2, 4, 6]$, devemos escrever os seguintes comandos:

```
> x:=5:

> L:=[0,2,4,6]:

> simbolo(x,L);
```

a resposta é

2

pois $5 \in X_2 = [4, 6]$.

Tendo este procedimento, criamos um procedimento que nos dá o itinerário de x pela partição L, mais precisamente, dá-nos a lista $[s(0), s(1), \ldots, s(n)]$.

Este procedimento, itinerario, tem como parâmetros de entrada f, L, x_0 e n, onde:

- f é a função em estudo;

- L é uma partição em forma de lista;
- x_0 é o ponto inicial;
- n é o número de iterações a considerar.

```
> itinerario:=proc(f,L,x0::float,n::integer)
  local codigo,y,i;# A variável codigo guarda os elementos do itinerário
```

O procedimento começa por avaliar o ponto x_0 pelo procedimento **simbolo** e guardar o resultado na variável codigo.

```
> codigo:=[simbolo(x0,L)]:
```

De seguida, calcula a órbita do ponto x_0 e para cada um dos seus elementos determina (através do procedimento **simbolo**) o símbolo que lhe está associado, guardando-o na variável codigo.

```
> y:=x0;
  for i from 1 to n do
  y:=f(y):
  codigo:=[op(codigo),simbolo(y,L)]:
  end do:
  end proc:
```

Como resposta **itinerario**(f, L, x_0, n) dá-nos a lista dos símbolos associados a cada elemento da órbita, ou seja, dá-nos uma lista $[s(1), \ldots, s(n)]$, em que $s(k) = i \iff f^k(x_0) \in X_i$ [4].

Suponhamos que queremos determinar o itinerário do ponto $\frac{1}{3}$ pela função $f(x) = \text{sen}\,(\frac{1}{x})$, no intervalo $[0, \pi]$. Consideremos a partição $L = [0, 0.4, 0.8, 1, 1.2, \pi]$ do intervalo $[0, \pi]$ e $n = 10$. Para usar o procedimento **itinerario**, devemos escrever o seguinte:

```
> f:=x->sin(1/x):
> x0:=evalf(1/3):
> n:=10:
> L:=[0,0.4,0.8,1,1.2,Pi]:
```

[4] À sequência $s(n)$ chamamos itinerário

```
> itinerario(f,L,x0,n);
```

$$[0, 0, 1, 2, 2, 2, 2, 2, 2, 2, 2]$$

Para funções multimodais [5], $f : [a, b] \to \mathbb{R}$, diferenciáveis com um número finito de pontos críticos $c_1 < \ldots < c_{p-1}$ é costume considerar a partição $a = c_0 < c_1 < \ldots < c_p = b$, em que $X_i = [c_i; c_{i+1}[$.(ver [2])

Paras determinar os pontos críticos de uma função f, no intervalo $[a, b]$, podemos recorrer ao comando **CriticalPoints**.

```
> with(Student[Calculus1]):#Necessário para usar o comando que determina
os pontos críticos

> CriticalPoints(f(x),x=a..b):
```

Este comando fornece os pontos críticos da função f no intervalo $[a, b]$, por ordem crescente e em forma de lista.

Por exemplo, para determinarmos os pontos críticos de $f(x) = 3x^5 - 5x^3 + 3$ no intervalo $[-2, 2]$ devemos escrever:

```
> with(Student[Calculus1]):#módulo que permite usar o comando CriticalPoints

> CriticalPoints( 3*x^5-5*x^3+3,x=-2..2);
```

$$[-1, 0, 1]$$

Tal como para a função anterior, se quisermos analisar, por exemplo, o itinerário para $f(x) = 2x^3 + 5x^2 - 4x$, no intervalo $[-3, 2]$, de $x_0 = \frac{1}{3}$ e para dez iterações, devemos primeiro determinar a partição L. Para isso, e uma vez que se trata de uma função multimodal, determinamos primeiro os pontos críticos de f.

```
> f:=x->2*x^3+5*x^2-4*x:

> a:=-3:

> b:=2:

> with(Student[Calculus1]):

> CP:=CriticalPoints(f(x),x=a..b):
```

[5] Com um número finito de pontos críticos.

Depois de termos a lista crescente dos pontos críticos (CP), podemos criar a partição L do intervalo $[a, b]$.

```
> L:=[a,op(L),b];
```

$$L := [-3, -2, \frac{1}{3}, 2]$$

Agora que já temos todos os parâmetros de entrada, estamos em condições de usar o procedimento **itinerario**.

```
> L:=[a,op(CP),b]:
> itinerario(f,L,1/3,10);
```

$$[2, 1, 3, 3, 3, 3, 3, 3, 3, 3]$$

Ao subdividir um intervalo em dois e fazendo a correspondente codificação, o estudo das órbitas reduz-se a estudar se os iterados estão no intervalo da esquerda ou no da direita. A codificação pode ser útil quando é mais fácil provar que uma órbita passa, por exemplo, mais vezes no intervalo esquerdo do que no direito. Para que se distinga rapidamente o intervalo mais frequente, é usual recorrer-se a um diagrama de frequências associado aos símbolos que correspondem a cada elemento da órbita.

O procedimento que a seguir apresentamos, diagramafreq, dá-nos o diagrama de frequências associado a um determinado itinerário. Com ele podemos ver quais os códigos mais frequentes correspondentes a um sistema. Como parâmetros de entrada tem f, L, x_0 e n, em que:

- f é a função em estudo;
- L é uma partição em forma de lista;
- x_0 é o ponto inicial;
- n é o número de iterados.

O primeiro bloco de instruções do procedimento:

```
> diagramafreq:=proc(f,L,x0::float,n::integer)
  local y,i,j,k,simbolo,data,legendas,V:
  global codigo:
  with(stats[statplots]):with(plots):
  codigo:=[simbolo(x0,L)]:
  y:=x0;
  for i from 1 to n do
  y:=f(y): codigo:=[op(codigo),simbolo(y,L)]:
  end do:
  simbolo:= proc(x,L)
  local ss,i;
  ss:=-1:
  for i from 1 to nops(L) while x>=L[i] do
  ss:=ss+1: od: ss;
  end proc:
```

serve para determinar a órbita do ponto inicial x_0 e determinar o símbolo (através do procedimento **simbolo**) que está associado a cada um dos elementos da órbita (itinerário).

As instruções que apresentamos a seguir têm como uma única função no procedimento, embelezá-lo para que se torne mais agradável visualmente e facilite a interpretação dos resultados obtidos. Para tal, recorremos a alguns comandos do Maple que permitem obter as legendas de cada uma das colunas.

```
> legendas:=NULL:
  for i from 1 to nops(L)+1 do
  legendas:=legendas,PLOT(POINTS(NULL),TEXT([i-3/2,-1/3],convert(i-2,string),
  ALIGNBELOW,FONT(HELVETICA,OBLIQUE,10))):
  V[i]:=0:
  end do:
```

Finalmente, as últimas instruções são usadas para determinar o valor de cada uma das colunas e para desenhar o histograma.

```
> for j from 1 to nops(codigo) do
  V[codigo[j]+2]:=V[codigo[j]+2]+1;
  end do:
  data:=[]:
  for k from -1 to nops(L)-1 do
  data:=[op(data), Weight(k..k+1,V[k+2])]:
  end do:
  legendas:=legendas,PLOT(POINTS(NULL),TEXT([0,-0.9],'''),AXESSTYLE(BOX),
  AXESTICKS(0,DEFAULT));
  display({legendas,histogram(data,color=red)});
  end proc:
```

Como resposta, o procedimento fornece-nos num diagrama de frequências a frequência de cada um dos símbolos associados ao itinerário. No eixo horizontal estão representados os símbolos e no eixo vertical a frequência, ou seja, o número de vezes que ocorreu determinado símbolo.

No caso do exemplo anterior, para $f(x) = 2x^3 + 5x^2 - 4x$, no intervalo $[-3, 2]$, $x_0 = \frac{1}{3}$ e para cem iterações, escrevemos:

```
> f:=x->2*x^3+5*x^2-4*x:
> a:=-3:  b:=2:
> x0:=evalf(1/3):
> with(Student[Calculus1]):
> L:=CriticalPoints(f(x),x=a..b):
> L:=[a,op(L),b]:
> diagramafreq(f,L,x0,100);
```

e obtemos o diagrama de frequências da figura 2.14.

Analisando esta figura, o aluno poderá observar que o símbolo mais frequente é o -1, pelo que pode concluir que a partir de determinado iterado, os elementos da órbita estão fora do intervalo $[-3, 2]$.

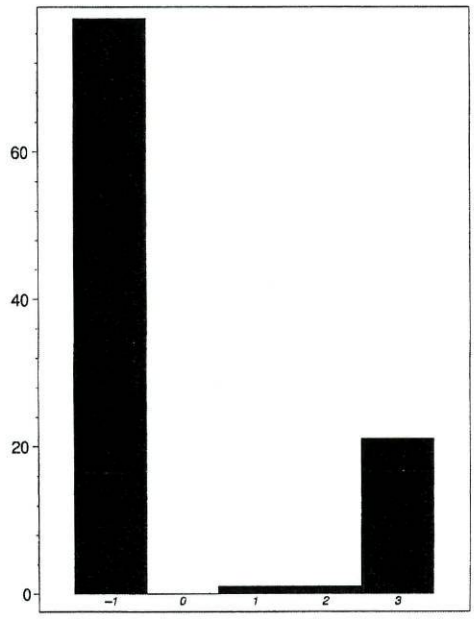


Figura 2.14: Diagrama obtido com o procedimento **diagramafreq**

Capítulo 3

A aplicação tenda

Neste capítulo vamos aplicar os utilitários desenvolvidos anteriormente ao estudo da família de aplicações do tipo $(T_a)_{a\in[0,2]}$, em que:

$$T_a(x) = \begin{cases} ax & \text{se } 0 \leq x < \frac{1}{2} \\ a(1-x) & \text{se } \frac{1}{2} \leq x \leq 1 \end{cases}. \tag{3.1}$$

Para um aluno do último ano do Ensino Secundário a expressão analítica deste tipo de funções é simples de entender e as suas propriedades dinâmicas básicas podem ser facilmente compreendidas com a ajuda de programas feitos em Maple.

Com o estudo que fazemos, pretendemos que o aluno no final tome consciência das seguintes características desta família de aplicações:

- o aspecto do gráfico de $T_a(x)$;
- os valores de a considerados são, de facto, aqueles para os quais $T_a(x) \in [0,1]$ para todo o $x \in [0,1]$ e por consequência a órbita de todos esses pontos está contida nesse intervalo;
- identificar comportamentos assimptóticos elementares, tais como:
  - se $a < 1$, então $\omega(x) = 0$, $\forall x \in \mathbb{R}$;
  - se $x \notin [0,1]$ e $a > 1$, então $\omega(x) = -\infty$.

3.1 Esboço gráfico

Como já foi referido na secção 1.5.1 para declararmos uma função definida por ramos temos de recorrer ao comando **piecewise**. Assim, o comando

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
```

define a família dada em 3.1. Formalmente temos $T : \mathbb{R}^2 \to \mathbb{R}^2$ e para um valor de a (variável real que chamaremos de parâmetro) temos então $T_a(x) = T(a, x)$.

Esta função depende do valor atribuído ao parâmetro a. Por exemplo, para $a = 0.8$, depois de termos definido a função, o gráfico da aplicação $T_{0.8}$ pode ser obtido através das seguintes instruções:

```
> a:=0.8:  #Escolheu-se 0.8 para o parâmetro
> with(plots):#Necessário para o esboço gráfico
> plot(T(a,x),x=0..1,scaling=constrained);
```

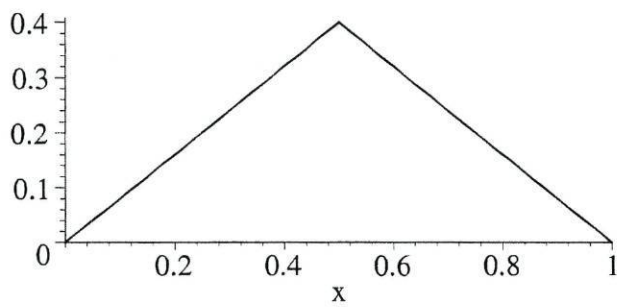


Figura 3.1: Gráfico da aplicação tenda para $a = 0.8$

Esta família de aplicações, devido ao aspecto geométrico do seu gráfico, é conhecida por família de aplicações do tipo tenda, ou mais simplesmente por aplicações tenda. Para domínio desta aplicação consideramos, quando nada é dito em contrário, o intervalo $[0, 1]$. Usando o procedimento **animafunção**, que a seguir apresentamos, verifica-se que $T_a([0, 1]) \subseteq [0, 1]$. Assim, o aluno pode ser facilmente conduzido à identificação desta propriedade de invariância da família tenda.

O procedimento animafunção permite visualizar o gráfico de uma aplicação f em função de um parâmetro a entre *amin* e *amax*. Como parâmetros de entrada tem f,

amin, *amax* e *passo*, em que:

- f representa a função em estudo:
- *amin* o valor mínimo de a;
- *amax* o valor máximo de a;
- *passo* o incremento usado para passar de um valor de a para o seguinte.

O procedimento é constituído pelos seguintes comandos:

```
> animafunção:=proc(f,amin,amax,passo)
  local a,timeplot:
  with(plots):
  timeplot:=NULL:
  for a from 0 to 2 by passo do
  timeplot:=timeplot,plot(f(a,x),x=0..1,y=0..1,title=cat('a=',convert(a,
  string))):
  end do:
  display([timeplot],insequence=true,scaling=constrained,thickness=3);
  end proc:
```

Na tabela que se segue são exibidos alguns dos gráficos, da aplicação tenda, obtidos com o procedimento **animafunção** para diferentes valores de a entre 0 e 2.

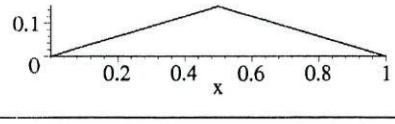
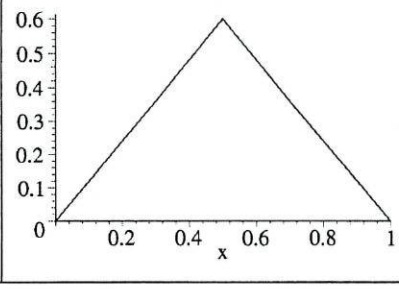
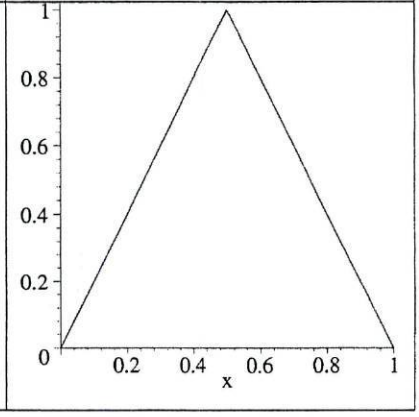
$a = 0.3$	$a = 1.2$	$a = 2$
		

Tabela 3.1: Gráficos da aplicação tenda obtidos com o procedimento **animafunção**

Com estes gráficos ou outros similares, o aluno será levado a concluir que para valores de a entre 0 e 2, qualquer valor de x em $[0,1]$ é enviado por T_a em $[0,1]$. Na figura 3.2 exibimos o gráfico de $T_{2.1}$ que pode servir para mostrar ao aluno que se $a > 2$ nem todos os pontos de $[0,1]$ são enviados por T_a em $[0,1]$. Para obter o gráfico representado na figura são necessários os seguintes comandos:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> a:=2.1:
> with(plots):
> plot(T(a,x),x=0..1,scaling=constrained,font=[TIMES,ROMAN,14],
  thickness=3);
```

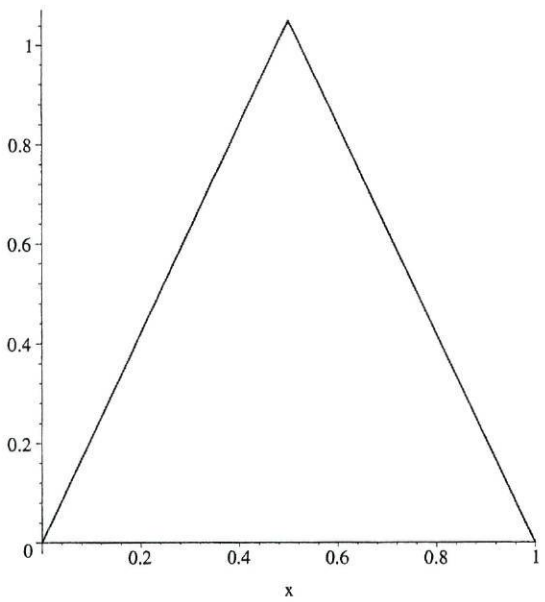


Figura 3.2: Gráfico da aplicação tenda para $a = 2.1$

Podemos pedir aos alunos que observem este gráfico ou até mesmo mais do que um e que concluam que para valores de $a > 2$, de facto nem todos os pontos de $[0,1]$ são enviados por T_a em $[0,1]$. Este também pode ser o momento indicado para justificar a restrição do estudo da aplicação tenda a valores de a no intervalo $[0,2]$. Se pedirmos aos alunos para observarem os vários gráficos da tabela 2.1, facilmente eles podem concluir que o vértice

(que corresponde ao ponto onde a função atinge o valor máximo) tem coordenadas $(\frac{1}{2}, \frac{a}{2})$ e portanto, para que $T_a([0,1]) \subseteq [0,1]$, é necessário que $0 \leq \frac{a}{2} \leq 1$, ou seja, $0 \leq a \leq 2$.

3.2 Análise de órbitas

Recorrendo aos métodos usados na secção 2.1 podemos determinar órbitas para diferentes valores do parâmetro a e do ponto inicial x_0.

3.2.1 Procedimento listaorbita

Nesta secção vamos usar o procedimento **listaorbita** para determinar os dez primeiros iterados de x_0 ($\{T_a(x_0), \ldots, T_a^{10}(x_0)\}$), para diferentes valores do parâmetro a e pontos iniciais em $[0,1]$.

Para podermos usar este procedimento é imprescindível que primeiro se defina a função T_a.

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
```

(i) Comecemos por observar o caso em que $a = \frac{1}{2}$ e $x_0 = \frac{1}{4}$:

```
> a:=1/2:#Valor do parâmetro a
```

```
> listaorbita(f,1/4,0,10);
```

$$[\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}, \frac{1}{512}, \frac{1}{1024}, \frac{1}{2048}, \frac{1}{4096}]$$

Apesar de termos considerado um pequeno número de iterados, o aluno deverá suspeitar que os iterados estão a aproximar-se de 0. Esta conjectura poderá ser reafirmada considerando **listaorbita**$(f, \frac{1}{4}, 0, n)$, com n "elevado".

(ii) Vamos agora ver o caso em que $a = 1$ e $x_0 = \frac{1}{3}$.

```
> a:=1
```

```
> listaorbita(T,1/3,0,10);
```

$$[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$$

Neste caso a órbita é um ponto fixo. Podemos fazer notar aos alunos que basta considerar o primeiro iterado para chegar a esta conclusão, uma vez que $T(\frac{1}{3}) = \frac{1}{3}$. Variando x_0, o aluno identifica os pontos fixos de $T_1(x)$, com o comando **listaorbita**$(T, x_0, 0, 1)$.

(iii) Para $a = \frac{3}{2}$ e $x_0 = \frac{3}{5}$, os resultados obtidos são os seguintes:

```
> a:=3/2
> listaorbita(T,3/5,0,10);
```

$$[\frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}, \frac{3}{5}]$$

Aqui, o aluno identifica um ponto fixo, mas para o mesmo valor do parâmetro só encontrará o 0 como outro ponto fixo.

(iv) Para $a = 2$ e $x_0 = \frac{2}{7}$ tem-se que:

```
> a:=2:
> listaorbita(T,2/7,0,10);
```

$$[\frac{2}{7}, \frac{4}{7}, \frac{6}{7}, \frac{2}{7}, \frac{4}{7}, \frac{6}{7}, \frac{2}{7}, \frac{4}{7}, \frac{6}{7}, \frac{2}{7}, \frac{4}{7}]$$

Neste caso, encontra-se um órbita periódica de período três $\{\frac{2}{7}, \frac{4}{7}, \frac{6}{7}\}$.

(v) Para $a = 2$ e $x_0 = \frac{1}{7}$ temos o seguinte:

```
> a:=2
> listaorbita(T,1/7,0,10);
```

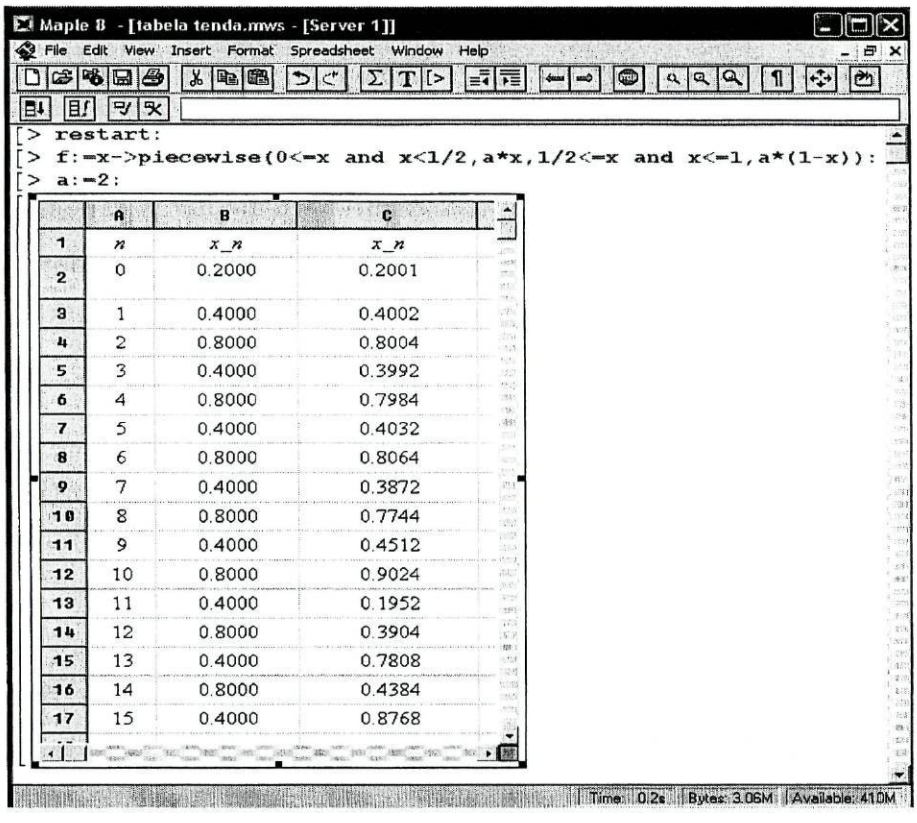
$$[\frac{1}{7}, \frac{2}{7}, \frac{4}{7}, \frac{6}{7}, \frac{2}{7}, \frac{4}{7}, \frac{6}{7}, \frac{2}{7}, \frac{4}{7}, \frac{6}{7}, \frac{2}{7}]$$

Neste caso, parece que $x_0 = \frac{1}{7}$ é enviado por T_a na órbita periódica de período três $\{\frac{2}{7}, \frac{4}{7}, \frac{6}{7}\}$.

3.2.2 Recurso às tabelas

Como já vimos na secção 2.1.2, o estudo das órbitas pode ser feito recorrendo às tabelas do Maple. Assim, vamos usar uma tabela para visualizar um pedaço da órbita da aplicação tenda com o parâmetro $a = 2$ e para duas escolhas do valor inicial: $x_0 = 0.2$ e $x_0 = 0.2001$.

A tabela que se mostra na figura 3.3 exibe os primeiros quinze iterados pela aplicação tenda, com $a = 2$, para os dois valores dos pontos iniciais x_0. Na coluna B aparecem os iterados $x_n = T_a^n(x_0)$ sendo $x_0 = 0.2$ e na coluna C aparecem os iterados $x_n = T_a^n(x_0)$ sendo $x_0 = 0.2001$. Comparando os valores destas duas colunas da tabela, podemos ver facilmente que $x_0 = 0.2$ é enviado na órbita periódica $\{0.4, 0.8\}$. Para $x_0 = 0.2001$ os resultados expressos na coluna C não permitem aferir um comportamento não periódico.



	A	B	C
1	n	x_n	x_n
2	0	0.2000	0.2001
3	1	0.4000	0.4002
4	2	0.8000	0.8004
5	3	0.4000	0.3992
6	4	0.8000	0.7984
7	5	0.4000	0.4032
8	6	0.8000	0.8064
9	7	0.4000	0.3872
10	8	0.8000	0.7744
11	9	0.4000	0.4512
12	10	0.8000	0.9024
13	11	0.4000	0.1952
14	12	0.8000	0.3904
15	13	0.4000	0.7808
16	14	0.8000	0.4384
17	15	0.4000	0.8768

Figura 3.3: Primeiros quinze elementos da órbita de $T_2(0.2)$ e de $T_2(0.2001)$

No caso da análise dos iterados, sempre que surja uma situação semelhante à apresentada na coluna B, isto é, onde se pode observar que $\exists k : T_a^k(x) = x$, podemos concluir que se trata de um comportamento periódico. Contudo, a não existência, na tabela, de uma órbita periódica explícita, nomeadamente quando existem poucos iterados, como no exemplo da coluna C, não implica que não exista uma órbita com um período maior do que o número de dados que são apresentados.

O facto de aumentarmos o número de iterados não facilita a análise da órbita, tonando-a, pelo contrário, mais difícil e parece-nos que pouco motivadora para um aluno. Parece-nos claro que se exibíssemos a um aluno uma tabela com um elevado número de entradas e fosse pedido a identificação de órbitas periódicas, provavelmente, ele não ficaria nada agradado com a tarefa e desistiria facilmente.

Para tentar ultrapassar esta dificuldade e como já foi referido na secção 2.2, o método do gráfico parece ser mais eficaz no estudo de órbitas.

3.2.3 Análise gráfica

No caso da aplicação tenda, para podermos usar o procedimento **plotorbita** fizemos uma pequena alteração, que foi introduzir-lhe mais um parâmetro de entrada que representa o valor do parâmetro a a considerar. Assim, os comandos do procedimento plotorbitamod são:

```
> plotorbitamod:=proc(f,a,n,x0)
  local x,p,i,diag:
  with(plots):
  x[0]:=x0:
  x[1]:=f(a,x[0]):
  p:=plot(f(a,x),x=0..1):
  p:=p, PLOT(CURVES([[x[0],x[0]],[x[0],x[1]]]),COLOR(RGB,0,0,1)):
  for i from 2 to n do x[i]:=f(a,x[i-1]):
  p:=p,PLOT(CURVES([[x[i-2],x[i-1]],[x[i-1],x[i-1]]]),COLOR(RGB,0,0,1)):
  p:=p,PLOT(CURVES([[x[i-1],x[i-1]],[x[i-1],x[i]]]),COLOR(RGB,0,0,1))
  od:
  diag:=plot(x,x=0..1,color=green):
  display({p,diag},font=[TIMES,ROMAN,14],thickness=2,scaling=constrained);
  end proc:
```

Para esboçar o gráfico da órbita de T_a para $a = 2$, para 100 iterados e para $x_0 = 0.2$ e $x_0 = 0.2001$, podemos escrever:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):

> plotorbitamod(T,2,100,0.2);#Para x0=0.2
```

```
> plotorbitamod(T,2,100,0.2001);#Para x0=0.2001
```

Obtendo-se os resultados da figura 3.4 .

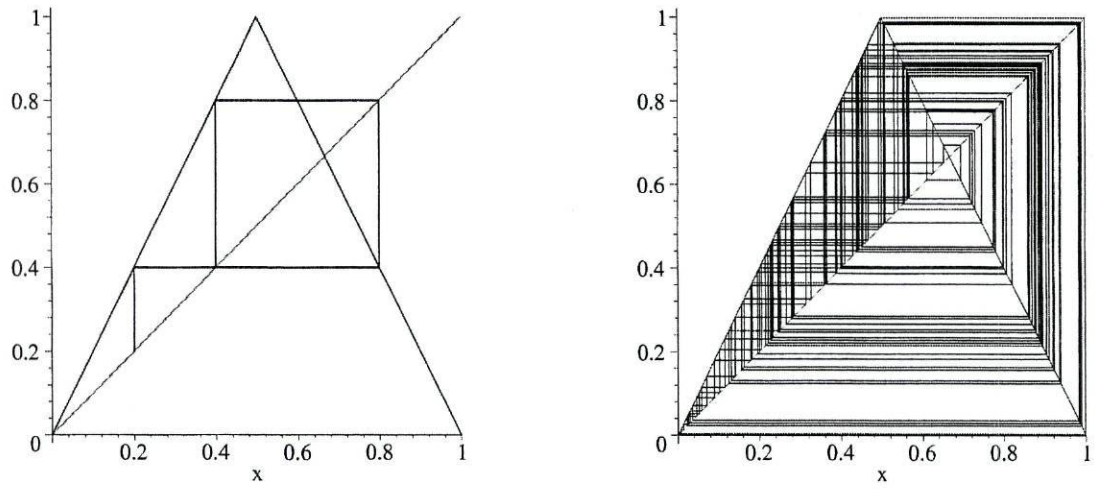


Figura 3.4: Representação gráfica da órbita para $x_0 = 0.2$ e para $x_0 = 0.2001$

Mais uma vez, o aluno pode ver que $x_0 = 0.2$ é enviado por T_a na órbita periódica $\{0.4; 0.8\}$ e x_0 não parece apresentar um comportamento periódico.[1]

3.2.4 Variação com o parâmetro a

Apesar da forma do gráfico destas aplicações ser simples e das equações que as envolvem serem lineares, ao analisarmos a órbita de diferentes pontos desta aplicação obtemos resultados distintos.

Com o recurso ao procedimento plotorbitamod será fácil interpretar as diferenças de comportamento existente para $a < 1$ e $a > 1$.

Para valores de $a < 1$

O aluno é levado a concluir que todas as órbitas convergem para 0. Depois de definida a função T_a, podemos executar os comandos:

```
> plotorbitamod(T,0.2,60,0.5);
```

[1] Convergir para uma órbita periódica

```
> plotorbitamod(T,0.5,60,0.5);
> plotorbitamod(T,0.8,60,0.6);
> plotorbitamod(T,0.95,60,0.6);
```

para obtermos os gráficos exibidos na tabela 3.2.

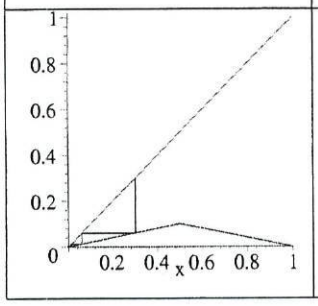
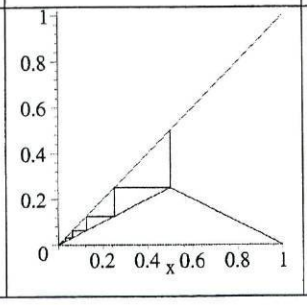
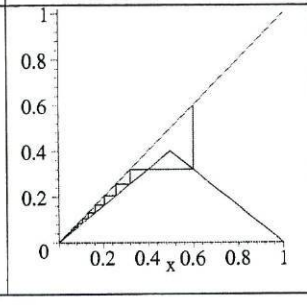
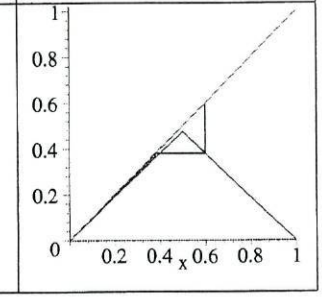
$a = 0.2$ e $x_0 = 0.3$	$a = 0.5$ e $x_0 = 0.5$	$a = 0.8$ e $x_0 = 0.6$	$a = 0.95$ e $x_0 = 0.6$
			

Tabela 3.2: Representação gráfica de órbitas com $a < 1$

Observando os gráficos da tabela 3.2, o aluno pode verificar que em todos os casos considerados $\omega(x) = 0$. Para além destes exemplos podem ser exibidos outros com $a < 1$ para que não restem dúvidas que para todo o $x_0 \in \mathbb{R}$ se tem $\lim_{n \to +\infty} T_a^n(x_0) = 0$.

Para valores de $a > 1$

O aluno deverá inferir que para $a > 1$ já o cenário é distinto do caso $a \leq 1$. Na tabela 3.3 exibimos algumas representações gráficas de órbitas para valores de a superiores a um. Depois de definida a função T_a, os gráficos foram obtidos com as instruções:

```
> plotorbitamod(T,1.3,60,0.8);
> plotorbitamod(T,1.8,60,0.3);
> plotorbitamod(T,1.5,60,0.4);
```

Para os valores considerados, parece não existir um comportamento periódico, uma vez que não se consegue encontrar um ponto ou um conjunto finito de pontos para o qual a órbita se esteja a "dirigir".

Para que o aluno fique com a noção que se $a > 2$ existem órbitas de pontos x para as quais $\omega(x) = -\infty$, podemos apresentar-lhe um ou vários exemplos em que tal acontece.

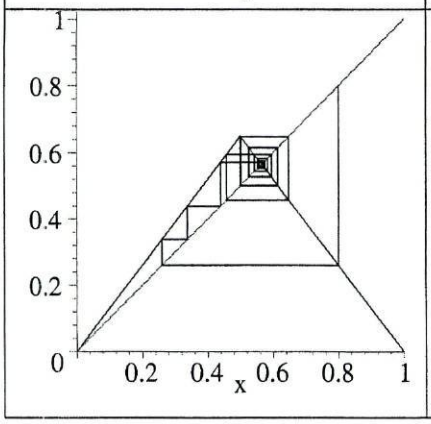
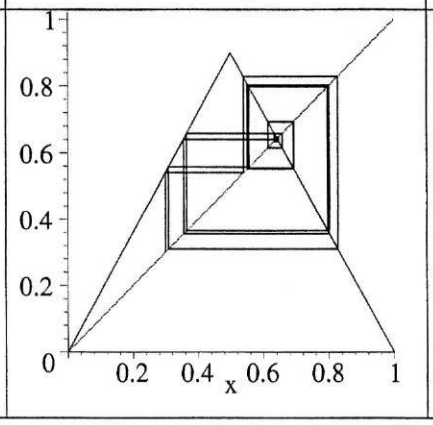
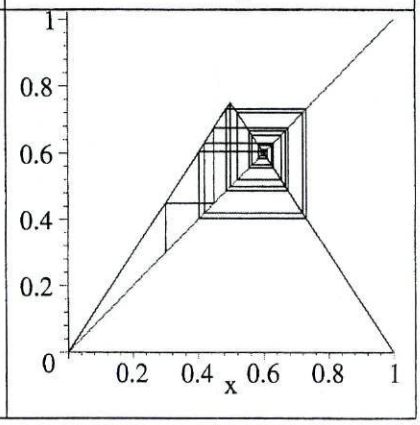
$a = 1.3$ e $x_0 = 0.8$	$a = 1.8$ e $x_0 = 0.3$	$a = 1.5$ e $x_0 = 0.4$
		

Tabela 3.3: Representação gráfica de órbitas com $a > 1$

```
> T:=(a,x)->piecewise(x<1/2,a*x,1/2<=x,a*(1-x)):
```

Para podermos usar o procedimento **plotorbita**, temos de fazer algumas alterações, nomeadamente no que diz respeito à amplitude dos eixos. Se quisermos, por exemplo, esboçar a órbita de $T_{2.3}(0.4)$ devemos escrever

```
> plotorbitamod:=proc(f,a,n,x0)
  local x,p,i,diag:
  with(plots):
  x[0]:=x0:
  x[1]:=f(a,x[0]):
  p:=plot(f(a,x),x=-10..10,y=-10..3):
  p:=p, PLOT(CURVES([[x[0],x[0]],[x[0],x[1]]]),COLOR(RGB,0,0,1)):
  for i from 2 to n do
  x[i]:=f(a,x[i-1]):
  p:=p,PLOT(CURVES([[x[i-2],x[i-1]],[x[i-1],x[i-1]]]),COLOR(RGB,0,0,1)):
  p:=p,PLOT(CURVES([[x[i-1],x[i-1]],[x[i-1],x[i]]]),COLOR(RGB,0,0,1))
  od:
  diag:=plot(x,x=-10..10,y=-10..3,color=green):
  display({p,diag},font=[TIMES,ROMAN,14],thickness=2,scaling=constrained);
  end proc:

> plotorbitamod(T,2.3,100,0.4);
```

Na figura 3.5 apresentamos o resultado obtido com estas instruções.

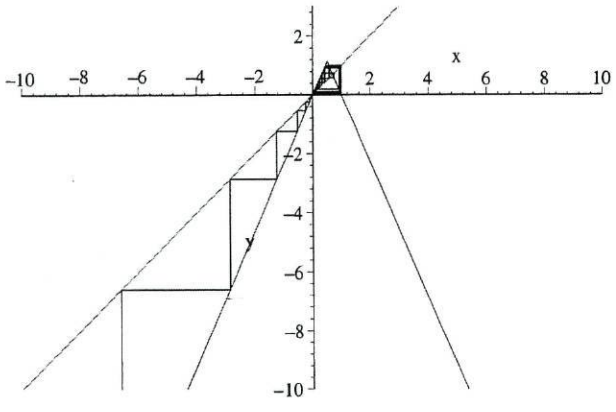


Figura 3.5: Órbita de $T_{2.3}(0.4)$

Através deste exemplo ou de outro semelhante, o aluno facilmente pode concluir que se $a > 2$ o $\omega(x) = -\infty$ para valores de x numa pequena vizinhança de $\frac{1}{2}$.

3.3 Pontos fixos

Como já vimos na secção 2.3.1, para determinar os pontos fixos de uma função podemos usar quer métodos analíticos quer métodos geométricos.

3.3.1 Métodos analíticos

Para determinar analiticamente os pontos fixos da função T_a vamos usar os comandos **solve** e **fsolve** conforme descrito na secção 2.3.1.

Comando solve

Vamos começar por tentar determinar algebricamente os pontos fixos. Para tal devemos escrever:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> solve(T(a,x)=x,x);
```

$$0$$

Com esta simples abordagem apenas obtemos como resposta um dos pontos fixos da função: o ponto 0. Tal acontece porque dependendo do valor de a considerado a equação pode ter um ou dois zeros e o Maple não consegue manipular simultaneamente as duas situações.

Para vermos a expressão geral dos zeros da função quando $a > 1$, devemos usar o comando

```
> assume(a>1):
```

que permite indicar ao Maple que o valor de a é maior do que um. Dada esta instrução, para visualizarmos todos os zeros possíveis da equação $T(a,x) = x$ no caso de $a > 1$, temos de escrever o comando:

```
> solve(T(a,x)=x,x);
```

O resultado obtido é:

$$0, \frac{a^{\sim}}{a^{\sim}+1}$$

Uma vez que utilizamos o comando **assume** para a variável a, sempre que esta for referenciada pelo Maple, aparecerá como $a^{\sim}$ para lembrar ao utilizador que esta variável está restrita a um conjunto de valores previamente definido.

Para obtermos a expressão geral dos pontos fixos de T_a quando $a < 1$ temos de escrever:

```
> assume(a<1):
> solve(T(a,x)=x,x);
```

$$0$$

O resultado obtido confirma que para $a < 1$, T_a tem um só ponto fixo que é o ponto 0.

O uso do comando **solve**, sem opções suplementares, para determinar os pontos fixos de T_a implica a indicação prévia do parâmetro com que estamos a trabalhar.

Tendo definida a família de aplicações, para determinar os pontos fixos para um parâmetro a, por exemplo $a = 1.5$, basta escrever:

```
> a:=1.5:

> solve(T(a,x)=x,x);
```

$$0., 0.6000000000$$

Note-se que neste caso, o valor de a é manipulado internamente pelo Maple, como sendo um número do tipo **float**. Nestes casos, o Maple responde ao comando **solve** como se se tratasse do comando **fsolve**. Se em vez de escrevermos $a = 1.5$ escrevermos $a = \frac{3}{2}$ o resultado será dado como um quociente entre dois inteiros; neste caso $\frac{3}{5}$, que de facto vale 0.6.

```
> a:=3/2:

> solve(T(a,x)=x,x);
```

$$0, \frac{3}{5}$$

Comando fsolve

Como já referimos, também podemos usar o comando **fsolve** para determinar os pontos fixos de uma dada função. Depois de a função T_a estar definida, ao escrevermos:

```
> fsolve(T(a,x)=x,x);
```

Obtemos a mensagem de erro:

```
Error, (in fsolve) a is in the equation, and is not solved for
```

Para evitar este erro, temos de indicar o valor de a para o qual queremos resolver a equação, uma vez que este comando dá soluções numéricas e não algébricas.

Ao procedermos de modo análogo ao que fizemos para o comando **solve**, obteremos uma só solução, que não é o correcto. Por exemplo, para determinar os pontos fixos de $T_{1.5}$ obtemos:

```
> a:=1.5:

> fsolve(T(a,x)=x,x);
```

$$0.$$

Se indicarmos o intervalo onde queremos verificar se existe solução, por exemplo, no caso anterior, no intervalo $[0, 1]$, também obtemos uma resposta incompleta. Como já referimos na secção 1.8 quando as funções não são polinomiais, com o comando **fsolve** é apresentada só uma das soluções.

```
> a:=1.5:
> fsolve(T(a,x)=x,x=0..1);
```

$$0.6000000000$$

Para contornarmos este tipo de resposta e sabendo-se de antemão que há mais do que uma solução, podemos usar a opção **avoid** do comando **fsolve**. Esta opção permite que depois de conhecida uma solução, a equação seja de novo resolvida, mas de modo a que a solução já conhecida seja evitada. No nosso exemplo, como já sabemos que 0 é uma solução, podemos escrever:

```
> a:=1.5:
> fsolve(T(a,x)=x,x,avoid={x=0});
```

$$\{x = 0.6000000000\}$$

Parece-nos que o uso dos comandos **solve** e **fsolve** não traz vantagens para o aluno, devido às manipulações cuidadosas que eles exigem. Assim, o professor ao usar estes comandos deverá estar preparado para que o aluno tire conclusões erradas, perantes as respostas dadas pelo Maple.

3.3.2 Método gráfico

Na secção 2.3.1 apresentamos o procedimento pontosfixosplot que serve para determinar graficamente os pontos fixos de uma função f. Uma vez que a aplicação tenda depende de um parâmetro a, para determinarmos graficamente os seus pontos fixos temos de previamente fixar o parâmetro a e de definir uma função g tal que $g(x) = T(a, x)$. Se quisermos ver, por exemplo, os pontos fixos de $T_{1.2}$ escrevemos:

```
> a:=1.2:

> g:=x->T(a,x):
```

Chamando o procedimento **pontosfixosplot**

```
> pontosfixosplot(g,0,1):
```

Obtemos a figura 3.6.

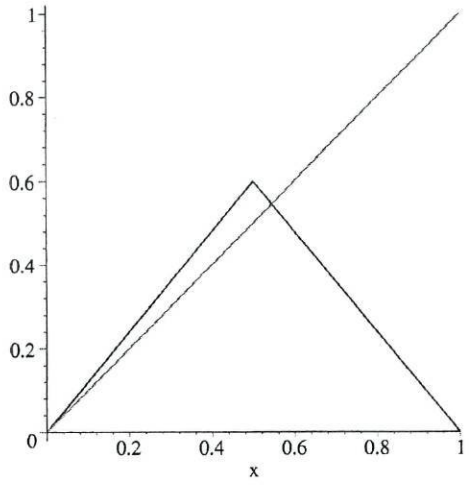


Figura 3.6: **pontosfixosplot** de $T_{1.2}$

Contudo, uma pequena modificação no procedimento **pontosfixosplot** permite-nos lidar com funções que dependem de um parâmetro a. Assim, criamos o procedimento pontosfixospmplot que a seguir se transcreve e que tem como parâmetros entrada:

- a função f que depende de duas variáveis x e a;
- o parâmetro a;
- os valores $xmin$ e $xmax$ que são os limites esquerdo e direito do intervalo;

```
> pontosfixospmplot:=proc(f,a,xmin,xmax)
  local x,id,p,q;
  with(plots):
  id:=x->x:
  q:=plot(id(x),x=xmin..xmax,color=green):
  p:=plot(f(a,x),x=xmin..xmax):
  display({p,q},title=cat('a=',convert(a,string)),font=[TIMES,ROMAN,14],
  titlefont=[TIMES,ROMAN,14],scaling=constrained,thickness=2);
  end proc:
```

Este procedimento pode ser usado com a aplicação tenda ou com outra que dependa de um parâmetro. Tal como o procedimento **pontosfixosplot**, dá-nos no mesmo sistema de eixos, a representação gráfica de uma função f e da identidade, permitindo-nos determinar graficamente os pontos fixos de f.

Para determinar os pontos fixos de $T_{1.2}$ com este procedimento, devemos escrever:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):

> pontosfixospmplot(T,1.2,0,1);
```

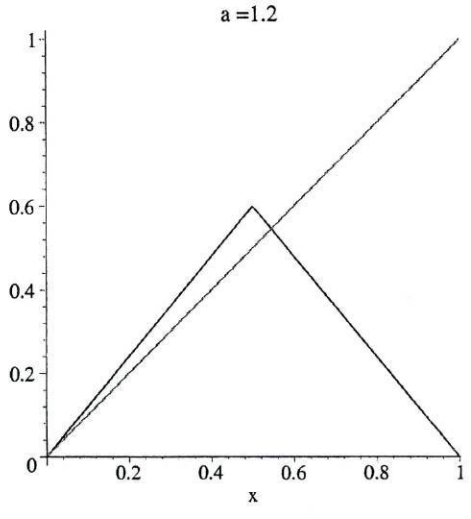


Figura 3.7: Pontos fixos de $T_{1.2}$

O Maple responde com o gráfico da figura 3.7. Podemos então verificar que existem dois pontos fixos: um que é o 0 e o outro que está no intervalo $[0.5, 0.6]$. Para visua-

lizarmos melhor o valor desse ponto fixo não nulo, executamos de novo o procedimento **pontosfixospmplot**, mas agora com $xmin = 0.5$ e $xmax = 0.6$.

```
> pontosfixospmplot(T,1.2,0.5,0.6);
```

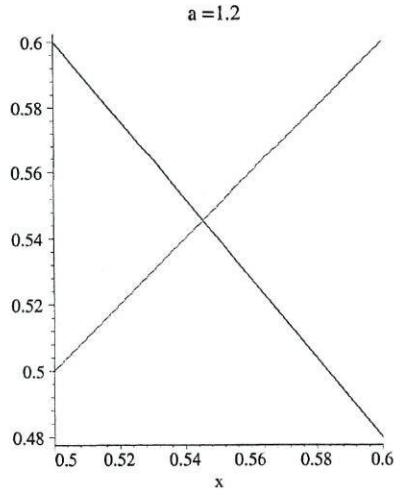


Figura 3.8: Maior precisão para o ponto fixo não nulo de $T_{1.2}$

Obtemos o gráfico da figura 3.8 que nos indica que o ponto fixo está no intervalo $[0.54, 0.56]$.

Até aqui só estudamos os pontos fixos da aplicação tenda para um caso particular de a. Como a aplicação tenda depende do parâmetro a, é natural que os seus pontos fixos também dependam desse parâmetro. A seguir vamos ver como se comportam os pontos fixos de T_a com a variação do parâmetro a. Para tal, criamos o procedimento animafixos que ilustra a variação dos pontos fixos de f para valores de a entre *amin* e *amax*, utilizando aos recursos de animação do Maple. Esta animação tem vantagens em relação ao procedimento **pontosfixospmplot**, uma vez que sugere a ideia de movimento e permite perceber melhor o que vai sucedendo aos pontos fixos de T_a com a variação de a.

Como parâmetros de entrada o procedimento tem: f, *amin*, *amax* e *npontos*, onde:

- f é a função em estudo;
- *amin* é o valor mínimo de a a considerar;

- *amax* é o valor máximo de a a considerar;

- *npontos* é o número de pontos em que se divide o intervalo $[amin, amax]$.

Para usar este procedimento é necessário definir a função com a qual estamos a trabalhar, e guardar numa variável, a que demos o nome de **timeplot**, o gráfico de f para cada valor de a considerado. Cada gráfico tem um título que varia de acordo com o valor de a considerado. O comando **convert(a, string)** é usado para que o valor de a, considerado em determinado instante, seja automaticamente encarado como um número quando aparece no título. Atribuímos a uma variável A o **display** do **timeplot**. Como opção usamos **insequence=true** para que apareça gráfico a gráfico e não todos ao mesmo tempo. Guardamos numa variável *Diag* o gráfico da diagonal e por fim fazemos um **display** das variáveis *Diag* e A.

```
> animafixos:=proc(f,amin,amax,npontos)
  localtimeplot,A,Diag,a:
  with(plots):
  timeplot:=NULL:
  for a from amin to amax by (amax-amin)/npontos do
  timeplot:=timeplot,plot(f(a,x),x=0..1,title=cat('a=',convert(a,string))):
  end do:
  A:=display([timeplot],insequence=true):
  Diag:=plot(x,x=0..1,color=BLUE):
  display({Diag,A},scaling=constrained,thickness=3);
  end proc:
```

Para vermos esta animação com a aplicação tenda e para a entre 0 e 2 com 200 pontos, temos de escrever os seguintes comandos:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):

> animafixos(T,0,2,200);
```

Uma vez que não é possível apresentar a animação na versão impressa deste trabalho, apresentamos na tabela 3.4 imagens intermédias dessa animação.

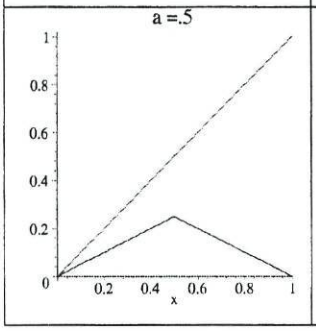
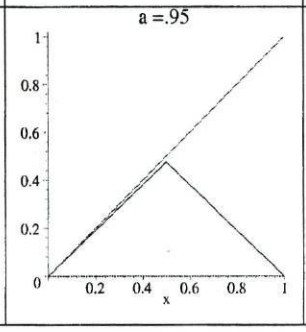
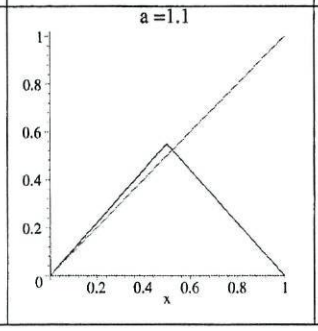
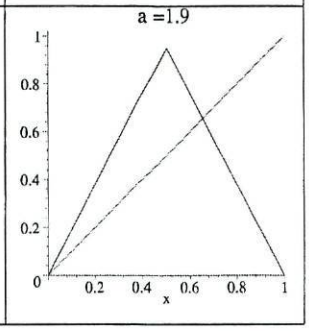
$a = 0.5$	$a = 0.95$	$a = 1.1$	$a = 1.9$
			

Tabela 3.4: Representação gráfica dos pontos fixos de T_a para diferentes valores de a.

Verificamos que:

- Se $0 < a < 1$, T_a tem um único ponto fixo que é $x = 0$.
- Quando $a = 1$ um dos ramos do gráfico da função coincide com a diagonal e portanto todos os pontos no intervalo $[0, \frac{1}{2}]$ são de período um.
- Se $1 < a \leq 2$ existem dois pontos fixos, que são $x = 0$ e $x = \frac{a}{1+a}$.

Mais uma vez podemos referir que para $a = 1$ se dá uma bifurcação no número de pontos fixos, ou seja, para $a < 1$ a função só tem um ponto fixo e para $a > 1$ a função passa a ter dois pontos fixos. Pode ainda referir-se de novo o uso da instrução

```
> assume(a>1):
```

dada na secção 3.3.1, uma vez que é de esperar que nesta fase do estudo o aluno esteja mais familiarizado com a aplicação e por conseguinte mais motivado para compreender o seu uso.

3.4 Período dois

Nesta secção vamos precisar de iterar a aplicação tenda e uma vez que esta depende de um parâmetro a e de uma variável x, criamos uma função g tal que fixado um parâmetro a, $g(x) = T(a, x)$.

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
```

```
> g:=x->T(a,x):
```

Na secção 2.3.2 explicamos como podemos recorrer aos comandos do Maple para determinar os pontos periódicos de período dois de uma função. Nesta secção, vamos usar os processos aí descritos para tentar determinar os pontos periódicos de período dois da aplicação tenda. Tal como fizemos para os pontos fixos, na secção anterior, vamos começar pelos métodos analíticos para terminar com o método gráfico.

3.4.1 Métodos analíticos

Para determinar analiticamente os pontos periódicos de período dois podemos recorrer tanto ao comando **solve** como ao comando **fsolve**.

Comando solve

Recorrendo ao procedimento perdois apresentado na secção 2.3.2 vamos tentar encontrar os pontos periódicos de período dois da tenda. Como parâmetros de entrada, este procedimento tem unicamente a função f em estudo e como resposta dá-nos a lista dos pontos periódicos de período dois de f.

Ao usarmos este procedimento sem dar qualquer indicação acerca do parâmetro a, obtemos como resposta uma lista vazia.

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> g:=x->T(a,x):
> perdois(g);
                                []
```

Isto não significa que a aplicação não tem pontos periódicos de período dois. Como já referimos na secção 3.3.1 tal acontece porque dependendo do valor do parâmetro a considerado, o número de raízes das equações $T_a(x) = x$ e $T_a^2(x) = x$ varia e o Maple não consegue manipular simultaneamente as duas situações.

Vamos usar a opção **assume** do comando **solve** e ver se deste modo o procedimento já nos dá uma resposta que esteja de acordo com o que realmente se passa.

Para determinarmos os pontos periódicos de T_a com $a < 1$, temos de escrever:

```
> assume(a<1):
> perdois(g);
```

Obtemos uma lista vazia

$$[]$$

o que quer dizer que neste caso é apresentada a resposta esperada: com $a < 1$, T_a não tem pontos periódicos de período dois.

Para determinar os pontos de T_a com $a > 1$, devemos escrever:

```
> assume(a>1):
> perdois(g);
```

Com o uso destes comandos, o Maple fornece uma lista com dois elementos que são, de facto, os pontos periódicos de período dois para $a > 1$.

$$[\frac{a^{\sim 2}}{a^{\sim 2}+1}, \frac{a^{\sim}}{a^{\sim 2}+1}]$$

Para determinar os pontos periódicos de T_a sem recorrer ao comando **assume** tem de se dar a indicação prévia do parâmetro a com que estamos a trabalhar. Tendo definida a família de aplicações, para determinar os pontos periódicos com o procedimento **perdois**, para um parâmetro a, por exemplo $a = \frac{3}{2}$, basta escrever:

```
> a:=3/2:
> perdois(g);
```

$$[\frac{6}{13}, \frac{9}{13}]$$

Sem recorrer ao procedimento **perdois** são simples as instruções necessárias para se chegar à expressão geral dos pontos periódicos de período dois. Para um parâmetro a, recorrendo aos comandos:

```
> assume(a>1):
> solve(g(x)=x,x);
```

$$0, \frac{a^{\sim}}{a^{\sim}+1}$$

```
> solve((g@@2)(x)=x,x);
```

Obtemos os pontos fixos de $T_a^2(x)$:

$$0, \frac{a^{\sim 2}}{a^{\sim 2}+1}, \frac{a^{\sim}}{a^{\sim 2}+1}, \frac{a^{\sim}}{a^{\sim}+1}$$

Uma vez que os pontos periódicos de período dois de uma função são aqueles que são solução da equação $f^2(x) = x$ e que não são solução da equação $f(x) = x$, podemos concluir que os valores presentes nesta lista e que não estão na lista anterior, a saber $\frac{a^2}{a^2+1}$ e $\frac{a}{a^2+1}$ são os pontos periódicos de período dois da tenda para $a > 1$.

É claro que podemos usar este método para determinar os pontos periódicos de período dois para qualquer valor de a previamente fixado. No caso de $a = \frac{3}{2}$ escrevemos:

```
> a:=3/2:
> solve((g@@2)(x)=x,x);
```

$$0, \frac{9}{13}, \frac{6}{13}, \frac{3}{5}$$

```
> solve(g(x)=x,x);
```

$$0, \frac{3}{5}$$

É fácil ver que para este parâmetro, $a = \frac{3}{2}$, os pontos periódicos de período dois são: $\frac{9}{13}$ e $\frac{6}{13}$.

Para o caso em que $a < 1$, por exemplo $a = \frac{2}{3}$, também podemos verificar a não existência de pontos periódicos de período dois através deste método. Para isso, basta escrever:

```
> a:=2/3:
> solve((g@@2)(x)=x,x);
```

$$0$$

```
> solve(g(x)=x,x);
```

$$0$$

Comparando os valores obtidos nas duas listas, concluímos que T_a não tem pontos periódicos de período dois, pois 0 é ponto fixo de T_a.

Comando fsolve

Para determinarmos numericamente os pontos periódicos de período dois recorremos ao comando **fsolve**. Na secção 2.3.2 apresentamos o procedimento perdoisf que nos dá os pontos periódicos de período dois de uma função usando o comando **fsolve**. Uma vez que o comando **fsolve** só aceita valores numéricos, temos de começar por definir o valor do parâmetro a para o qual queremos estudar a aplicação. Se escolhermos por exemplo $a = 1.2$ e quisermos ver os resultados obtidos no intervalo $[0, 1]$ devemos escrever:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> g:=x->T(a,x):
> a:=1.2:
> perdoisf(g,0,1);
```

Como resposta obtemos:

$$[0.4918032787]$$

Esta resposta não é a verificada na realidade. Numa tentativa de perceber o que leva este procedimento a responder deste modo, vamos começar por determinar os zeros da equação $T_a(x) = x$ com $a = 1.2$.

```
> a:=1.2:
> fsolve(g(x)=x,x);
```

Na secção 3.3.2 concluímos que se $a > 1$ a função tem dois pontos fixos; portanto, neste caso em que $a = 1.2$ é esperado obter duas soluções para a equação $T_{1.2}(x) = x$. De facto, o que o Maple responde é:

$$0.$$

Este resultado deve-se ao uso do comando **fsolve** que, como já referimos na secção 1.8, nem sempre dá todas as soluções possíveis e de facto muitas vezes só dá uma solução.

Ao resolver a equação $T^2_{1.2}(x) = x$

```
> a:=1.2:
> fsolve((g@@2)(x)=x,x);
```

o Maple volta a responder apenas com a solução:

0.

Como sabemos, para $a > 1$ a aplicação tenda tem dois pontos periódicos de período dois, logo esta resposta não está completa.

Com base nestas observações parece-nos que o mais indicado é usar este procedimento somente para confirmar resultados já obtidos através de outros meios e mesmo neste caso ter extremo cuidado para não tirar conclusões erradas.

Para tentar superar as dificuldades inerentes ao uso do comando **fsolve**, podemos usar o comando

```
> fsolve(g(x)=x,x,avoid={x=0});
```

que permite obter mais uma das soluções da equação $T_{1.2}^2(x) = x$, para além da já conhecida:

0.5454545455

Deste modo conseguimos obter as duas soluções da equação $T_{1.2}(x) = x$.

O uso da opção **avoid** no comando **fsolve** não é o remédio para conseguirmos resolver numericamente a equação $T_{1.2}^2(x) = x$. Se recorrermos a esta opção do comando **fsolve** para tentar determinar as outras raízes da equação, conseguimos ainda determinar uma segunda raiz.

```
> fsolve((g@@2)(x)=x,x,avoid={x=0});
```

0.4918032787

Contudo, se usarmos de novo esta opção para obtermos mais uma das soluções e agora retirando as soluções $x = 0$ e $x = 0.4918032787$, obtemos um resultado que não nos permite retirar o valor pretendido, conforme se mostra na figura 3.9.

Mais uma vez ficamos limitados e sem hipótese para determinar as soluções da equação $T_{1.2}^2(x) = x$ só com esta opção do comando **fsolve**.

Como só estamos interessados nas raízes da equação $T_{1.2}^2(x) = x$ que estão no intervalo [0,1], podemos ainda tentar encontrar todos os pontos periódicos de T_a restringindo o intervalo no qual queremos que o Maple verifique se existe solução. Assim, se escrevermos

```
> fsolve((g@@2)(x)=x,x=0..1,avoid={x=0,x=.4918032787});
```

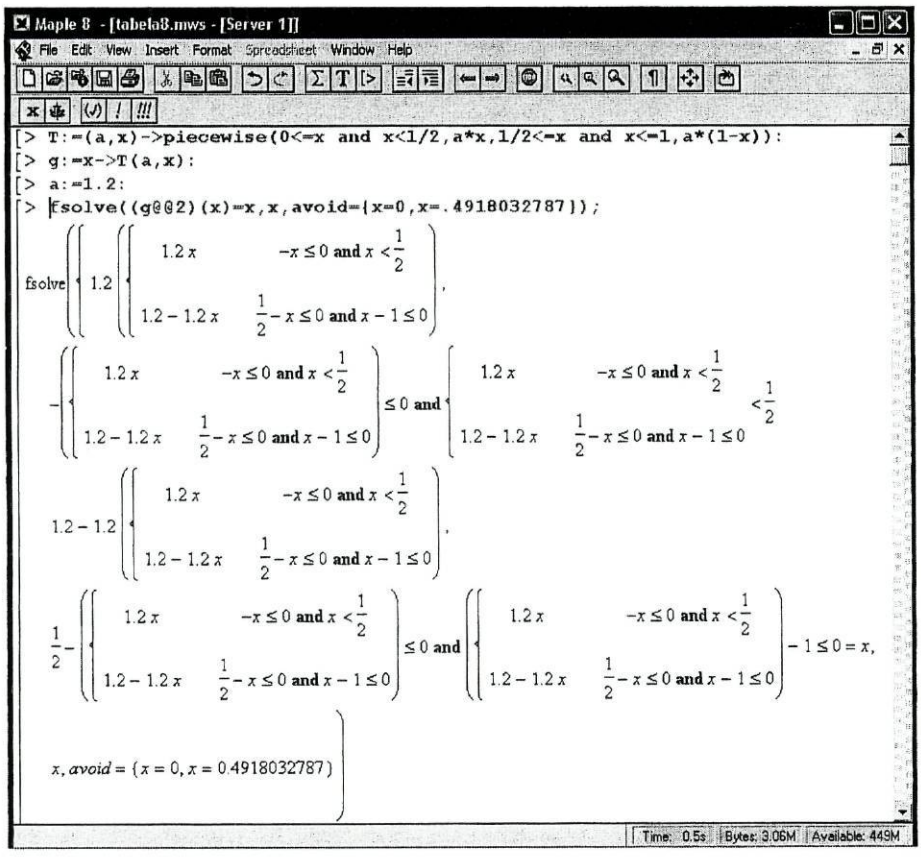


Figura 3.9: Resposta dada ao tentar determinar a terceira raíz da equação $T_{1.2}^2(x) = x$

Obtemos mais uma raiz da equação $T_{1.2}^2(x) = x$

$$0.5454545455$$

Se voltarmos a restringir o intervalo e evitarmos as raízes 0, 0.5454545455 e 0.4918032787

```
> fsolve((g@@2)(x)=x,x=0..1,avoid={x=0,x=.4918032787,x=.5454545455});
```

obtemos a quarta raiz da equação $T_{1.2}^2(x) = x$

$$0.5901639344$$

Uma vez que a equação não tem mais soluções, ao repetir o comando **avoid** seria de esperar que a resposta dada fosse uma lista vazia. Contudo, a resposta dada é a que se pode observar na figura 3.10 que não permite, por este meio, ao utilizador inferir se a equação tem ou não mais soluções.

Em conclusão, podemos dizer que os comandos **solve** e **fsolve** só devem ser usados

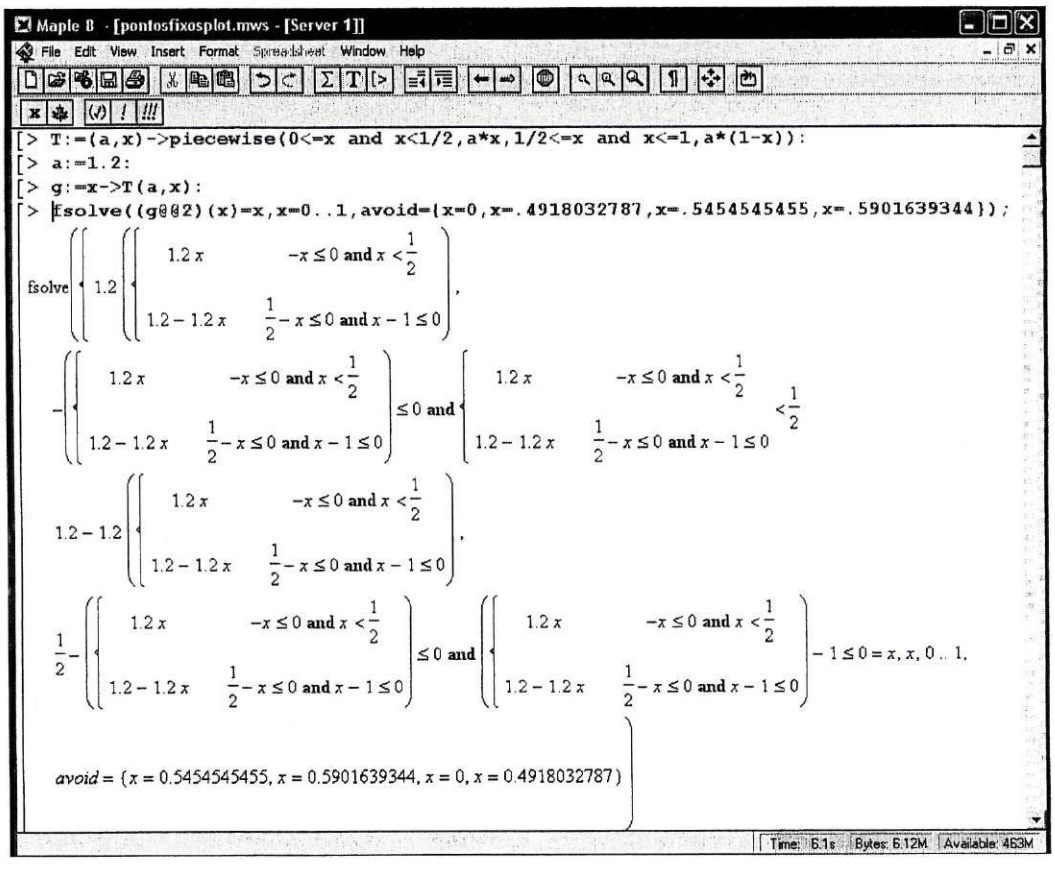


Figura 3.10: Resposta inesperada do comando **avoid**

para confirmar resultados, ou quando já temos informação, adquirida por outros meios, acerca dos resultados que devemos obter. Os métodos apresentados nesta secção devem ser evitados quando se tem a expectativa de obter os resultados correctos.

Na secção seguinte será apresentada a manipulação gráfica, que é aquela em que o Maple produz os resultados esperados e, de facto, aquela que tem mais impacto nos alunos.

3.4.2 Método gráfico

Para determinar graficamente os pontos periódicos de período dois da aplicação tenda vamos usar o procedimento perdoisplot descrito na secção 2.3.2. Como parâmetros de entrada tem: f, x_{min} e x_{max}, onde f é a função pretendida e $[x_{min}, x_{max}]$ o intervalo do seu domínio. Como resposta, o procedimento fornece-nos no mesmo sistema de eixos o

gráfico de f, de f^2 e da identidade.

Se escolhermos para o parâmetro a da família das aplicações tenda, por exemplo, o valor 1.5 e escrevendo as instruções:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> g:=x->T(a,x):
> a:=1.5:
> perdoisplot(g,0,1);
```

Obtemos o gráfico da figura 3.11 onde se consegue ver que $T_{1.5}^2(x)$ tem quatro pontos fixos.

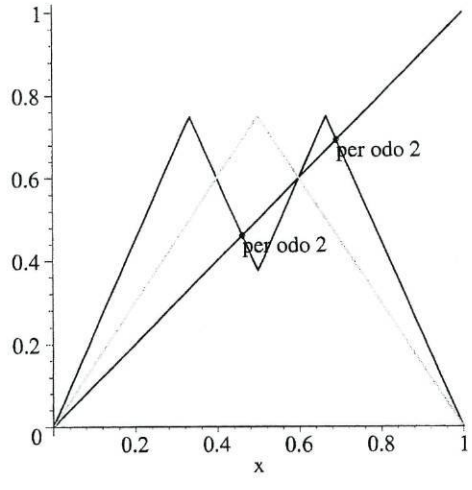


Figura 3.11: Pontos periódicos de período dois de $T_{1.2}$

É claro que dois desses pontos são os pontos fixos de $T_{1.5}$: 0 e 0.6 e os outros dois novos pontos são os pontos de uma órbita periódica de período dois: um deles está no intervalo $[0.4; 0.6]$ e o outro está no intervalo $[0.6; 0.8]$.

Se pretendermos saber com maior precisão o valor de um dos pontos periódicos, basta usar o procedimento **perdoisplot** com outros valores de *xmin* e *xmax*. Por exemplo, para obtermos o ponto periódico de período dois que está no intervalo $[0.4; 0.6]$, basta escrever:

```
> perdoisplot(g,0.4,0.6);#Escolhemos xmin=0.4 e xmax=0.6
```

O resultado obtido está exposto na figura 3.12.

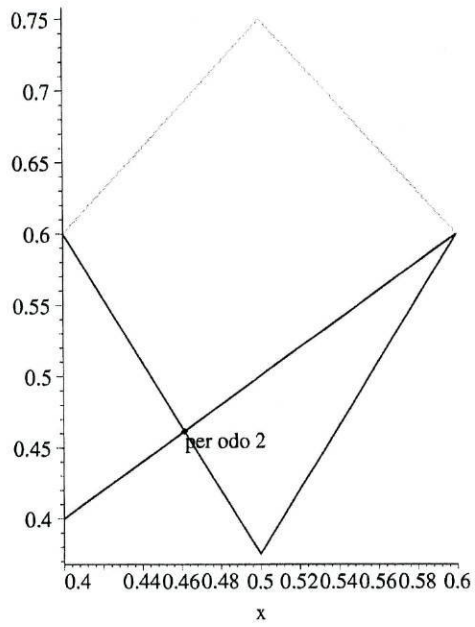


Figura 3.12: **perdoisplot** de $T_{1.2}(x)$ com $xmin = 0.4$ e $xmax = 0.6$

Podemos concluir que o ponto periódico de período dois está entre 0.44 e 0.48. Repetindo este processo poderíamos obter com a precisão pretendida quaisquer pontos periódicos de período dois.

De modo análogo ao que fizemos para os pontos fixos, podemos usar uma animação para estudar os pontos periódicos de período dois da aplicação tenda. Para tal podemos usar o procedimento perdoisanima que esboça os gráficos de f, f^2 e da identidade para uma função f que dependa de um parâmetro a. Como parâmetros de entrada este procedimento tem f, $amin$, $amax$ e $npontos$, em que:

- f é a aplicação a estudar;
- $amin$ é o valor mínimo de a;
- $amax$ é o valor máximo de a;
- $npontos$ é o número de pontos em que se divide $[amin, amax]$

```
> perdoisanima:=proc(f,amin,amax,npontos)
  local a,A,g,Diag,timeplot:
  with(plots):
  g:=x->f(a,x):
  timeplot:=NULL: for a from amin to amax by (amax-amin)/npontos do
  timeplot:=timeplot,plot({(g@@2)(x),g(x)},x=0..1,title=cat('a
  =',convert(a,string)),color=[black,yellow]):
  end do:
  A:=display([timeplot],insequence=true):
  Diag:=plot(x,x=0..1,color=black):
  display({Diag,A},scaling=constrained,thickness=2);
  end proc:
```

Este procedimento fornece-nos o gráfico de f a cor amarela e os gráficos de f^2 e da identidade a cor preta. Os pontos periódicos de período dois são aqueles em que só se intersectam os gráficos de cor preta e os pontos que resultam da intersecção dos gráficos representados a cor preta com o gráfico representado a amarelo são os pontos fixos de f.

Para se obter uma representação gráfica intermédia desta animação, temos de definir a função $g(x) = T(a, x)$, o parâmetro a para o qual queremos esboçar o gráfico e por fim usar o procedimento **perdoisanima**.

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> g:=x->T(a,x):
> perdoisanima(g,0,2,200);# Exemplo em que 0<a<2 e npontos=200
```

Na tabela 3.5 apresentamos alguns dessas representações intermédias, considerando $a = 0.5$, $a = 0.95$, $a = 1$, $a = 1.1$, $a = 1.9$ e $a = 2$.

Por estas ou por similares representações gráficas dos pontos periódicos de período dois, podemos fazer com que o aluno, baseado na tabela, conclua que:

- Se $0 \leq a \leq 1$, T_a não tem pontos periódicos de período dois, pois o único ponto onde os dois gráficos que estão representados com a cor preta se intersectam é o ponto 0 que é um ponto fixo de T_a.
- Se $1 < a \leq 2$, T_a tem dois pontos periódicos de período dois.

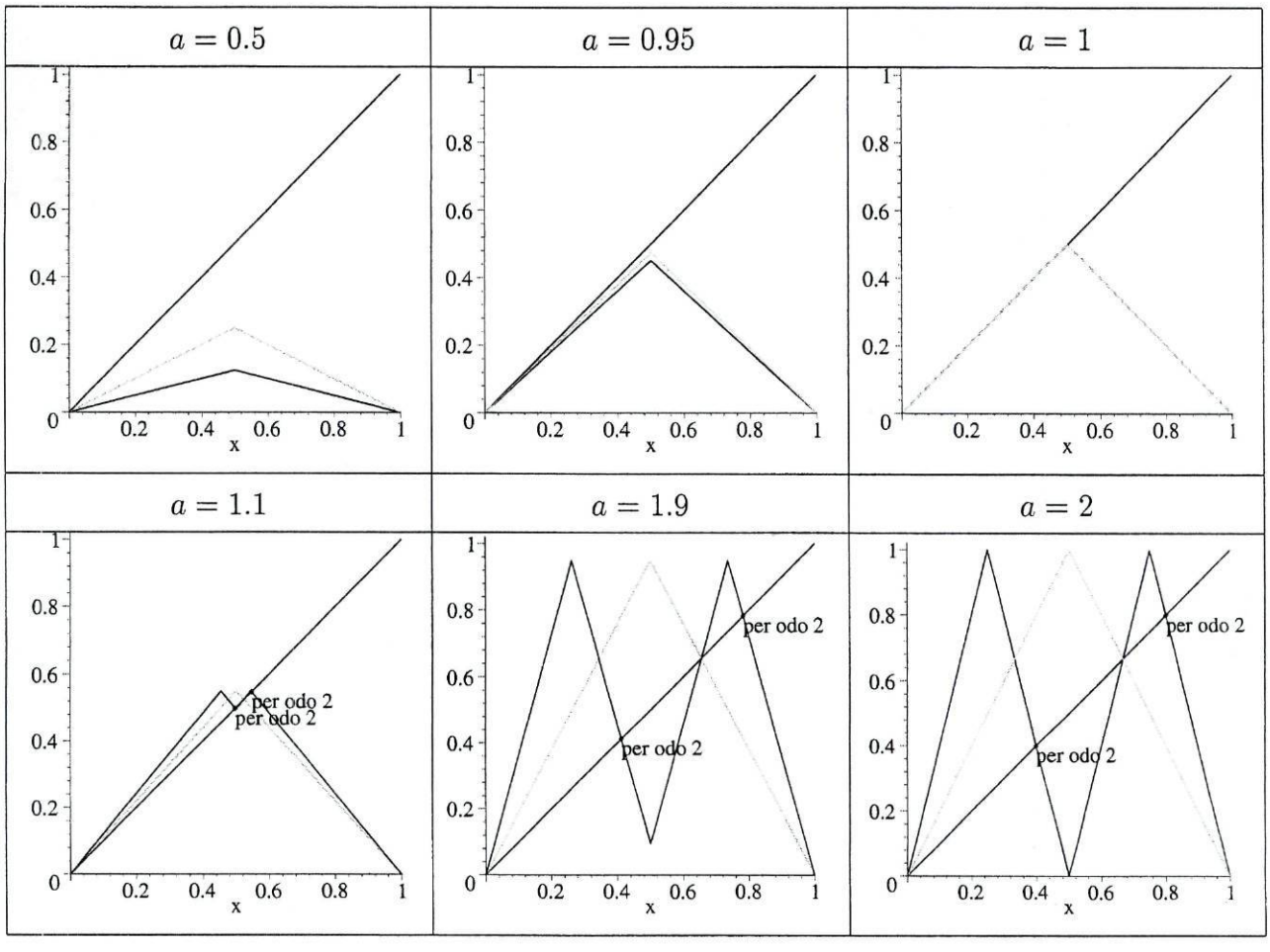


Tabela 3.5: Pontos periódicos de período dois de T_a para diferentes valores de a.

Contrariamente aos métodos analíticos, os métodos gráficos podem ser usados sem receio que apresentem resultados enganadores para o aluno.

3.5 Período três

Como seria de esperar, os métodos usados para determinar os pontos periódicos de período três não têm alterações significativas relativamente aos métodos usados na secção anterior para os pontos periódicos de período dois. Os métodos analíticos revelam-se pouco úteis e potenciais geradores de confusões no ensino enquanto que os métodos gráficos são bastante eficazes.

3.5.1 Métodos analíticos

A tentativa de determinar os pontos periódicos de período três conduz à resolução de equações mais complexas e portanto é natural que as respostas dadas pelos procedimentos analíticos apresentem ainda mais problemas do que para o período dois.

Comando solve

À semelhança do que fizemos para os pontos periódicos de período dois, vamos usar o procedimento pertres descrito na secção 2.3.2 e que usa o comando **solve** para determinar os pontos periódicos de período três.

Vamos começar por tentar determinar a expressão geral dos pontos periódicos de período três. Para que o Maple consiga manipular a equação $T_a^3(x) = x$ temos de indicar o ramo da função para o qual queremos resolver a equação. Como já vimos nas secções 3.3.1 e 3.4.1, para indicar que $a < 1$, temos de escrever:

```
> assume(a<1):
```

Para determinarmos os pontos periódicos de período três para estes valores de a, basta escrever:

```
> pertres(g);
```

e obtemos

□

O que corresponde ao esperado para $a < 1$.

Para determinarmos os pontos periódicos de período três de T_a para um valor de $a < 1$, por exemplo $a = \frac{2}{3}$, escrevemos:

```
> a:=2/3:

> pertres(g);
```

Obtemos uma lista vazia como já esperávamos.

□

Para estudarmos o caso em que $a > 1$, temos de começar por definir que $a > 1$ através do comando

```
> assume(a>1):
```

E, de modo análogo ao que se fez para o caso em que $a < 1$, escrever:

```
> pertres(g);
```

Como resposta obtemos uma lista vazia.

$$[]$$

O que não quer dizer que a aplicação não tem pontos periódicos de período três. Se pedirmos ao Maple que considere o parâmetro a com valores superiores a 1.8 conseguimos obter a expressão geral dos zeros da equação $T_a^3(x) = x$. Para isso temos de escrever

```
> assume(a>1.8):

> pertres(g);
```

Obtemos os pontos periódicos de período três de T_a para $a > 1.8$.

$$[\frac{a^{\sim}}{a^{\sim 3}+1}, \frac{a^{\sim 3}}{a^{\sim 3}+1}, \frac{a^{\sim 2}}{a^{\sim 3}+1}, \frac{a^{\sim}}{a^{\sim 2}+a^{\sim}+1}, \frac{(a^{\sim}+1)\,a^{\sim}}{a^{\sim 2}+a^{\sim}+1}, \frac{a^{\sim 2}}{a^{\sim 2}+a^{\sim}+1}].$$

Estando definida a família de aplicações e fixando um parâmetro a, ainda é possível determinar os pontos periódicos de período três. Por exemplo para determinar os pontos periódicos de período três para $a = \frac{19}{10}$, temos de escrever:

```
> a:=19/10:

> pertres(g);
```

e obtemos:

$$[\frac{3610}{7859}, \frac{6859}{7859}, \frac{361}{651}, \frac{1900}{7859}, \frac{190}{651}, \frac{551}{651}]$$

Depois de sabermos que há duas órbitas periódicas de período três, recorrendo ao procedimento listaorbita é possível confirmar que, de facto, estes são os pontos periódicos de período três de $T_{\frac{19}{10}}$. Ao determinar os três primeiros iterados da órbita de $\frac{3610}{7859}$ por $T_{\frac{19}{10}}$, através do comando

```
> listaorbita(T,19/10,0,4);
```

obtemos

$$[\frac{3610}{7859}, \frac{6859}{7859}, \frac{1900}{7859}, \frac{3610}{7859}]$$

o que permite ao aluno concluir que $\frac{3610}{7859}$ pertence à órbita periódica de período três ($\{\frac{3610}{7859}, \frac{6859}{7859}, \frac{1900}{7859}\}$) para $T_{\frac{19}{10}}$. Analogamente podemos apresentar os cálculos que permitam

ao aluno concluir que $\frac{361}{651}$ pertence à órbita periódica de período três ($\{\frac{361}{651}, \frac{551}{651}, \frac{190}{651}\}$) para $T_{\frac{19}{10}}$. Ao escrevermos

```
> listaorbita(T,19/10,0,4);
```

Obtemos a órbita de $\frac{361}{651}$ por $T_{\frac{19}{10}}$.

$$[\frac{361}{651}, \frac{551}{651}, \frac{190}{651}, \frac{361}{651}]$$

De qualquer modo, este método de determinar os pontos periódicos de período três não parece um método simples para ser apresentado a um aluno do Ensino Secundário, requerendo um esforço suplementar para compreender as respostas obtidas no Maple.

Comando fsolve

Como já referimos na secção 3.4.1 o uso deste comando para determinar pontos periódicos deixa de ser de grande utilidade, uma vez que com os métodos fornecidos não é possível exibi-los na totalidade. Se usarmos o procedimento pertresf para determinar os pontos periódicos de período três para a tenda, o resultado obtido é semelhante ao que se obtém para determinar os pontos periódicos de período dois com o procedimento **perdoisf**, ou seja, obtém-se uma lista vazia. Com o parâmetro $a = 1.9$, por exemplo, escrevemos:

```
> a:=1.9:
> pertresf(g);
```

e obtemos:

$$[]$$

Para esta aplicação este procedimento parece não ser de grande utilidade e se quisermos usar o comando **fsolve** para tentar determinar os pontos periódicos de período três de T_a para um parâmetro a previamente fixado temos de recorrer à opção **avoid**. Na figura 3.13 exibimos a folha de trabalho do Maple onde usamos o comando **fsolve** para tentar determinar os pontos fixos de $T_a^3(x)$ com $a = 1.9$.

Se não indicarmos o intervalo onde queremos que as raízes sejam encontradas, o uso

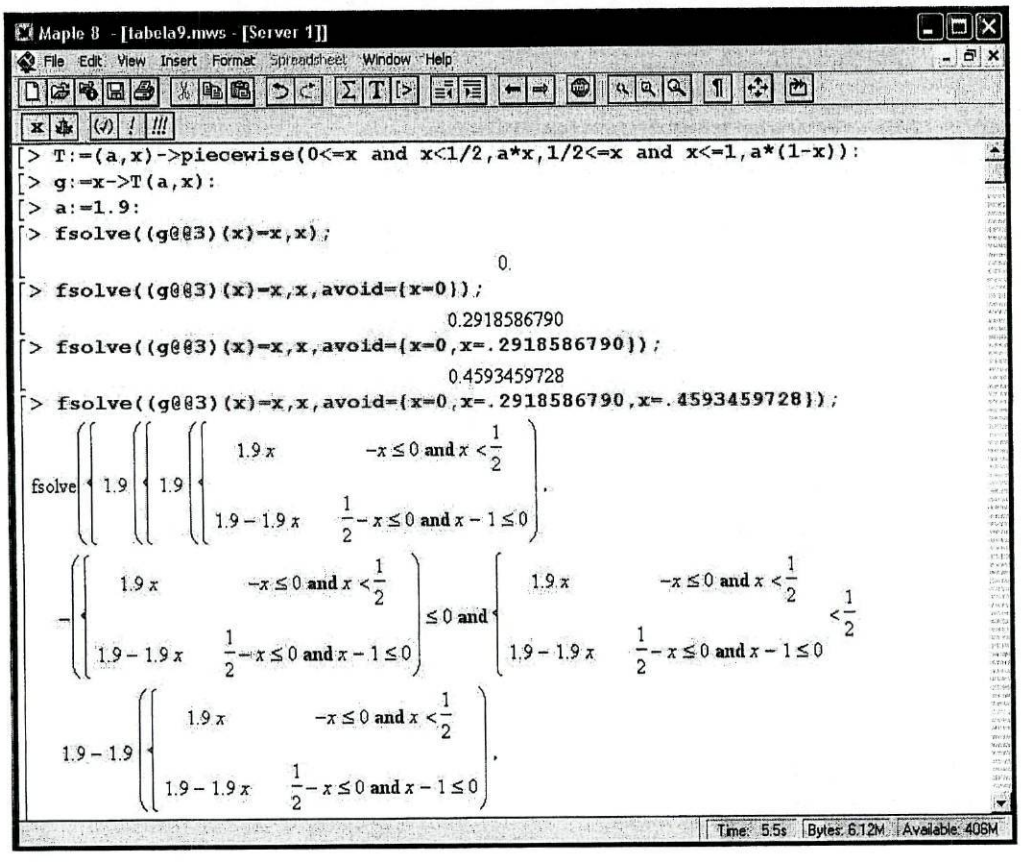


Figura 3.13: Folha de trabalho do Maple-resposta obtida com o comando **avoid**

da opção **avoid** do comando **fsolve** para determinar as raízes da equação $T_{1.9}^3(x) = x$ não é eficaz como podemos observar na figura 3.13. Tal como já fizemos para os pontos periódicos de período dois, se limitarmos o intervalo onde queremos que o Maple verifique se há solução para $[0, 1]$:

```
> fsolve((g@@3)(x)=x,x=0..1,avoid={x=0,x=.2918586790,x=.4593459728});
```

Obtemos uma outra solução da equação.

$$0.2417610383$$

Se voltarmos a usar este último comando mas agora evitando as raízes 0, 0.2918586790, 0.4593459728 e 0.2417610383:

```
> fsolve((g@@3)(x)=x,x=0..1,avoid={x=0,x=.2918586790,x=.4593459728,
  x=.2417610383});
```

obtemos como resposta mais uma das soluções.

0.8463901690

Se o Maple fosse sempre capaz de encontrar todos os valores numéricos das soluções da equação $T_a^3(x) = x$ e apesar de este ser um método pouco prático, uma vez que só se consegue determinar uma solução de cada vez, poder-se-ía optar por ele para determinar numericamente os pontos periódicos de período três de T_a. Tal não acontece. Se voltarmos a usar a opção **avoid**, para determinar o sexta raíz da equação $T_a^3(x) = x$ voltamos a obter uma resposta da do género obtida na tabela 3.13, o que não facilita a determinação das raízes. Contudo, conjugando este método com o procedimento **listaorbita** é possível chegar ao valor das órbitas periódicas. No entanto, este processo não é de aplicar na sala de aula.

3.5.2 Método gráfico

Para determinar graficamente os pontos periódicos de período três da aplicação tenda vamos usar o procedimento pertresplot descrito na secção 2.3.3. Como parâmetros de entrada tem: f, x_{min} e x_{max}, onde f é a função pretendida e $[x_{min}, x_{max}]$ o intervalo do seu domínio. Como resposta, o procedimento fornece-nos no mesmo sistema de eixos o gráfico de f, de f^3 e da identidade.

Se escolhermos, por exemplo, o valor 1.9 para o parâmetro a, devemos escrever:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):

> g:=x->T(a,x):

> a:=1.9:

> pertresplot(g,0,1);
```

Com estas instruções, obtemos o gráfico da figura 3.14.

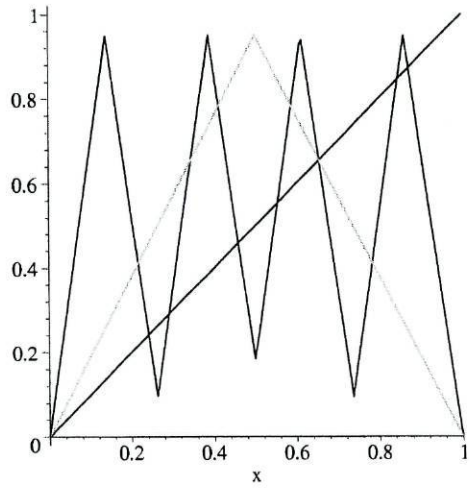


Figura 3.14: Pontos periódicos de período três de $T_{1.9}(x)$

Como já vimos na secção anterior o comando **fsolve** não responde adequadamente ao pedido de resolução da equação $T^3_{1.9}(x) = x$ pelo que na figura 3.14 não são indicados os pontos periódicos de período três. De qualquer modo como o gráfico da identidade e o de T^3_a estão representados a cor preta e o gráfico de T_a está representado a cor amarela é fácil para um aluno concluir que existem seis pontos no gráfico onde só se intersectam os gráficos de T^3_a e da identidade, ou seja, T_a tem seis pontos periódicos de período três (duas órbitas periódicas de período três).

Para os pontos periódicos de período três também criamos uma animação que permite estudar a sua variação com o parâmetro a. Com o procedimento pertresanima, que recorre a uma animação, podemos visualizar os pontos periódicos de período três da tenda com a variação de a. Como parâmetros de entrada este procedimento tem f, *amin*, *amax* e *npontos*, em que:

- f é a aplicação tenda;
- *amin* é o valor mínimo de a a considerar;
- *amax* é o valor máximo de a a considerar;
- *npontos* é o número de pontos em que se divide $[amin, amax]$

A única diferença entre este procedimento e o **perdoisanima** reside no facto de na instrução

```
> timeplot:=timeplot,plot((g@@2)(x),g(x),x=0..1,title=cat('a=',
  convert(a,string)),color=[black,yellow]):
```

substituirmos

```
> (g@@2)(x)
```

por

```
> (g@@3)(x)
```

Tal como fizemos para os pontos periódicos de período dois, na figura 3.6 apresentamos imagens intermédias da animação produzidas por **pertresanima**.

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):

> g:=x->T(a,x):

> pertresanima(g,0,2,200);# exemplo em que 0<a<2 e npontos=200
```

Baseado na animação e nesta tabela, ou numa outra equivalente, o aluno facilmente conclui que para valores de a próximos de 1.6, a função passa a ter pontos periódicos de período três. Para precisar melhor o valor de a para o qual tal acontece podemos usar a animação com diferentes enquadramentos para o parâmetro a. Através da tabela da figura 3.7 o aluno conclui que para a próximo de 1.625 a função T_a passa a ter pontos periódicos de período três.

Com a ajuda da animação e das tabelas das figuras 3.6 e 3.7 poderá ainda observar que para $a > 1.6$, T_a passa a ter, de facto, duas órbitas periódicas de período três.

Mais uma vez o método gráfico revela ser aquele que pode ser mais útil a conduzir o aluno a inferir os resultados pretendidos.

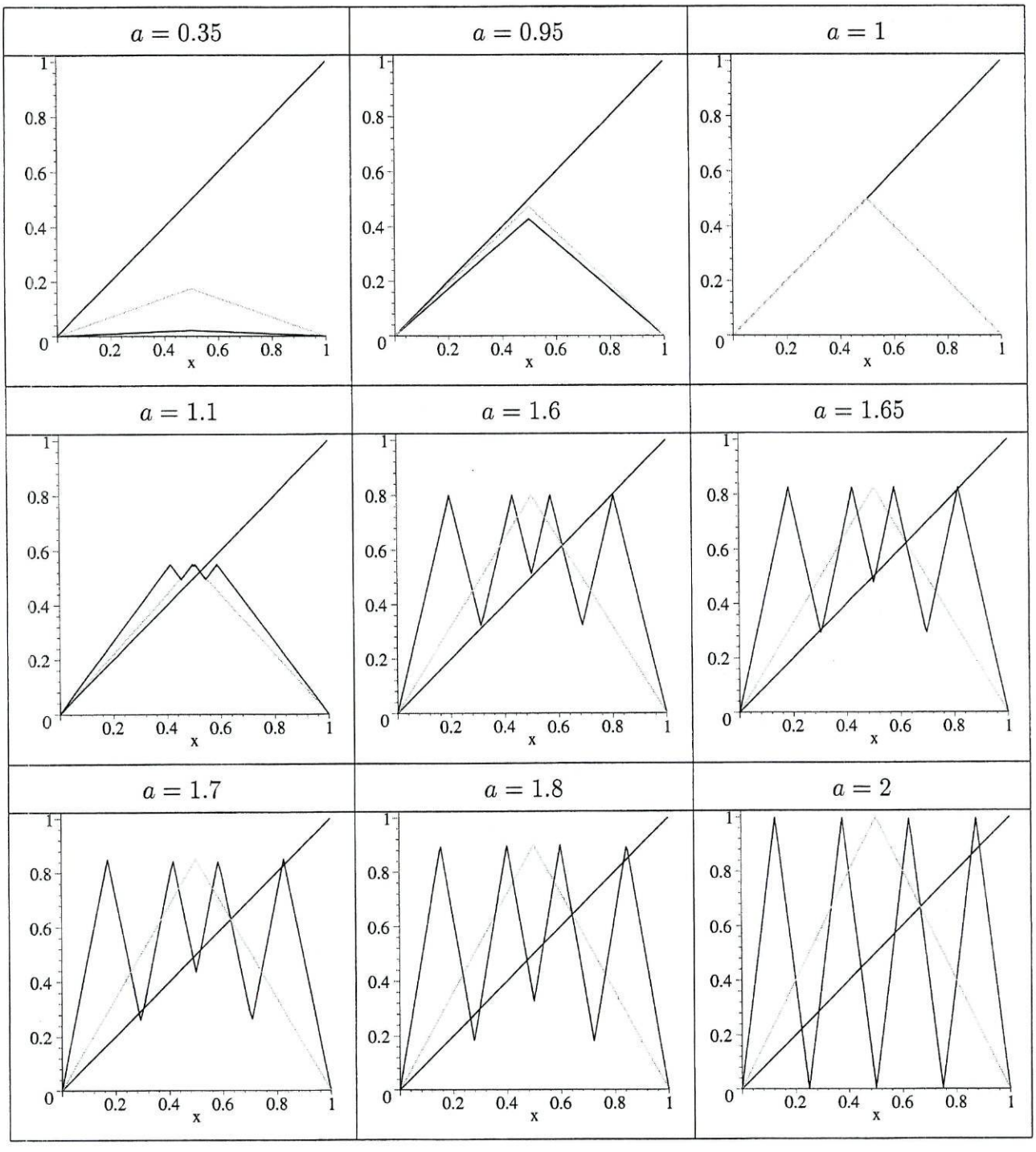


Tabela 3.6: Representação gráfica dos pontos periódicos de período três de T_a para diferentes valores de a.

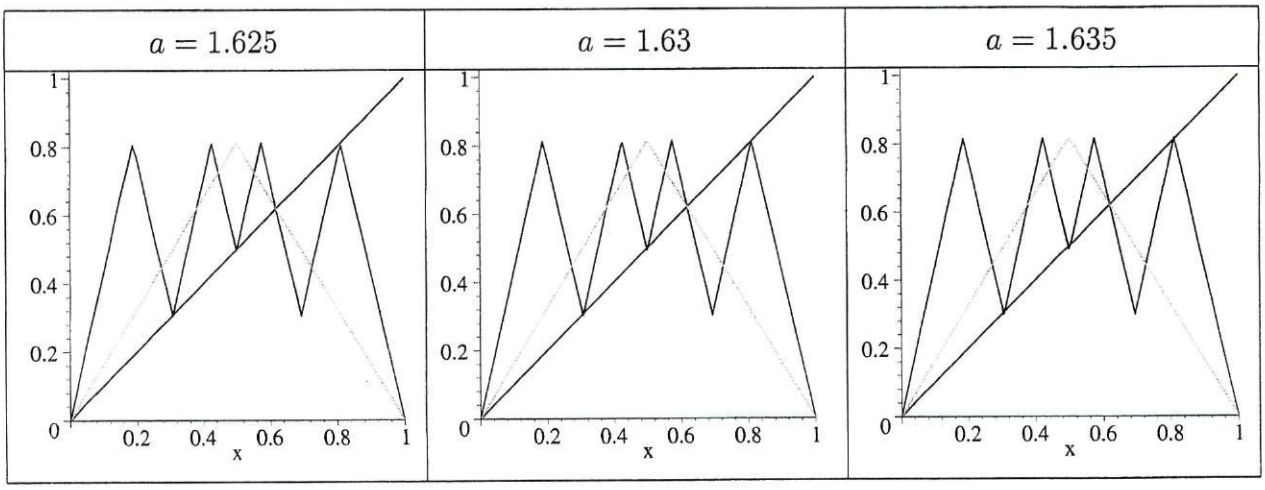


Tabela 3.7: Representação gráfica de órbitas obtidas com o procedimento **pertresanima**

3.6 Período arbitrário

Nesta fase do estudo dos pontos periódicos da aplicação tenda parece-nos que qualquer professor entenderá que os métodos analíticos usados anteriormente se tornam impraticáveis, no contexto do Ensino Secundário, para a determinação dos pontos periódicos de período maior do que três desta aplicação e que o uso de um procedimento semelhante ao **pertres** para os determinar não é uma boa opção.

Para determinar graficamente os pontos periódicos de período n podemos recorrer ao procedimento pernplot que como parâmetros de entrada tem a função f, o período pretendido n e os valores entre os quais se pretende visualizar o eixo dos xx: $xmin$ e $xmax$. Para determinar graficamente os pontos periódicos de período seis da aplicação tenda para $a = 1.4$, devemos escrever

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):

> g:=x->T(a,x):

> a:=1.4:

> pernplot(g,6,0,1);
```

e, obtemos o gráfico da figura 3.15

Os gráficos de $T_{1.4}^6$ e da identidade estão representados com a cor preta e os gráficos de $T_{1.4}$, $T_{1.4}^2$ e de $T_{1.4}^3$ estão representados com a cor amarela. Uma vez que no gráfico

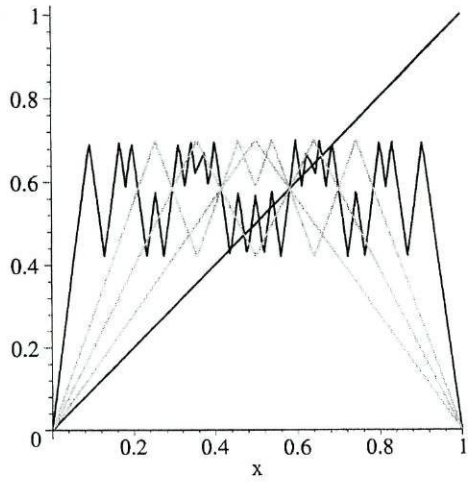


Figura 3.15: Pontos periódicos de período seis de $T_{1.4}(x)$

da figura 3.15 existem doze pontos que resultam da intersecção de gráficos que só estão representados a cor preta, o aluno pode concluir que $T_{1.4}$ tem duas órbitas periódicas de período seis, ou seja, tem doze pontos periódicos de período seis. Para determinarmos com maior exactidão a localização no intervalo $[0, 1]$ de quaisquer um desses pontos periódicos, basta usar o procedimento **pernplot** com diferentes valores para *xmin* e *xmax*. Se quisermos visualizar com maior pormenor o intervalo $[0.4; 0.6]$, devemos escrever:

```
> pernplot(g,6,0.4,0.6);
```

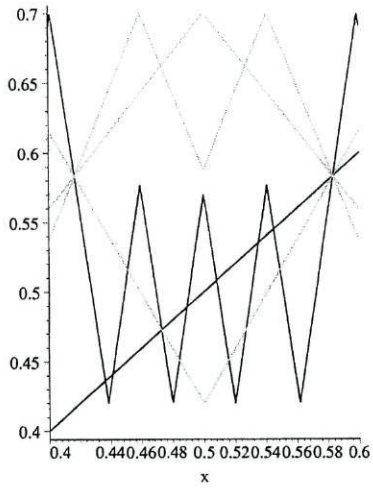


Figura 3.16: Pontos periódicos de período seis de $T_{1.4}(x)$ no intervalo $[0.4; 0.6]$

Para determinarmos com maior precisão, por exemplo, os pontos periódicos que se encontram no intervalo $[0.43; 0.45]$, escrevemos

```
> pernplot(g,6,0.43,0.45);
```

e obtemos o gráfico da figura 3.17.

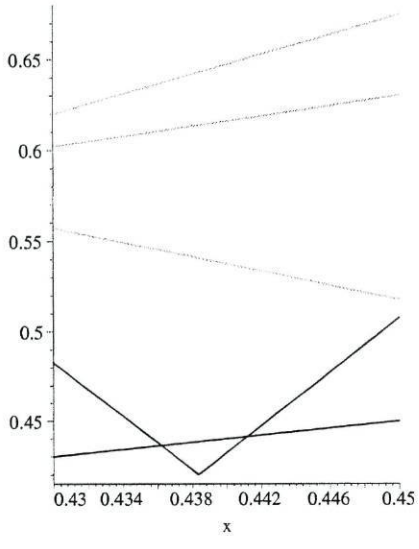


Figura 3.17: Pontos periódicos de período seis de $T_{1.4}(x)$ no intervalo $[0.43; 0.45]$

O aluno pode concluir agora, que um dos pontos periódicos de período seis está no intervalo $[0.434; 0.438]$ e o outro ponto periódico está no intervalo $[0.438; 0.442]$.

Mais uma vez acentuamos que o método gráfico é o mais fiável [2] e por isso o mais indicado para analisar órbitas na sala de aula.

3.7 Análise estatística de órbitas

Na secção 3.2.4, nos gráficos da tabela 3.3, vimos que a aplicação tenda, para certos valores do parâmetro a parece exibir um comportamento não periódico. Como já referimos na secção 2.4 nestes casos, em que o esboço gráfico de uma órbita não nos transmite informações suficientes acerca do seu comportamento, podemos recorrer a um diagrama de frequências e a outros utilitários estatísticos.

[2] Para uma manipulação elementar no Maple

3.7.1 Diagrama de frequências

Nesta secção vamos recorrer ao procedimento diagramafreqs descrito na secção 2.4 para analisarmos estatisticamente órbitas para diferentes valores do parâmetro da família das aplicações tenda. Este procedimento permite identificar numericamente regiões do intervalo que sejam mais significativas na evolução do sistema, ou seja, que correspondam a passagens mais frequentes da órbita. Como parâmetros de entrada temos:

- a função f (que se supõe definida no intervalo $[a, b]$);
- a que é o extremo esquerdo do intervalo;
- b que é o extremo direito do intervalo ;
- N que é o número pontos em que se divide o intervalo $[a, b]$, ;
- x_0 que é o ponto inicial;
- n que é o número de iterados.

Executando o comando

```
> diagramafreqs(f,a,b,N,x0,n);
```

obtém-se um histograma referente à amostra do pedaço de órbita atá ao instante n. Mais precisamente, obtemos um gráfico de barras em que a área de cada barra representa a frequência de visitas da órbita ao subintervalo determinado pela base do referido rectângulo.

O professor pode pedir ao aluno que use este procedimento para obter os diagramas de frequências para diferentes valores do parâmetro a e do ponto inicial x_0. Na tabela 3.8 exibimos alguns dos diagramas de frequências que o aluno pode obter. Os diagramas desta tabela são os obtidos para diferentes valores do parâmetro a entre 0 e 2, no intervalo $[0, 1]$ (com 100 subintervalos), para $x_0 = \frac{1}{3}$ e para 50000 iterações. Para obtermos cada um deles, devemos escrever as seguintes instruções:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> f:=x->T(a,x):
```

```
> a:=#substituir pelo parâmetro pretendido

> diagramafreqs(f,0,1,100,1/3,50000);
```

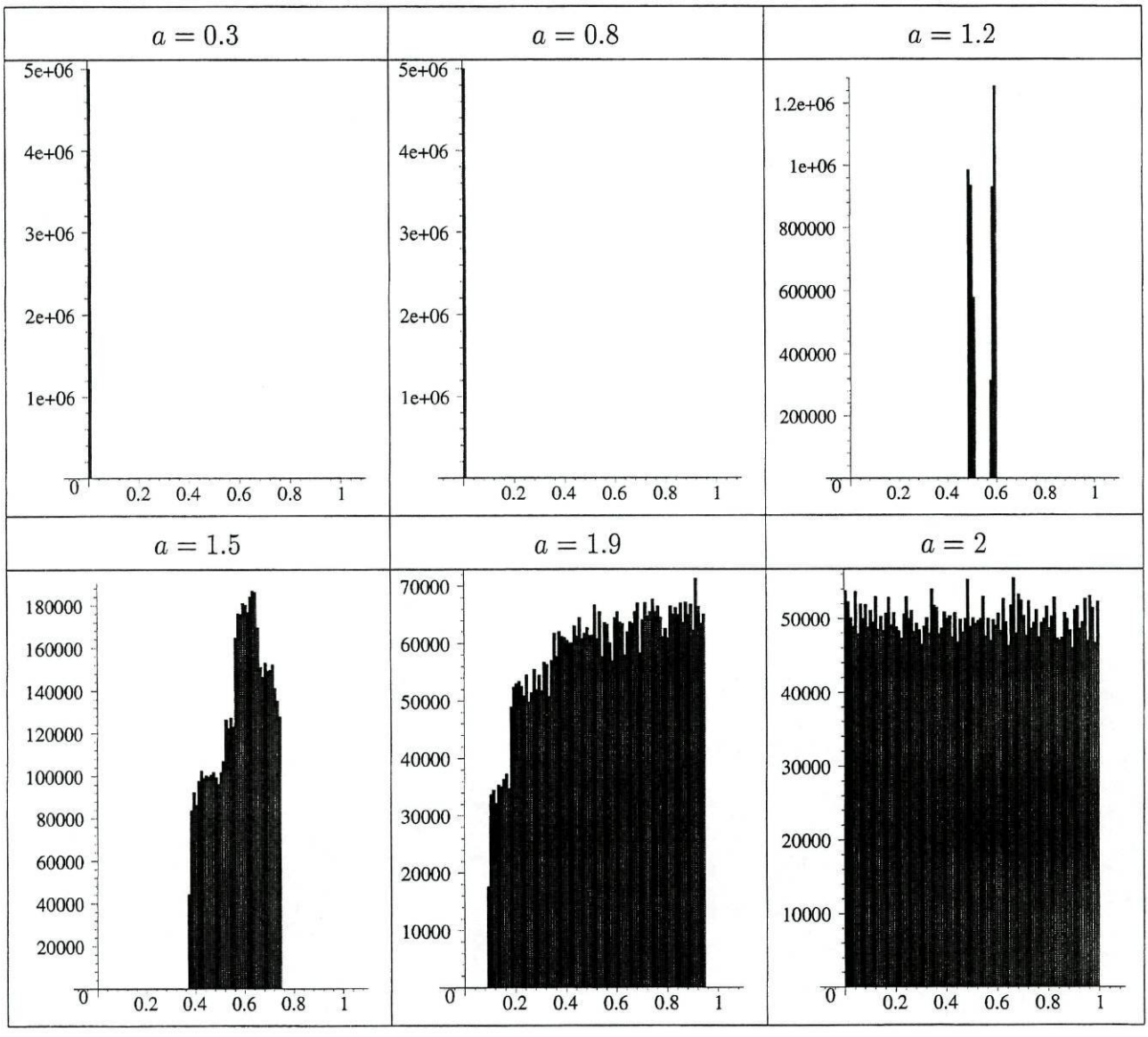


Tabela 3.8: Diagramas de frequências para T_a com $x_0 = \frac{1}{3}$

Com várias experimentações, o aluno poderá ser confrontado com uma grande variedade de situações, que a um nível elementar nem sempre são fáceis de responder. Ao analisar os resultados obtidos, o aluno pode conjecturar que à medida que o valor do parâmetro a aumenta a órbita distribui-se de uma forma mais uniforme pelo intervalo $[0, 1]$. Para $a = 2$ pode verificar que a frequência é constante para todos os subintervalos

considerados.

A seguir, o professor pode pedir ao aluno que fixe um valor para o parâmetro a e que ao variar o ponto inicial, tire conclusões. Fixando, por exemplo, o parâmetro $a = 1.5$, vamos ver os diagramas de frequências obtidos com $x_0 = 0.1$, $x_0 = 0.4$, $x_0 = 0.4$ e $x_0 = 0.8$. Os resultados estão expressos na tabela 3.9. Com os dados desta tabela, o aluno pode

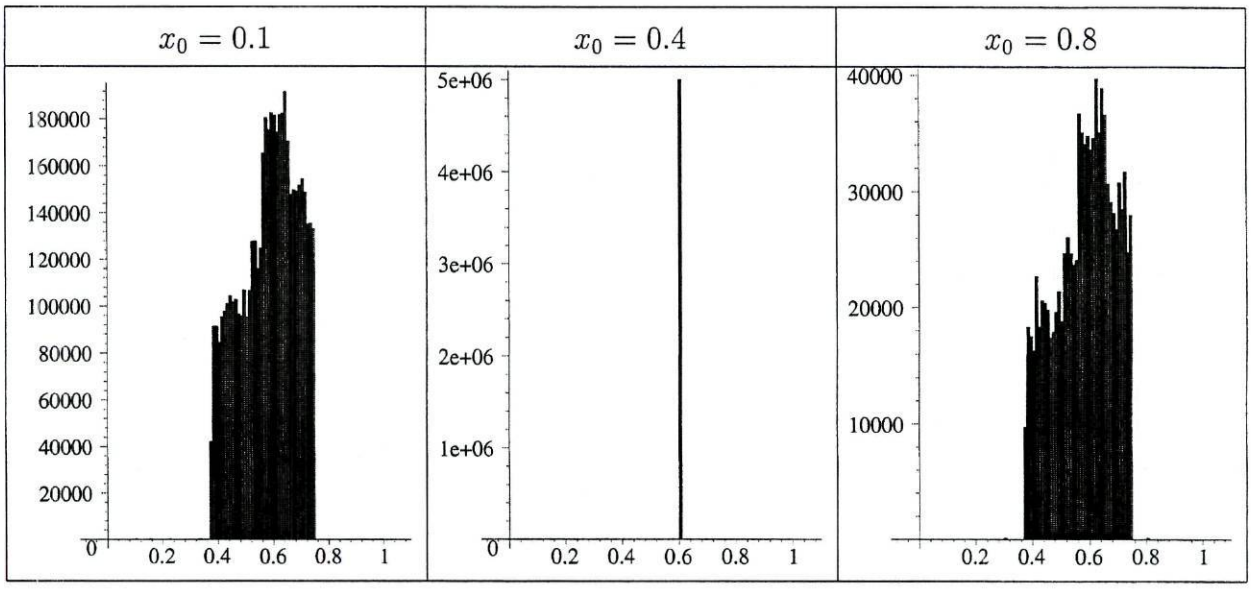


Tabela 3.9: Diagramas de frequência de $T_{1.5}$ para diferentes valores de x_0

concluir que aspecto final do diagrama não depende do valor inicial considerado. No caso de $x_0 = 0.4$ obteve-se uma só coluna porque 0.4 é enviado por $T_{1.5}$ em 0.6 e este é um ponto fixo de $T_{1.5}$. Para verificar esta última afirmação, o aluno pode recorrer ao procedimento **listaorbita**. Escrevendo:

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> f:=x->T(a,x):
> a:=1.5:
> listaorbita(f,0.4,0,5);
```

Obtemos os cinco primeiros iterados da órbita de $f_{1.5}(0.4)$.

[0.4, 0.60, 0.600, 0.6000, 0.60000, 0.60000]

Este resultado confirma, que de facto 0.6 é um ponto fixo de $T_{1.5}$.

3.7.2 Funções teste

Considerando $h(x) = ln|T'_a(x)|$, podemos proceder ao estudo da soma de Birkhoff associada à função h. Neste caso, como referido na secção 2.4.2, obtemos os conhecidos expoentes de Lyapunov. Cada um desses expoentes (L) é definido por

$$L = \lim_{n\to\infty} \frac{1}{n} \left(\ln|f'_a(x_0)| + \ln|f'_a(x_1)| + \cdots + \ln|f'_a(x_n)|\right),$$

quando este limite existe (mais geralmente, considera-se como limite superior e claro que se supõe que f é derivável em toda a órbita de x_0 e tem derivada não nula). Não entrando em detalhes sobre o que representam, podemos afirmar que de certa maneira, estes expoentes, no caso de serem positivos, dão uma indicação de existência de complexidade no Sistema Dinâmico. No caso da aplicação tenda, o expoente de Lyapunov está definido para todo o ponto que não é pré-imagem de $\frac{1}{2}$ (não existe derivada de f em $\frac{1}{2}$).

Na tabela 3.10 são exibidos alguns exemplos de resultados que podem ser obtidos com o procedimento **Lyapunov**.

a	x_0	Expoente Lyapunov	$\log(a)$
0.3	0.01	-1.204213761	-1.203972804
0.3	0.8	-1.204213761	-1.203972804
0.9	0.1	-.1053815758	-.1053605157
1.2	0.3	.1823580599	.1823215568
1.2	0.6	.1823580599	.1823215568
1.5	0.2	.4055461428	.4054651081
1.8	0.3	.5879044540	.5877866649
2	0.7	.6932858100	.6931471806

Tabela 3.10: Expoentes de Lyapunov obtidos com o procedimento **Lyapunov**

No caso da aplicação tenda, ao aplicar o procedimento **Lyapunov** para diferentes valores de a e x_0, o aluno pode verificar que o valor obtido é sempre constante e igual ao valor de $\ln(a)$. O professor pode levar o aluno a concluir que tal acontece porque

$|T_a'(x)| = a$, logo pela fórmula $L = \frac{1}{n} \times n \times \ln(a) = \ln(a)$. Outras funções teste poderão ser consideradas, alterando convenientemente o procedimento **Lyapunov**.

Para que o aluno tenha uma visão global da variação dos expoentes de Lyapunov com a variação do parâmetro a, construímos o procedimento lyaplot. Como parâmetros de entrada tem g, *amin*, *amax*, *npontos*, *imax* e x_0, em que:

- g é a função em estudo;
- *amin* é o valor mínimo para o parâmetro a;
- *amax* é o valor máximo para o parâmetro a;
- *npontos* é o número de pontos em que se divide o intervalo $[amin, amax]$;
- *imax* é o número de iterações;
- x_0 é o ponto inicial.

Com estes parâmetros, o procedimento dá-nos um diagrama onde são esboçados os pontos da forma (a, L) em que a representa o parâmetro e L o respectivo expoente de Lyapunov.

```
> lyaplot:=proc(g,amin,amax,npontos,imax,x0)
  local lyap,dg,h,x,L,x1,i,t:  global a:  with(plots):
  lyap:=NULL:
  for a from amin to amax by (amax-amin)/npontos do
  dg:=unapply(diff(g(x),x),x);
  h:=x->ln(abs(dg(x))):
  x1:=x0:
  L:=0:
  for i from 0 to imax do
  L:=L+h(x1):
  x1:=g(x1):
  od:  L:=L/imax:
  lyap:=lyap,pointplot([a,L],style=point,symbol=POINT,symbolsize=20,
  color=red):  end do:
  display(lyap);
  end proc:
```

As primeiras instruções do procedimento servem para calcular,para cada valor de a entre *amin* e *amax*, o expoente de Lyapunov.

```
> for a from amin to amax by(amax-amin)/npontos do
  dg:=unapply(diff(g(x),x),x);
  h:=x->ln(abs(dg(x))):
  x1:=x0:
  L:=0:
  for i from 0 to imax do
  L:=L+h(x1):
  x1:=g(x1):
  od:  L:=L/imax:
```

As instruções que se seguem são para esboçar os pontos da forma (a, L).

Para visualizarmos o diagrama dos expoentes de Lyapunov para um valor inicial de 0.01 e valores do parâmetro a entre 0.1 (não podemos escolher o parâmetro $a = 0$, pois a derivada seria nula) e 2 (com 200 subintervalos) e com 50 iterações, devemos escrever:

```
> lyaplot(g,0.1,2,200,50,0.01);
```

e obtemos a figura 3.18.

Observando este diagrama, o aluno facilmente conclui que o resultado obtido é semelhante ao que se obteria se fosse esboçado o gráfico da função logarítmica entre 0.1 e 2.

3.8 Codificação de órbitas

Como já referimos na secção 2.5 muitas vezes torna-se interessante conhecer o caminho percorrido por $f^n(x_0)$ numa determinada partição do intervalo $[a, b]$. No caso da aplicação tenda vamos ver como se comportam as órbitas de um ponto inicial x_0 na partição $[0, \frac{1}{2}, 1]$. Para esse estudo iremos usar o procedimento **itinerario** descrito na secção 2.5. Este procedimento dá-nos, para $x_0 \in [0, 1]$, uma lista de símbolos $\{s_0, \ldots, s_n\}$ em que $s_i = 0$ ou $s_i = 1$. No caso de $s_i = 0$ significa que $T_a^i(x_0) \in [0, \frac{1}{2}]$ e no caso de $s_i = 1$ significa que $T_a^i(x_0) \in [\frac{1}{2}, 1]$.

Nos exemplos que se seguem demos como parâmetros de entrada a aplicação tenda e a partição $L = [0, \frac{1}{2}, 1]$.

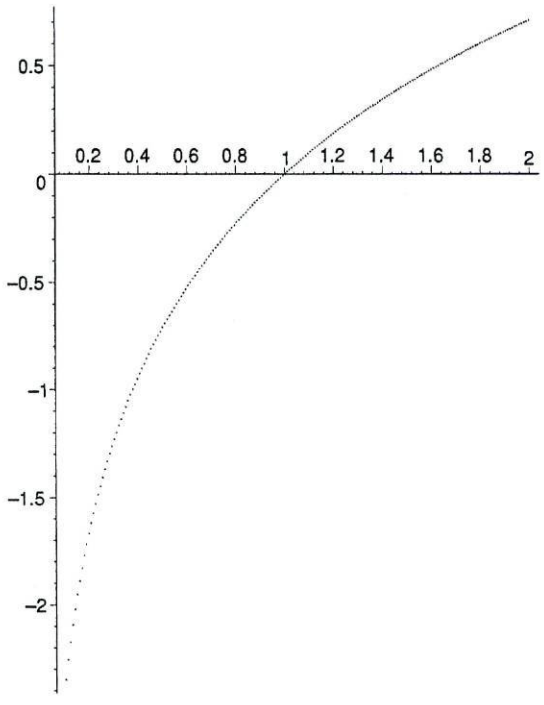


Figura 3.18: Diagrama dos expoentes de Lyapunov de T_a com $x_0 = 0.01$

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> g:=x->T(a,x):
> L:=[0,1/2,1]:
```

Foram consideradas cem iterações para diferentes valores de x_0 e diferentes elementos da família tenda.

Vamos começar por observar diferentes itinerários para valores de a menores do que um.

- Para experimentarmos, por exemplo, $a = 0.2$ e $x_0 = \frac{1}{3}$, devemos escrever

```
> a:=0.2:
> itinerario(f,L,evalf(1/3),100);
```

o resultado obtido é:

[0, 0, 0,

- Para vermos o resultado obtido com $a = 0.2$ e $x_0 = \frac{2}{3}$, escrevemos

```
> a:=0.2:

> itinerario(f,L,evalf(2/3),100);
```

e obtemos

[1, 0, 0, 0

- Para $a = 1.5$ e para $x_0 = \frac{9}{13}$ escrevemos

```
> a:=1.5:
> itinerario(f,L,evalf(9/13),100);
```

[1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1]

- Para $a = 1.5$ e para $x_0 = \frac{6}{13}$ escrevemos

```
> a:=1.5:
> itinerario(f,L,evalf(6/13),100);
```

[0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1]

Na secção 3.4.1, vimos que os pontos $\frac{6}{13}$ e $\frac{9}{13}$ formam uma órbita periódica de período dois de $T_{1.5}$, pelo que é natural que os seus itinerários sejam sequências alternadas de símbolos 0 e 1.

- Para $a = 2$ e para $x_0 = \frac{1}{3}$ escrevemos

```
> a:=2:
> itinerario(f,L,evalf(1/3),100);
```

e obtemos

[0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1]

Com $a > 1$ temos diferentes tipos de itinerários. Se construirmos vários exemplos para diferentes valores de a e x_0, podemos obter as mais variadas sequências de 0's e 1's.

Neste ponto pode ser útil, recorrendo ao método gráfico e fixando uma sequência de 0's e 1's, pedir ao aluno que determine um ponto inicial x_0 que deu origem àquele itinerário. Isso servirá para inferir se o aluno domina ou não a análise das órbitas pelo método gráfico.

Para analisarmos estatisticamente os casos em que $a > 1$ podemos usar o procedimento **diagramafreq**. Relembramos que para usar este procedimento temos de inserir a função

f, a partição L, o ponto inicial x_0 e o número de iterações pretendido n. Como resposta obtemos o histograma com a frequência com que os intervalos $[0;\frac{1}{2}]$ e $[\frac{1}{2};1]$ são visitados pelos iterados $f_a^i(x_0)$ com $i \in \{1,\ldots,n\}$. Mais precisamente, o procedimento associa ao intervalo $[0;\frac{1}{2}]$ o símbolo 0 e ao intervalo $[\frac{1}{2};1]$ o símbolo 1, contando o número de iterados que estão associados ao símbolo 0 e o número de iterados que estão associados ao símbolo 1.

Para usarmos o procedimento temos de definir a função $g(x) = T(a,x)$, a partição L e o número de iterações n.

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> g:=x->T(a,x):
> L:=[0,1/2,1]:
```

Depois definimos o parâmetro a e escrevemos uma expressão do tipo

```
> a:=2#por exemplo
```

Por fim, usamos o comando

```
> diagramafreq(g,L,evalf(1/3),100);
```

para obter o histograma para $x_0 = \frac{1}{3}$.

Na tabela 3.11 exibimos os histogramas obtidos para $a = 1.2$, $a = 1.5$ e $a = 2$.

É de fazer notar ao aluno que pelos gráficos obtidos parece que a frequência do símbolo 1 é maior do que a frequência quando $a > 1$. Contudo, como o número de iterações é muito pequeno não se pode concluir nada e o melhor será exibir outros histogramas com um número elevado de iterações. Na tabela 3.12 exibimos alguns histogramas que podem ser usados para verificar se estas primeiras observações podem ou não ser conclusivas. Todos eles foram obtidos com o procedimento **diagramafreq** e testaram-se 50000 iterações.

Analisando estes novos histogramas, o aluno pode concluir que para $1 < a < 2$ a frequência relativa do símbolo 0 é menor do que a frequência relativa do 1, ou seja, os elementos da órbita visitam mais o intervalo $[\frac{1}{2},1]$ do que o intervalo $[0,\frac{1}{2}]$. Para $a = 2$ tal já não se verifica: a frequência do símbolo 0 e do símbolo 1 são quase iguais o que quer dizer que os elementos da órbita se distribuem igualmente pelos intervalos $[0,\frac{1}{2}]$ e $[\frac{1}{2},1]$.

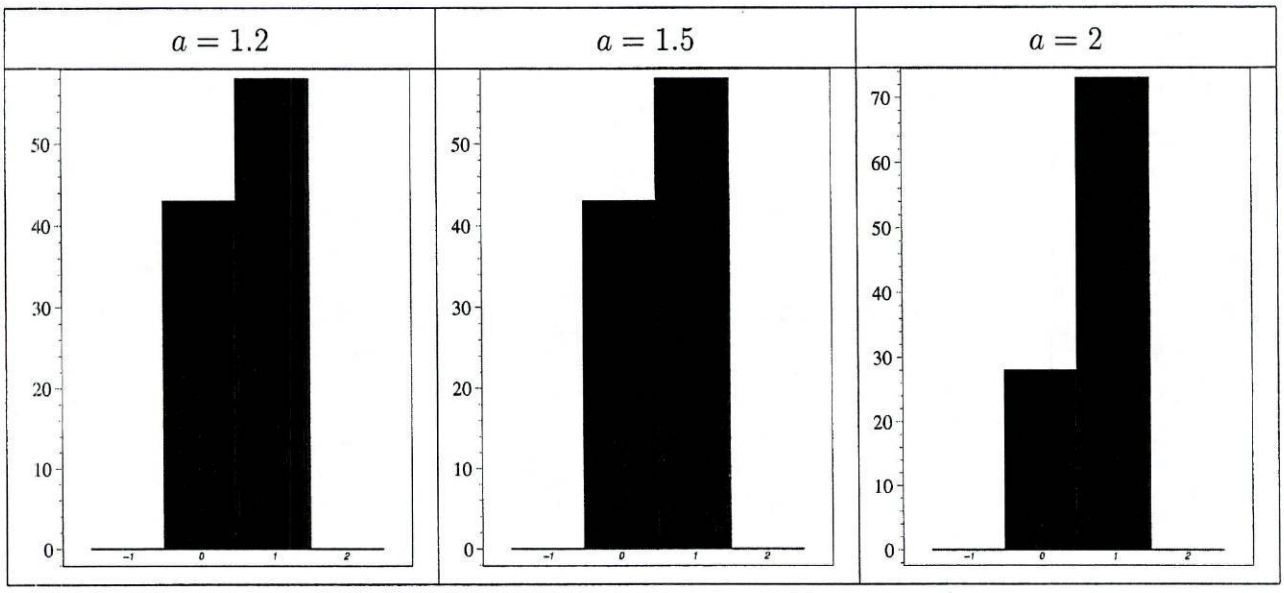


Tabela 3.11: Frequência relativa dos símbolos para diferentes valores de a e para 100 iterações

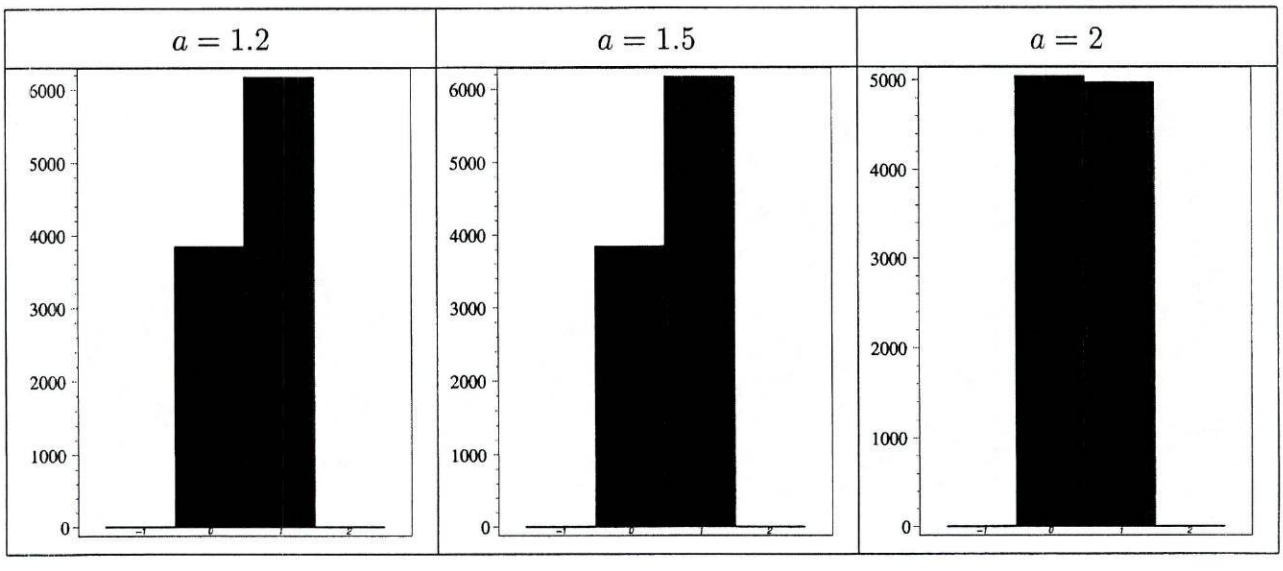


Tabela 3.12: Frequência relativa dos símbolos para diferentes valores de a e com 50000 iterações

Capítulo 4

A aplicação logística

Neste capítulo vamos estudar a família de funções $f_a(x) = ax(1-x)$ com $a > 0$, conhecida como a família das aplicações logísticas. Esta é uma aplicação definida por uma expressão muito simples, mas tal como a aplicação tenda à medida que a varia, exibe um comportamento bastante variado.

O aluno pode ser estimulado para esta família de aplicações recorrendo à modelação sobre o crescimento de uma população de bactérias. De facto, suponhamos que queremos criar um modelo matemático que descreva o tamanho de uma população de bactérias dentro de um tanque. Suponhamos também que sabíamos que em cada hora o tamanho da população das bactérias duplica. Assumamos que no início temos x_0 bactérias e, denotemos por x_n o tamanho da população na hora n. Perante estes dados, o aluno deverá proceder à modelação do problema através de um esquema equivalente a:

$$\begin{aligned} x_1 &= x_0 + x_0 = 2x_0 \\ x_2 &= x_1 + x_1 = 4x_0 \\ &\vdots \\ x_{i+1} &= x_i + x_i = 2^n x_0 \end{aligned}$$

De seguida, podemos referir que de acordo com este modelo a população das bactérias cresce (exponencialmente) e sem qualquer limite, o que contraria o facto de as fontes de sobrevivência serem limitadas, pois a partir de um certo valor o crescimento da população

estaria comprometido. Assim, o aluno conclui que têm de existir valores a partir dos quais, este modelo deixa de poder ser aplicado, pelo que surge a necessidade de criar um modelo mais sofisticado, que tem em conta os limites de crescimento da população. Os alunos são informados que um modelo possível é o *modelo logístico de crescimento* e que se apoia na função $h(x) = ax(1-x)$ (em que x tem como unidade 10^{12} bactérias).

Admitindo este modelo existem algumas perguntas que podem ser feitas: Será que a população pode manter-se constante de uma hora para a seguinte, ou seja, será que $x_{n+1} = x_n$?; Será que o tamanho da população nas unidades indicadas tem limite?; Esse limite depende do valor inicial x_0 considerado?

4.1 Esboço gráfico

A família $(f_a)_{a\in[0,4]}$, chamada de logística é definida por aplicações $f_a : [0,1] \to [0,1]$ em que

$$f_a(x) = ax(1-x), \quad 0 \leq x \leq 1, \quad 0 \leq a \leq 4.$$

Para que o Maple a reconheça, escrevemos:

```
> f:=(a,x)->a*x*(1-x):
```

Para se esboçar o gráfico da aplicação logística para um valor particular do parâmetro a, por exemplo, $a = 3.85$ escrevemos:

```
> f:=(a,x)->a*x*(1-x):
> with(plots):
> a:=3.85:
> plot(f(a,x),x=0..1,color=blue,scaling=constrained);
```

Como resposta obtemos o gráfico da figura 4.1.

O procedimento animafunção pode ser usado com a aplicação logística, permitindo visualizar a variação do gráfico de f com o parâmetro a, para o aluno perceber que $f_a([0,1]) \subseteq [0,1]$ com $a \in [0,4]$.

Na tabela 4.1 exibimos alguns dos gráficos intermédios obtidos com esse procedimento.

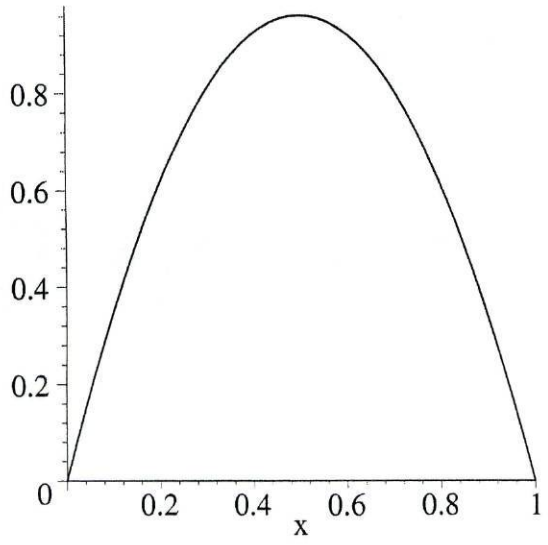


Figura 4.1: Gráfico da aplicação logística para $a = 3.85$

Observando os gráficos da tabela 4.1, o aluno pode concluir que o gráfico da aplicação logística é sempre uma parábola com a concavidade voltada para baixo, que intersecta o eixo dos xx em $x = 0$ e em $x = 1$. Também podemos usar os gráficos desta tabela para justificarmos a escolha de $a \in [0, 4]$. Analisando os gráficos da tabela, o aluno pode observar que o valor máximo da função é dado pela ordenada do vértice da parábola. Para determinar analiticamente o valor desse máximo, o aluno poderá aplicar os seus conhecimentos de cálculo diferencial. Assim, recorrendo à derivada da função f_a infere que o ponto máximo é atingido quando a derivada de f_a se anula. Embora possa chegar ao resultado sem o recurso ao Maple, escrevendo:

```
> f:=(a,x)->a*x*(1-x):

> g:=(a,x)->diff(f(a,x),x):# derivada de f

> max:=solve(g(a,x)=0,x);#zeros da derivada
```

Obtém os zeros da derivada de f_a:

$$\frac{1}{2}$$

De seguida, determina esse máximo que corresponde ao valor de $f_a(max)$.

```
> f(a,max);
```

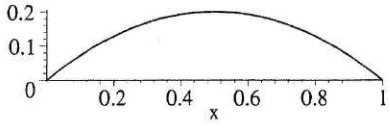
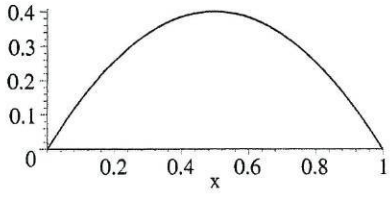
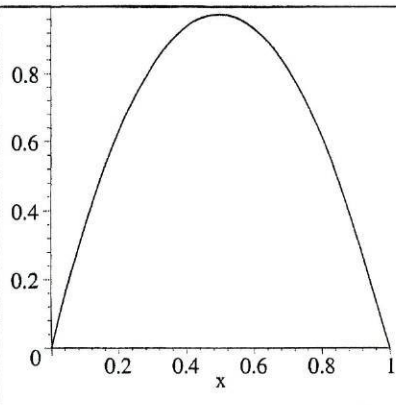
$a = 0.3$	$a = 1.2$	$a = 2$
		

Tabela 4.1: Exemplos de gráficos da aplicação logística

E obtemos:

$$\frac{a}{4}$$

Com estes cálculos o aluno conclui que, de facto o vértice da parábola tem coordenadas $(\frac{1}{2}, \frac{a}{4})$ e que o valor do parâmetro a tem de variar entre 0 e 4. Para que o aluno fique convencido que de facto se $a > 4$, $f_a([0,1]) \not\subseteq [0,1]$, podemos apresentar-lhe um exemplo como o da figura 4.2 ou outro semelhante.

No gráfico da figura 4.2, escolhemos $a = 4.8$ e para o obtermos escrevemos os seguintes comandos:

```
> a:=4.8:
```

```
> plot(f(a,x),x=0..1,color=blue,scaling=constrained);
```

Ao analisar o gráfico o aluno pode facilmente observar que o vértice da parábola está fora de $[0,1]$, concluindo que se $a > 4$, $[0,1]$ não é totalmente enviado em $[0,1]$ por f_a.

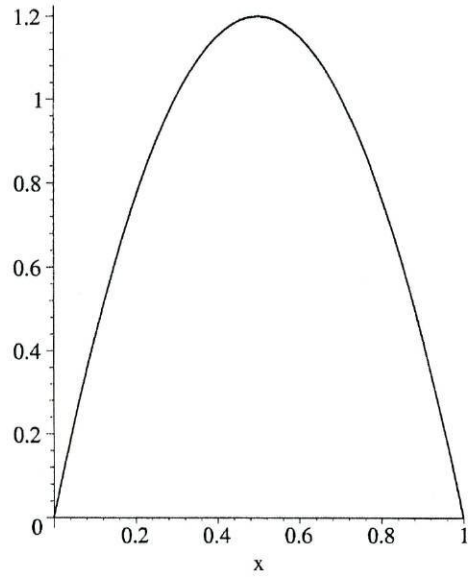


Figura 4.2: Gráfico da aplicação logística para $a = 4.8$

4.2 Análise de órbitas

Nesta secção vamos usar os procedimentos apresentados na secção 2.1 para estudar alguns pedaços de órbitas da aplicação logística para diferentes valores do parâmetro a e do ponto inicial x_0.

4.2.1 Procedimento listaorbita

Vamos começar por usar o procedimento **listaorbita** considerando os dez primeiros iterados.

Para usar este procedimento é necessário definir previamente a função f_a.

```
> f:=(a,x)->a*x*(1-x):
```

- Vamos começar por observar um exemplo em que $0 < a < 1$. Para isso escolhemos, por exemplo, $a = 0.5$ e $x_0 = 0.4$.

  ```
  > a:=0.5:
  ```

  ```
  > listaorbita(f,0.4,0,10);
  ```

[0.4, 0.120, 0.0528000, 0.02500608000, 0.01219038798, 0.006020891210, 0.002992320040, 0.001491683030, 0.0007447289559, 0.0003720871674, 0.0001859743593]

O que os dez primeiros iterados parecem indicar é que x_n converge para 0.

- Vamos agora ver o caso em que $a > 1$, por exemplo $a = 2.5$. Se escolhermos $x_0 = 0.2$, por exemplo, escrevemos:

```
> a:=2.5:
> listaorbita(f,0.2,0,10);
```

E obtemos a seguinte lista:

[0.2, 0.400, 0.6000000, 0.6000000000, 0.6000000000, 0.6000000000, 0.6000000000, 0.6000000000, 0.6000000000, 0.6000000000, 0.6000000000]

Tudo indica que o terceiro iterado (0.6) é um ponto fixo.

- Vamos ver um exemplo em que $a = 3.1$ e $x_0 = 0.2$. Ao escrevermos

```
> a:=3.1:
> listaorbita(f,0.2,0,10);
```

obtemos

[0.2, 0.496, 0.7749504, 0.5406470604, 0.7698782310, 0.5492137953, 0.7674918071, 0.5531892127, 0.7662298136, 0.5552772276, 0.7655277273]

Os primeiros dez iterados parecem indicar que a órbita está a convergir para uma órbita periódica de período dois. O aluno pode confirmar esta conjectura experimentando com valores de n mais elevados.

- Vejamos agora o exemplo em que $a = 3.5$ e $x_0 = 0.2$. Se escrevermos

```
> a:=3.5:
> listaorbita(f,0.2,0,10);
```

obtemos

[0.2, 0.560, 0.8624000, 0.4153318400, 0.8499095594, 0.4464715509, 0.8649714679, 0.4087853967, 0.8458796362, 0.4562854705, 0.8683116398]

Aqui o aluno pode conjecturar que x_0 é enviado por f_a numa órbita periódica de período quatro. Procedendo à experimentação com valores de n mais elevados esta conjectura pode ser confirmada.

4.2.2 Recurso às tabelas

Para estudarmos a órbita, ou melhor um pedaço de órbita, de um ponto inicial x_0 por uma função f podemos também recorrer às potencialidades do Maple na criação de tabelas, conforme referido na secção 1.10.

Maple 8 - [orbitatabelalogistica.mws - [Server 1]]

File Edit View Insert Format Spreadsheet Window Help

```
> restart:
> f:=x->0.8*x*(1-x):# parâmetro a=0.8
> g:=x->3.1*x*(1-x):# parâmetro a=3.1
> h:=x->3.5*x*(1-x):# parâmetro a=3.5
> x0:=0.2:m:=0:n:=15:
```

	A	B	C	D
1	n	x_n	x_n	x_n
2	0	0.2000	0.2000	0.2000
3	1	0.1280	0.4960	0.5600
4	2	0.0893	0.7750	0.8624
5	3	0.0651	0.5406	0.4153
6	4	0.0487	0.7699	0.8499
7	5	0.0370	0.5492	0.4465
8	6	0.0285	0.7675	0.8650
9	7	0.0222	0.5532	0.4088
10	8	0.0173	0.7662	0.8459
11	9	0.0136	0.5553	0.4563
12	10	0.0108	0.7655	0.8683
13	11	0.0085	0.5564	0.4002
14	12	0.0068	0.7651	0.8401
15	13	0.0054	0.5571	0.4700
16	14	0.0043	0.7649	0.8719
17	15	0.0034	0.5575	0.3910

Time: 40.6s | Bytes: 5.94M | Available: 385M

Figura 4.3: Primeiros quinze elementos da órbita de $f_{0.8}(0.2)$, $f_{3.1}(0.2)$ e de $f_{3.5}(0.2)$

A tabela que exibimos na figura 4.3 mostra os primeiros quinze iterados pela aplicação logística para diferentes valores do parâmetro a e para o mesmo ponto inicial $x_0 = 0.2$. Nas colunas B, C e D aparecem os iterados x_n com $x_n = f_a(x_0)$ tendo sido considerado $a = 0.8$, $a = 3.1$ e $a = 3.8$, respectivamente. Comparando os valores destas três colunas, o aluno poderá facilmente concluir que para $a = 0.8$ a órbita está a dirigir-se para 0; para $a = 3.1$ x_0 é enviado por f_a numa órbita periódica de período dois e para $a = 3.5$ x_0 é enviado por f_a numa órbita periódica de período quatro.

Com a análise desta tabela, o aluno pode concluir que a aplicação logística apresenta comportamentos diferentes de acordo com o valor considerado para o parâmetro a. Claro está que considerámos um número muito pequeno de iterados e que os pedaços de órbita considerados podem não demonstrar o que realmente acontece. Mais uma vez deve-se apelar à experimentação por parte do aluno.

4.2.3 Análise gráfica

Para usarmos o procedimento **plotorbita** com a aplicação logística temos de fazer a mesma alteração que fizemos para a aplicação tenda, ou seja, temos de acrescentar ao procedimento mais um parâmetro de entrada que representa o valor do parâmetro a a considerar. Assim, o procedimento **plotorbitamod** tem como parâmetros de entrada: f que é a função em estudo, a que é o parâmetro, n que é o número de iterações e x_0 que é o ponto inicial.

Para esboçar o gráfico da órbita de f_a para $a = 1.5$, com 100 iterações e para $x_0 = 0.1$ devemos escrever:

```
> f:=(a,x)->a*x*(1-x):

> plotorbitamod(f,1.5,100,0.1);
```

Com estas instruções obtemos o gráfico da figura 4.4.

Observando o gráfico o aluno pode inferir que a órbita de x_0 converge para um único ponto. Poderá ser colocada a questão: "Qual a órbita do ponto para o qual converge a função?".

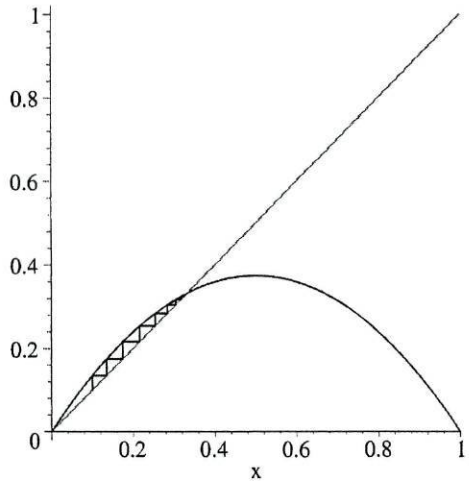


Figura 4.4: Representação gráfica da órbita da aplicação logística para $a = 1.5$ e $x_0 = 0.1$

4.2.4 Variação com o parâmetro a

Através de gráficos obtidos com o procedimento **plotorbitamod**, podemos analisar a órbita de um ponto para diferentes valores do parâmetro a.

Estudo do caso em que $a < 1$

Vamos começar por considerar valores do parâmetro a menores do que um.

Depois de definida a função f_a, se executarmos os comandos

```
> plotorbitamod(f,0.2,100,0.6):
> plotorbitamod(f,0.4,100,0.5):
> plotorbitamod(f,0.7,100,0.4):
> plotorbitamod(f,0.9,100,0.4):
```

obtemos os gráficos representados na tabela 4.2.

Nessa tabela apresentamos quatro exemplos de representações gráficas de órbitas para diferentes valores de a e x_0.

Com base em tabelas deste tipo, o aluno pode concluir que para valores de $a < 1$ o ω-limite da órbita é 0.

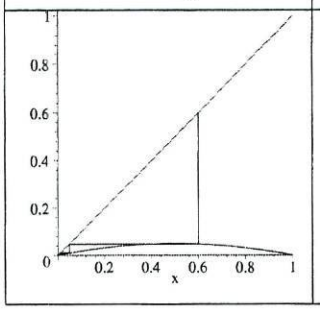
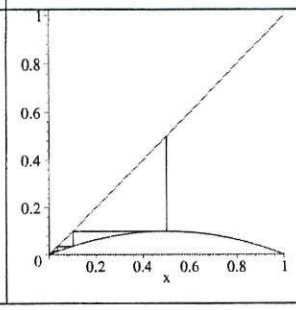
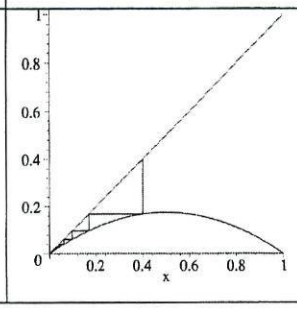
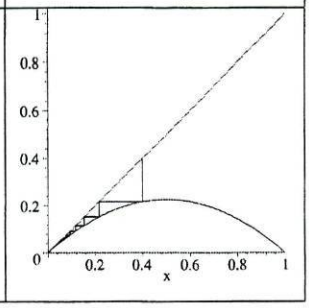
$a = 0.2$ e $x_0 = 0.6$	$a = 0.4$ e $x_0 = 0.5$	$a = 0.7$ e $x_0 = 0.4$	$a = 0.9$ e $x_0 = 0.4$
			

Tabela 4.2: Representação gráfica de órbitas

Para se exibir graficamente a aproximação do ω-limite de determinado ponto criamos o procedimento omegalimite, que tem como parâmetros de entrada f, a, x_0, m e n em que:

- f é a função a estudar;
- a é o parâmetro fixado para a função;
- x_0 é o ponto inicial;
- m é o tempo a partir do qual queremos visualizar os iterados;
- n é o instante final que se pretende visualizar.

O procedimento **omegalimite** é constituído pelos seguintes comandos:

```
> omegalimite:=proc(f,a,x0,m,n)
  local i,x:
  with(plots):
  x[0]:=x0:
  for i from 1 to n do
  x[i]:=f(a,x[i-1]):
  od:
  pointplot([seq([i,x[i]],i=m+1..n)],view=[m..n,0..1]);
  end proc:
```

A ideia de funcionamento deste procedimento é simples. A linha inicial que o define

```
> omegalimite:=proc(f,a,x0,m,n)
```

é seguida pelo grupo de instruções:

```
> x[0]:=x0:
  for i from 1 to n do
  x[i]:=f(a,x[i-1]):
  od:
```

e servem para se determinar os primeiros n elementos da órbita do ponto inicial x_0.

Depois de termos os n elementos da órbita, mandamos desenhar os pontos de coordenadas (i, x_i) para valores de i entre m e n.

```
> pointplot([seq([i,x[i]],i=m+1..n)],view=[m..n,0..1],color=red);
```

Para que sejam facilmente visíveis esses pontos são esboçados de cor vermelha e o eixo dos xx é visualizado entre m e n.

Este procedimento esboça um gráfico com os pontos de coordenadas (i, x_i) sendo que a sua projecção no eixo das ordenadas dá-nos o valor do ω-limite. Depois de testarmos o procedimento e se identificarmos uma linha horizontal, então o ω-limite é um ponto fixo; se obtivermos duas linhas distintas, então o ω-limite converge para uma órbita periódica de período dois, etc. O aluno deverá concluir que o ω-limite se obtém sempre através da projecção no eixo das ordenadas dos gráficos obtidos com este procedimento.

Para testarmos, o procedimento com, por exemplo, $a = 0.5$ e $x_0 = 0.2$ para $m = 100$ e $n = 200$, devemos escrever os seguintes comandos:

```
> omegalimite(f,0.5,0.2,100,200);
```

Na figura 4.5 exibimos o gráfico obtido com a instrução acima indicada.

Observando o gráfico da figura 4.5, nota-se a presença de uma linha horizontal, ou seja, o ω-limite é um ponto fixo.

Através do gráfico da figura 4.5 ou de outros análogos, o aluno sem dificuldade pode concluir que para valores de $a < 1$ o ω-limite das órbitas de um ponto inicial x_0 é 0.

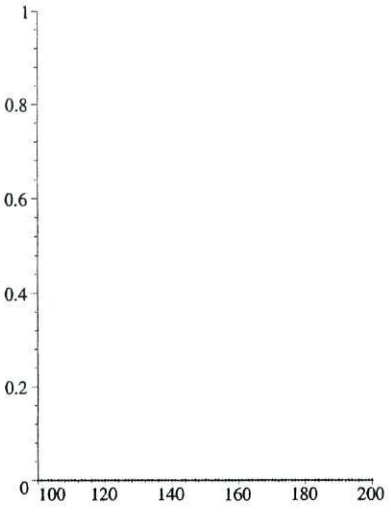


Figura 4.5: ω-limite da órbita de $f_{0.5}(0.2)$

Estudo do caso em que $1 < a < 3$

Recorrendo ao procedimento **plotorbitamod** e fazendo experimentação para valores de a entre 1 e 3, o aluno pode verificar que todo o ponto $x_0 \in [0,1]$ é enviado num ponto fixo.

Nos exemplos a seguir exibidos, escolhemos $a = 1.2$ e $x_0 = 0.4$, $a = 1.8$ e $x_0 = 0.1$, $a = 2.4$ e $x_0 = 0.1$, $a = 2.9$ e $x_0 = 0.1$. Depois de definida a função, para obtermos as respectivas representações gráficas das órbitas com o procedimento **plotorbitamod** escrevemos as instruções:

```
> plotorbitamod(f,1.2,100,0.4):
> plotorbitamod(f,1.8,100,0.1):
> plotorbitamod(f,2.4,100,0.1):
> plotorbitamod(f,2.9,100,0.1):
```

Os resultados obtidos estão expressos nos gráficos da tabela 4.3.

Recorrendo a estes gráficos ou a outros idênticos, podemos levar o aluno a concluir que para valores de a maiores do que um e menores do que três, x_0 é enviado por f_a numa órbita periódica de período um. Para reforçar esta ideia, podemos recorrer ao procedimento

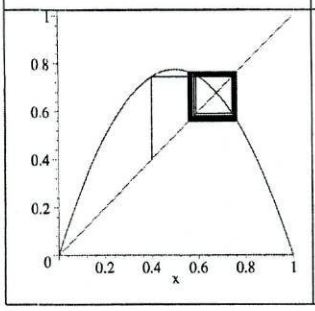
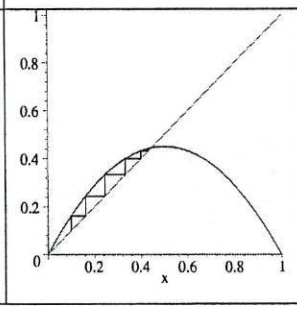
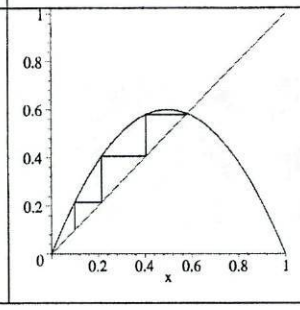
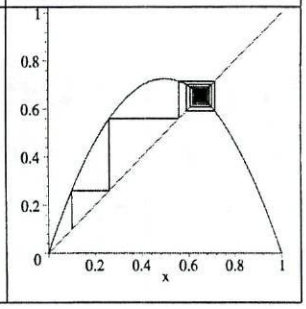
$a = 1.2$ e $x_0 = 0.4$	$a = 1.8$ e $x_0 = 0.1$	$a = 2.4$ e $x_0 = 0.1$	$a = 2.9$ e $x_0 = 0.1$
			

Tabela 4.3: Órbitas para $1 < a < 3$

omegalimite com diferentes parâmetros. Por exemplo, os gráficos da tabela 4.4 que foram obtidos com as instruções:

```
> omegalimite(f,1.2,0.4,100,200);
> omegalimite(f,1.8,0.1,100,200);
> omegalimite(f,2.4,0.1,100,200);
> omegalimite(f,2.9,0.1,100,200);
```

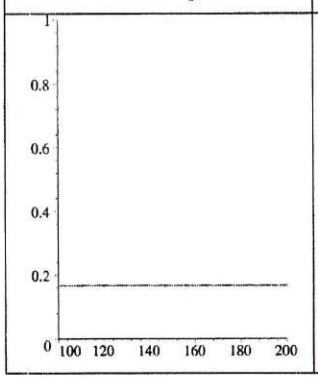
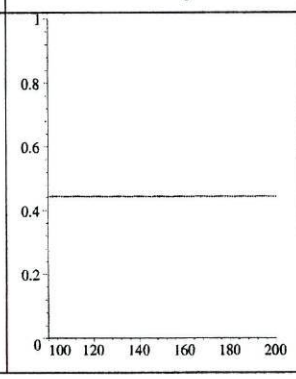
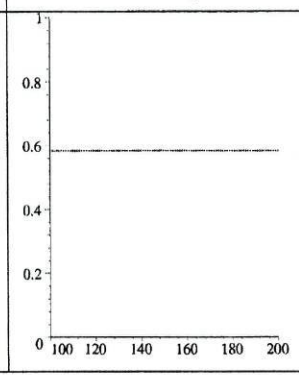
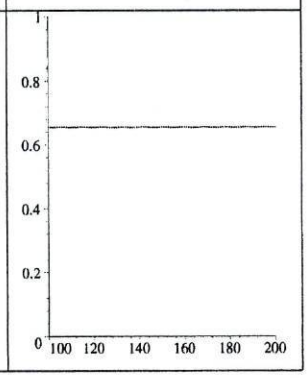
$a = 1.2$ e $x_0 = 0.4$	$a = 1.8$ e $x_0 = 0.1$	$a = 2.4$ e $x_0 = 0.1$	$a = 2.9$ e $x_0 = 0.1$
			

Tabela 4.4: ω-limite para valores de $1 < a < 3$

levam mais uma vez a concluir que para valores de $1 < a < 3$ o ponto inicial x_0 é enviado por f_a num ponto fixo.

Nesta fase do estudo podemos reforçar a ideia que o ω-limite não depende do valor inicial x_0 considerado, ou seja, para um parâmetro previamente arbitrado, todas as órbitas

convergem para um ponto fixo de f_a. Para isso, podemos exibir algumas representações gráficas do ω-limite de várias órbitas para o mesmo valor do parâmetro a e para diferentes valores das condições iniciais. Valendo-nos do procedimento **omegalimite**, se escolhermos, por exemplo, $a = 2.3$ e $x_0 = 0.2$, $x_0 = 0.4$, $x_0 = 0.6$ e $x_0 = 0.8$ e escrevendo os seguintes comandos

```
> omegalimite(f,2.3,0.2,100,200);
> omegalimite(f,2.3,0.4,100,200);
> omegalimite(f,2.3,0.6,100,200);
> omegalimite(f,2.3,0.8,100,200);
```

obtemos os gráficos da tabela 4.5.

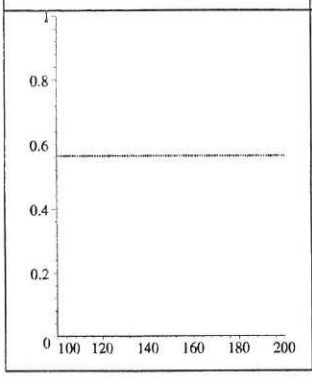
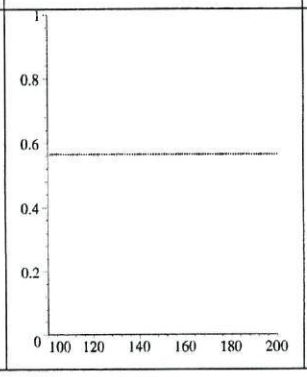
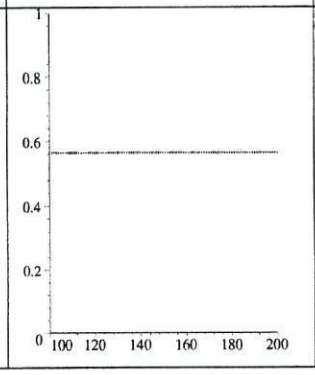
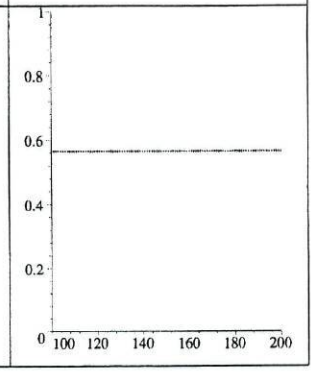
$a = 2.3$ e $x_0 = 0.2$	$a = 2.3$ e $x_0 = 0.4$	$a = 2.3$ e $x_0 = 0.6$	$a = 2.3$ e $x_0 = 0.8$
			

Tabela 4.5: O ω-limite não depende do valor de x_0 considerado

Com base neste tipo de resultados, o aluno sem grande dificuldade percebe que $f_a^n(x_0)$ converge para o mesmo ponto fixo para qualquer que seja o ponto inicial $x_0 \in [0, 1]$.

Estudo do caso em que $a > 3$

Depois de estudarmos os casos em que $a < 3$ podemos iniciar o aluno nos novos fenómenos que aparecem com $a > 3$. Ao estudar este caso, o aluno será confrontado com pontos periódicos de período dois, pontos periódicos de período quatro e também com comportamento não periódico.

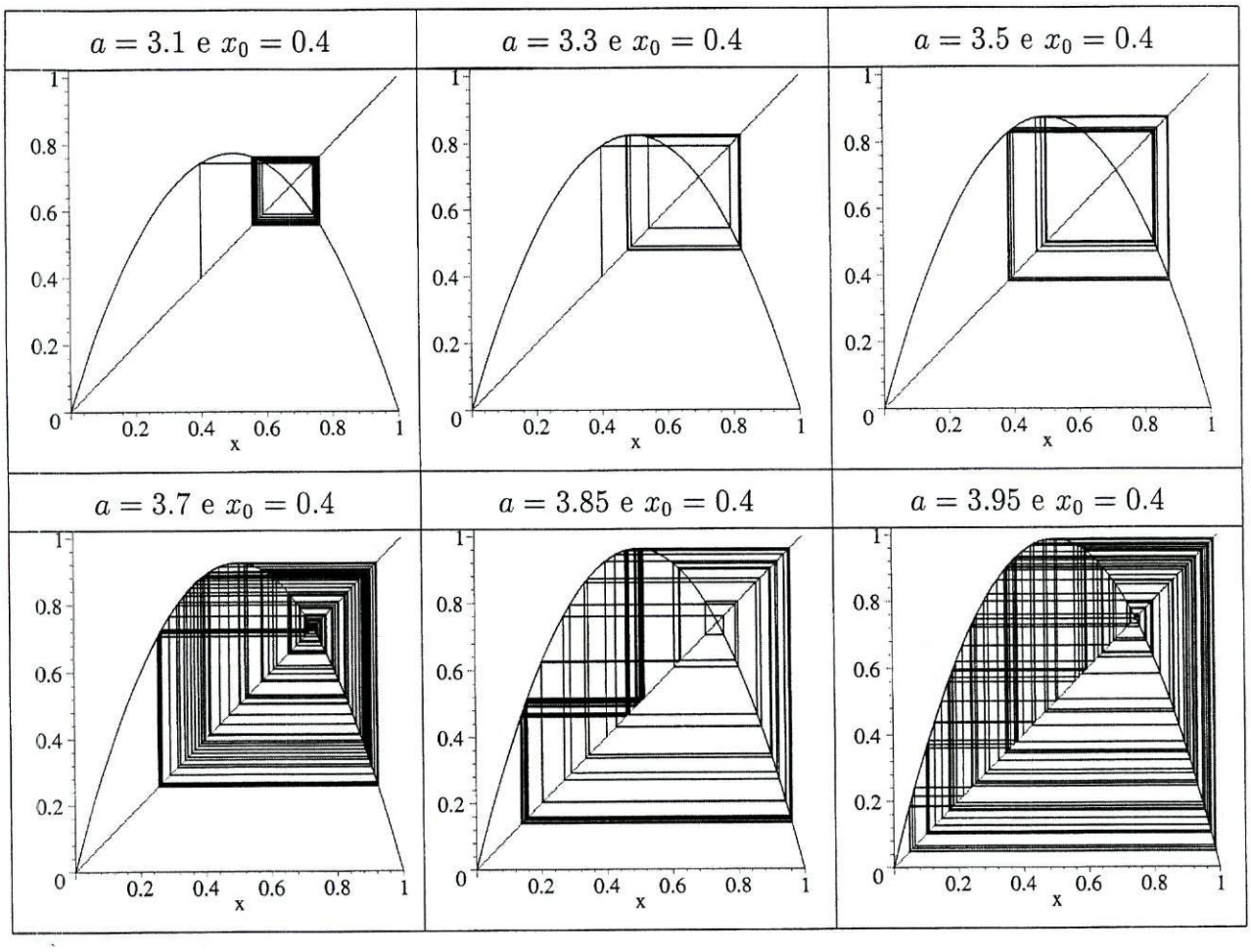


Tabela 4.6: Órbitas de f_a para diferentes valores de a.

Na tabela 4.6 exibimos algumas representações gráficas de órbitas para valores de a entre três e quatro. Após termos definido a função f_a, os gráficos foram obtidos com as instruções:

```
> plotorbitamod(f,3.1,100,0.4):
> plotorbitamod(f,3.3,100,0.4):
> plotorbitamod(f,3.5,100,0.4):
> plotorbitamod(f,3.7,100,0.4):
> plotorbitamod(f,3.85,100,0.4):
> plotorbitamod(f,3.95,100,0.4):
```

Com o recurso às representações gráficas obtidas e a outras que o aluno pode e deve experimentar, este deverá concluir que experimentando valores de a entre 3 e 3.7 e $x_0 \in [0,1]$, os resultados obtidos são semelhantes aos obtidos para $a = 3.1$ e que todo o ponto inicial $x_0 \in [0,1]$ é enviado por f_a numa órbita periódica de período dois.

Experimentando valores de $a \geq 3.7$ o aluno deverá então reproduzir figuras similares às obtidas na tabela 4.6 e assim concluir que para estes valores de a não existe uma órbita com as características das órbitas obtidas para $a < 3.6$, ou seja, as órbitas não parecem convergir para uma órbita periódica.

Para valores de a superiores a 3.5 não se consegue perceber se as órbitas são periódicas ou se têm período elevado. Usando o procedimento **omegalimite**, o aluno pode tirar conclusões de carácter mais assertivo. Uma vez que não existe dúvida que para $a \leq 3.6$ as condições iniciais são enviadas por f_a em órbitas periódicas de período dois, só exibiremos na tabela 4.7 um exemplo com $a \leq 3.6$. Nos restantes exemplos consideramos valores de $a > 3.6$.

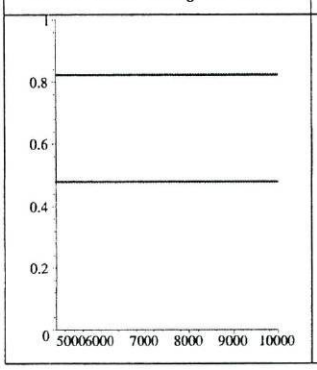
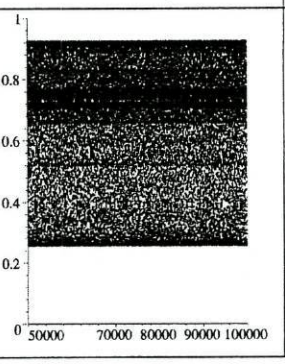
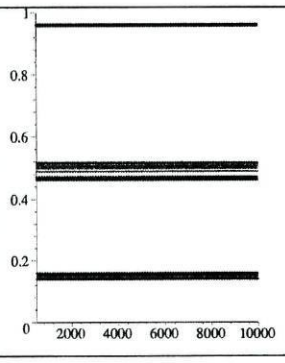
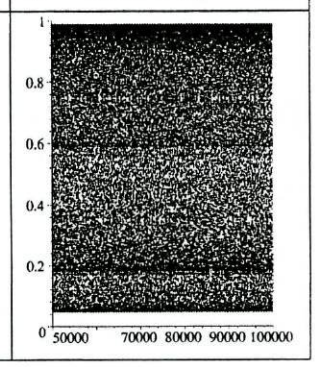
$a = 3.3$ e $x_0 = 0.2$	$a = 3.7$ e $x_0 = 0.2$	$a = 3.85$ e $x_0 = 0.2$	$a = 3.95$ e $x_0 = 0.2$
			

Tabela 4.7: ω-limite para valores de $a > 3$

Analisando os gráficos obtidos na tabela 4.7 o aluno pode concluir que:

- Para $a = 3.3$, de facto, x_0 é enviado por f_a numa órbita periódica de período dois;
- Para $a = 3.7$ e para $a = 3.95$ o sistema apresenta um comportamento não periódico;
- Para $a = 3.85$ o ω-limite é constituído por três intervalos distintos, ou seja, reparte-se por regiões distintas.

4.3 Pontos fixos

Nesta secção vamos rever os métodos já usados no capítulo anterior, para determinar analiticamente os pontos fixos da aplicação logística. Primeiro vamos usar os métodos analíticos e depois os métodos geométricos.

4.3.1 Métodos analíticos

Conforme escrito na secção 2.3.1 existem dois comandos no Maple que permitem determinar analiticamente os pontos fixos de uma função: são eles o comando **solve** e o comando **fsolve**.

Comando solve

Para determinar os pontos fixos de f, podemos resolver a equação $f_a(x) = x$. Uma vez que se trata de um polinómio do 2º grau, podemos, sem receio, deixar o aluno usar o comando **solve**. Escrevemos, então os comandos:

```
> f:=(a,x)->a*x*(1-x):
> solve(f(a,x)=x,x);
```

Como soluções da equação $f_a(x) = x$, obtemos:

$$0, \frac{a-1}{a}$$

Com estes comandos, o aluno tem mais um meio ao seu dispor para reforçar a ideia que a aplicação logística tem dois pontos fixos: 0 e $\frac{a-1}{a}$.

A seguir, podemos pedir ao aluno que faça uma análise mais crítica acerca dos resultados obtidos, nomeadamente que indique para que valores do parâmetro a, f_a tem o ponto fixo $\frac{a-1}{a}$ no intervalo $[0,1]$. Uma vez que, como já vimos, esta região ($[0,1]$) é invariante por f_a, podemos levá-lo a concluir que no intervalo $[0,1]$:

- Para $0 \leq a \leq 1$, f_a tem um só ponto fixo;
- Para $a > 1$, f_a tem dois pontos fixos: 0 e $\frac{a-1}{a}$.

Em princípio, o aluno sentir-se-á mais à vontade em atribuir valores ao parâmetro a e a partir deste, determinar os pontos fixos de f_a. Para determinar algebricamente os pontos fixos de f_a para um parâmetro a, por exemplo, $a = \frac{23}{10}$, escrevemos os comandos:

```
> a:=23/10:

> solve(f(a,x)=x,x);
```

E obtemos os pontos:

$$0, \frac{13}{23}$$

Comando fsolve

Ao recorrermos ao comando **fsolve** para determinar numericamente os pontos fixos da função f_a é necessário primeiro atribuir um valor ao parâmetro a. Depois de definida a função f_a e atribuindo, por exemplo o valor 3.6 ao parâmetro a, ao escrevermos:

```
> a:=3.6:

> fsolve(f(a,x)=x,x);
```

obtemos

$$0., 0.7222222222$$

Para obtermos o resultado com mais casas decimais, por exemplo cem casas, podemos usar, como já referimos na secção 1.3 a instrução:

```
> Digits:=100:
```

Com esta instrução, ao escrevermos

```
> a:=3.6:

> fsolve(f(a,x)=x,x);
```

obtemos

```
0., 0.722222222222222222222222222222222222222222222222222222222222\
22222222222222222222222222222222222222
```

4.3.2 Método gráfico

Para que o aluno tenha uma visão geométrica da varição dos pontos fixos de f_a, vamos usar o procedimento **pontosfixospmplot** descrito na secção 3.3.2, que permite obter no mesmo sistema de eixos a função f_a e a função identidade. Para visualizarmos, por exemplo os pontos fixos de $f_{2.6}$ entre 0 e 1 devemos escrever:

```
> pontosfixospm(f,2.6,0,1);
```

Obtemos o gráfico da figura 4.6

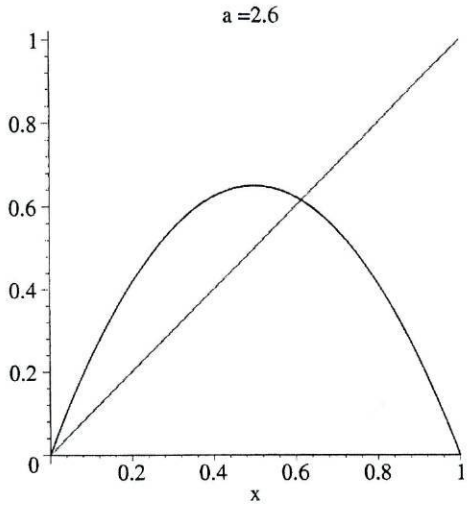


Figura 4.6: Pontos fixos de $f_{2.6}$

O aluno pode observar a figura 4.6 e concluir que existem dois pontos de intersecção do gráfico da identidade com o gráfico da função f_a, ou seja, f_a tem dois pontos fixos: o 0 e outro que corresponde a um valor entre 0.6 e 0.7. Para obter com maior precisão o ponto fixo que está em $[0.6; 0.7]$, podemos usar de novo o procedimento **pontosfixospmplot** com os seguintes parâmetros de entrada:

```
> pontosfixospmplot(f,2.6,0.6,0.7);
```

Obtendo o gráfico da figura 4.7.

Com este gráfico, o aluno obtém um melhor enquadramento para o ponto fixo que está entre 0.6 e 0.7, podendo concluir que, de facto, esse ponto fixo se situa entre 0.6 e 0.62.

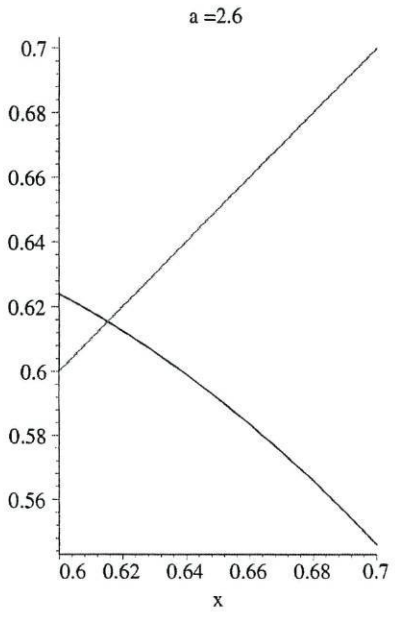


Figura 4.7: Ponto fixo de $f_{2.6}$ entre $[0.6; 0.7]$

Com este método de experimentação, podemos pedir ao aluno que indique uma aproximação para o ponto fixo usando, por exemplo três ou quatro casas decimais.

Como a aplicação logística depende de um parâmetro a, podemos usar o procedimento **animafixos** descrito na secção 3.3.2, para mostrar aos alunos como se comportam os pontos fixos da função, variando o parâmetro a. Na tabela 4.8 exibimos alguns dos gráficos intermédios que podem ser obtidos com a instrução

```
> animafixos(f,0,4,400);
```

Neste exemplo, consideramos valores de a entre 0 e 4 com 400 pontos de amostragem.

Observando os gráficos da tabela 4.8 ou com mais experimentação, o aluno pode concluir sem dificuldade que:

- Se $0 < a \leq 1$, f_a tem só um ponto fixo;

- Se $1 < a \leq 4$, f_a tem dois pontos fixos: 0 e $\frac{a-1}{a}$.

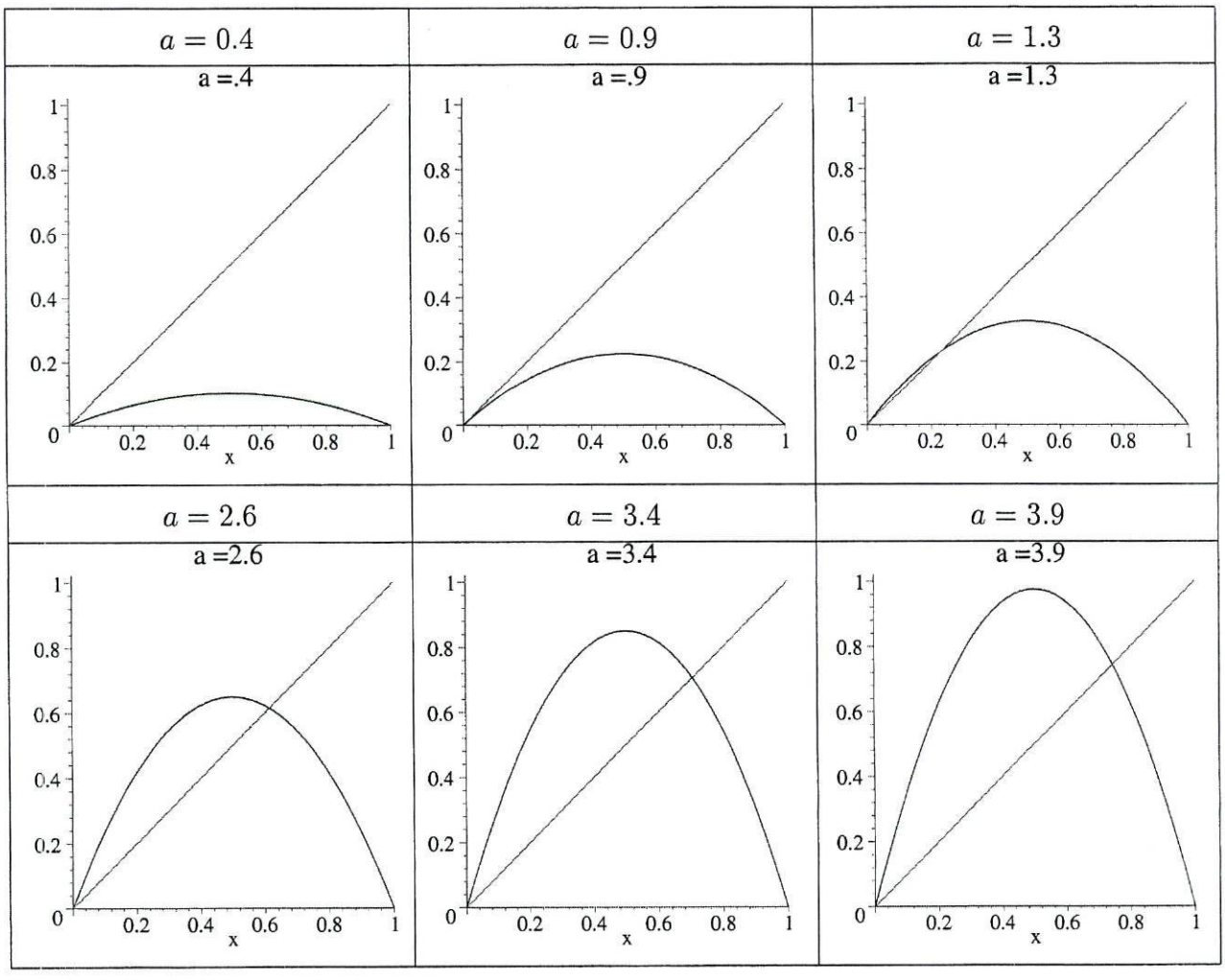


Tabela 4.8: Pontos fixos de f_a usando **animafixos**

4.4 Período dois

Para determinarmos os pontos periódicos de período dois vamos precisar de introduzir no Maple a instrução $f \circ f$. Como a função f_a tem como conjunto partida dois elementos e como conjunto de chegada um só, para conseguirmos obter a expressão de $f \circ f$ temos de definir a função g tal que fixado um parâmetro a, $g(x) = f(a, x)$(tal como foi feito no capítulo 3).

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):
```

Depois de termos definido esta função, usamos os utilitários desenvolvidos na secção 2.3.2. Primeiro vamos tentar determinar os pontos periódicos de período dois pelos métodos analíticos prosseguindo depois com o método gráfico.

4.4.1 Métodos analíticos

Como sabemos, para determinar analiticamente os pontos periódicos de período dois de uma função g temos de determinar as soluções reais da equação $g^2(x) = x$ e da equação $g(x) = x$. Os pontos periódicos de período dois são aqueles que são solução da primeira equação e não são solução da segunda. Baseados nesta ideia, vamos à semelhança do que já se fez no capítulo anterior, ver como se pode instruir o Maple no sentido de obter os pontos periódicos de período dois da família logística.

Comando solve

Na secção 2.3.2, recorrendo ao comando **solve**, criamos o procedimento perdois que nos dá a lista dos pontos periódicos de período dois. Vamos usar este procedimento para ver se conseguimos determinar a expressão geral dos pontos periódicos de período dois. Como já aconteceu na secção 3.4.1, ao escrevermos as instruções

```
> f:=(a,x)->a*x*(1-x):
> g:=x->f(a,x):
> perdois(g);
```

sem darmos alguma indicação sobre o parâmetro a, obtemos como resposta uma lista vazia.

[]

Se deixarmos o aluno usar os comandos descritos, com o resultado obtido, seria levado a concluir erradamente que não existem pontos periódicos de período dois para a aplicação logística. De facto, quando o Maple resolve algebricamente a equação $f^2(a, x) = x$ e há variação no número de soluções dependendo do parâmetro a considerado, a resposta dada não é a resposta conhecida. Para evitar estes erros, temos de fornecer ao Maple, uma

restrição para os valores que o parâmetro a pode tomar e onde seguramente há soluções para a equação $f_a^2(x) = x$. A condição que devemos fornecer é $a > 3$. Para isso usamos o comando:

```
> assume(a>3):
```

Com esta instrução, ao executar:

```
> perdois(g);
```

é apresentada a resposta:

$$\left[\frac{\frac{a\tilde{}}{2}+\frac{1}{2}+\frac{\sqrt{-3-2\,a\tilde{}+a\tilde{}^2}}{2}}{a\tilde{}},\ \frac{\frac{a\tilde{}}{2}+\frac{1}{2}-\frac{\sqrt{-3-2\,a\tilde{}+a\tilde{}^2}}{2}}{a\tilde{}}\right]$$

que, de facto, são os pontos periódicos de período dois da aplicação logística para $a > 3$.

Note-se que se indicarmos que $a \leq 3$, a resposta dada é a mesma que é dada sem usar o comando **assume**$(a > 3)$.

```
> assume(a<=3):
> perdois(g);
```

[]

Se usarmos o procedimento **perdois** para determinar os pontos periódicos de período dois, para um parâmetro pré-definido, por exemplo $a = \frac{77}{20}$, basta escrever:

```
> a:=77/20:
> perdois(g);
```

obtemos:

$$[\frac{97}{154}+\frac{\sqrt{1649}}{154},\ \frac{97}{154}-\frac{\sqrt{1649}}{154}]$$

Contrariamente ao que se obtém para um valor genérico do parâmetro a, se definirmos previamente o parâmetro a utilizar, o procedimento **perdois** dá-nos, neste caso, um resultado condizente com as expectativas. Assim, o procedimento **perdois** poderá ser usado pelo aluno, tendo sempre em atenção que este introduza aplicações da família logística com um parâmetro que já esteja definido.

Sem recorrer ao procedimento **perdois**, as instruções para determinar os pontos

periódicos de período dois são simples, mas podem levar o aluno a concluir que existem sempre dois pontos periódicos de período dois.

Para determinar os pontos fixos de f_a já vimos na secção 4.3.1 que basta escrever:

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):

> solve(g(x)=x,x);
```

que obtemos os pontos:

$$0,\ \frac{a-1}{a}$$

De modo análogo, para determinar os pontos fixos de f_a^2 escrevemos

```
> solve((g@@2)(x)=x,x);
```

e obtemos quatro soluções, que são:

$$0,\ \frac{a-1}{a},\ \frac{\frac{a}{2}+\frac{1}{2}+\frac{\sqrt{-3-2\,a+a^2}}{2}}{a},\ \frac{\frac{a}{2}+\frac{1}{2}-\frac{\sqrt{-3-2\,a+a^2}}{2}}{a}$$

Uma vez que os pontos periódicos de período dois de uma função são aqueles que são solução da equação $f^2(x) = x$ e que não solução da equação $f(x) = x$, o aluno pode concluir que $x_1 = \frac{a+1+\sqrt{(a-3)(a+1)}}{2a}$ e $x_2 = \frac{a+1-\sqrt{(a-3)(a+1)}}{2a}$ são as expressões gerais dos pontos periódicos de período dois da aplicação logística. Ao determinar os pontos periódicos de período dois por este método, deve ser pedido ao aluno que discuta a existência destes pontos em $\mathbb{R}$ com a variação do parâmetro a. Facilmente, o aluno concluirá que para x_1 e x_2 serem reais $(a-3)(a+1) > 0$, ou seja, o parâmetro a deve tomar valores maiores do que três, uma vez que estamos a considerar valores positivos para o parâmetro a. Agora, o professor pode justificar o uso do comando

```
> assume(a>3):
```

para determinar a expressão geral dos pontos periódicos de período dois com o procedimento **perdois**.

Comando fsolve

Para determinarmos numericamente os pontos periódicos de período dois da logística, vamos recorrer ao procedimento perdoisf que foi apresentado na secção 2.3.2. Este procedimento, recorrendo ao comando **fsolve** indica os pontos periódicos de período dois para uma função f, para valores de x entre $xmin$ e $xmax$. Uma vez que este comando resolve numericamente uma equação é preciso definir o parâmetro a para o qual queremos determinar os pontos periódicos de f_a. Se escolhermos, por exemplo, $a = 3.6$ para determinar os pontos periódicos de período dois de $f_{3.6}$ no intervalo $[0, 1]$, devemos escrever:

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):

> a:=3.6:

> perdoisf(g,0,1);
```

Como resposta obtemos

$$[0.4081493371,\ 0.8696284406]$$

No seguimento do que se disse para o comando **solve**, o procedimento **perdoisf** pode ser usado pelo aluno para determinar os pontos periódicos de período dois de uma aplicação fixada da família logística, sempre que se fixe o parâmetro a.

4.4.2 Método gráfico

Nesta secção vamos usar o procedimento perdoisplot apresentado na secção 2.3.2 para determinar graficamente os pontos periódicos de período dois da aplicação logística. Na secção anterior vimos que o comando **fsolve** responde prontamente e sem qualquer entrave quando é usado para determinar os pontos periódicos de período dois para um parâmetro a previamente fixado. Assim, podemos usar a versão modificada do procedimento **perdoisplot**, ou seja, a versão que nos dá a indicação no gráfico dos pontos periódicos. Esse procedimento tem como parâmetros de entrada a função f, e o intervalo do domínio a considerar $[xmin, xmax]$.

Ao escolhermos, por exemplo, 3.5 para o parâmetro a da aplicação logística e escrevermos as instruções:

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):

> a:=3.5:

> perdoisplot(g,0,1);
```

Com estas instruções obtemos o gráfico da figura 4.8.

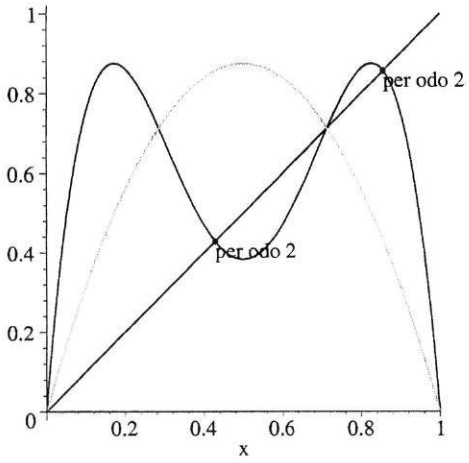


Figura 4.8: **perdoisplot** $f_{3.5}$ com $xmin = 0$ e $xmax = 1$

Analisando esse gráfico, o aluno pode observar que $f_{3.5}$ tem dois pontos periódicos de período dois: um situa-se no intervalo $[0.4; 0.5]$ e o outro no intervalo $[0.8; 1]$.

Para obtermos um dos pontos periódicos com maior precisão, podemos usar de novo o procedimento **perdoisplot**, mas agora com outros valores para $xmin$ e $xmax$ para variação do domínio. Se quisermos, por exemplo, determinar com maior precisão a localização do ponto periódico que está no intervalo $[0.8; 1]$ podemos começar por enquadrar o gráfico da função entre 0.8 e 1. Para isso, escrevemos:

```
> perdoisplot(g,0.8,1);
```

Obtendo o gráfico da figura 4.9.

Observando esse gráfico, o aluno conclui que o ponto periódico se situa no intervalo

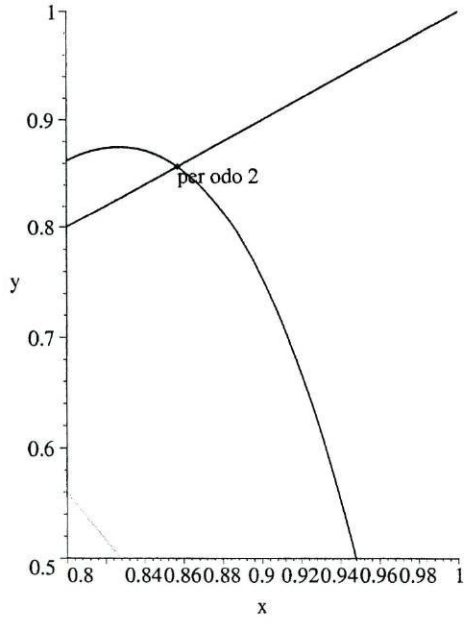


Figura 4.9: **perdoisplot** $f_{3.5}$ com $xmin = 0.8$ e $xmax = 1$

$[0.84; 0.88]$. Se quiser obter com maior precisão este ou outro ponto periódico, basta escolher para $xmin$ e $xmax$ os valores entre os quais se pretende enquadrar o ponto periódico.

Uma vez que o procedimento perdoisanima, apresentado na secção 3.4.2, nos permite ter uma visão mais global acerca da variação dos pontos periódicos de período dois, podemos recorrer a ele para estudar os pontos periódicos da aplicação logística. Como este procedimento produz o gráfico de f a cor amarela e os gráficos de f^2 e da identidade a cor preta é fácil identificar os pontos que resultam só da intersecção dos gráficos de cor preta, ou seja, os pontos periódicos de período dois.

Na tabela 4.9 exibimos algumas representações gráficas intermédias desta animação para valores de a entre 0.6 e 3.99 e com $xmin = 0$ e $xmax = 1$.

Através da análise dos gráficos da tabela 4.9, o aluno pode concluir que:

- Se $0 \leq a \leq 3$, f_a não tem pontos periódicos de período dois, pois não há nenhum ponto do gráfico onde só se intersectem gráficos representados a cor preta;

- Se $3 \leq a \leq 4$, f_a tem dois pontos periódicos de período dois que são os assinalados

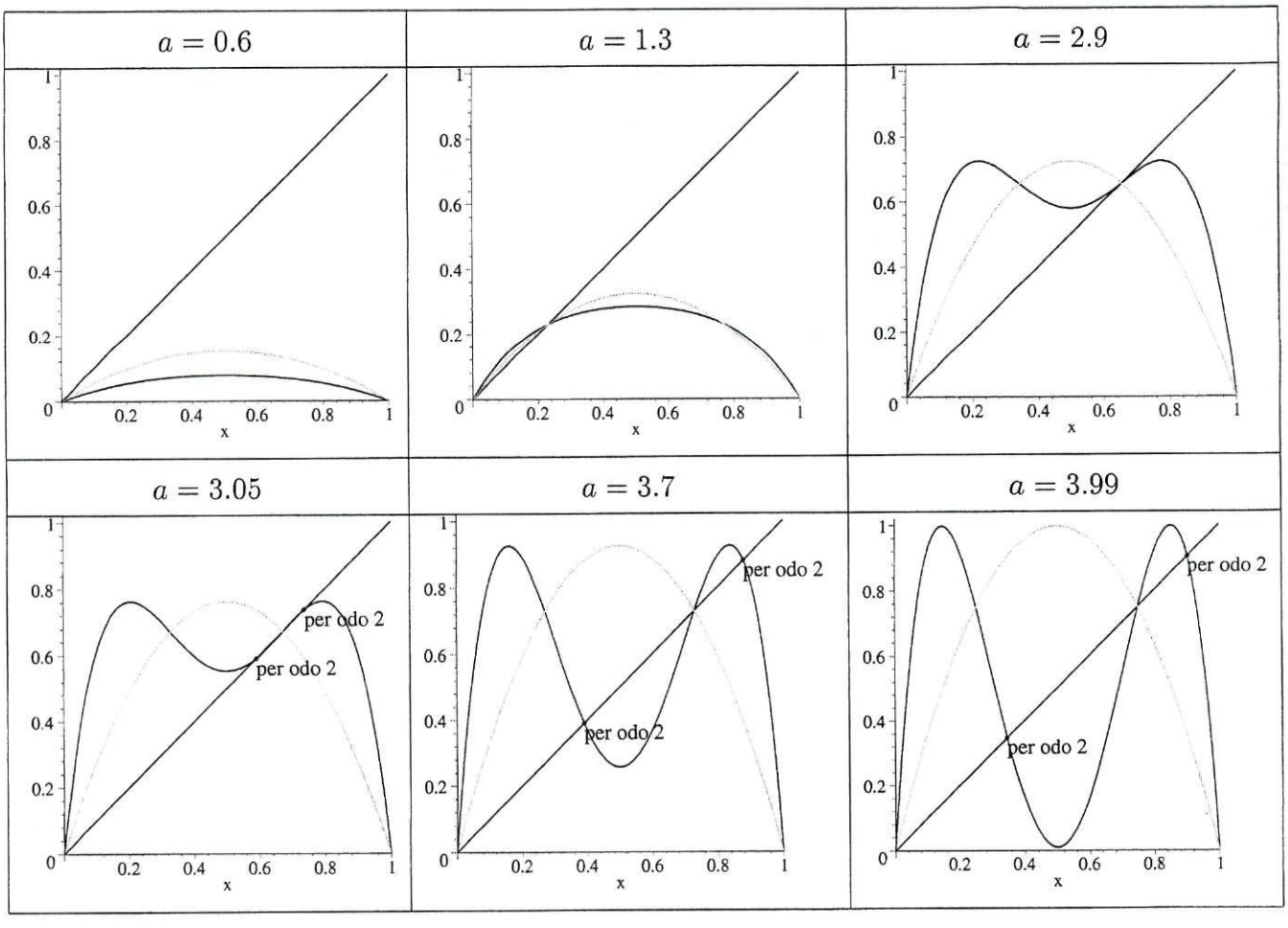


Tabela 4.9: Pontos periódicos de f_a para diferentes valores de a.

em cada um dos gráficos.

4.5 Período três

4.5.1 Métodos analíticos

É de esperar que neste caso a utilização do Maple usando estratégias similares às usadas para os pontos periódicos de período dois não seja tão satisfatória e exija da parte do professor uma maior atenção para que o aluno não seja confrontado com situações pouco claras e que têm a ver com dificuldades algorítmicas intrínsecas ao Maple. Lembramos ainda que para equações de grau igual ou superior a cinco não há fórmula resolvente. Por-

tanto, todos os métodos que tentem induzir soluções têm falhas e o Maple não "escapa" a esta realidade.

Comando solve

Para determinar os pontos periódicos de período três, seguindo estratégias anteriores, somos naturalmente levados a usar o procedimento pertres já usado na secção 3.5.1. Numa tentativa de determinar a expressão dos pontos periódicos de período três para um parâmetro a da aplicação logística podemos escrever:

```
> f:=(a,x)->a*x*(1-x):
> g:=x->f(a,x):
> pertres(g);
```

e como resposta obtemos:

[]

Como já aconteceu na secção 3.5.1, quando se discutia a existência dos pontos periódicos de período três, também aqui obtemos uma lista vazia. Em princípio, poderíamos pensar que com a escolha acertada de um certo a_0, o comando **assume**$(a > a_0)$, o procedimento **pertres** daria a lista dos pontos periódicos de período três, pelo menos para os valores do parâmetro a onde sabemos que existem (na secção 4.5.2 veremos que para $a > 3.8$ a aplicação tem pontos periódicos de período três). Contudo, devido à complexidade da equação polinomial de grau oito e do parâmetro a, ao escrevermos

```
> assume(a>3.8):
```

obtém-se uma lista vazia.

[]

Assim, o procedimento **pertres** e estratégias similares não devem ser usadas para determinar os pontos periódicos de período três.

Para reforçar esta ideia de que os métodos algébricos não devem ser usados vamos, sem usar o procedimento **pertres**, recorrer ao comando **solve**, para averiguar as respostas fornecidas pelo Maple.

Para determinar as raízes da equação $f_a(x) = x$ escrevemos:

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):

> solve(g(x)=x,x);
```

e obtemos

$$0, \frac{-1+a}{a}$$

Para determinar as raízes da equação $f_a^3(x) = x$ escrevemos os comandos:

```
> solve((g@@3)(x)=x,x);
```

e obtemos

$$0, \frac{-1+a}{a}, \text{RootOf}(a^2+a+1+(-2\,a^2-2\,a-1-a^3)\,_Z+(3\,a+1+2\,a^3+3\,a^2)\,_Z^2$$
$$+(-1-3\,a-5\,a^2-a^3)\,_Z^3+(1+4\,a+3\,a^2)\,_Z^4+(-1-3\,a)\,_Z^5+_Z^6,$$
$$label = _L1)/a$$

A resposta obtida fornece umas expressões que são estranhas para o aluno e que têm o significado descrito na secção 1.8. Fica, então visto que com os métodos usados, não é possível determinar a expressão geral dos pontos periódicos de período três da família logística. Isto já devia ser conhecido do professor,uma vez que o problema conduzia a determinar soluções de equações polinomiais de grau oito.

Fixando um parâmetro a, se este for interpretado internamente pelo Maple como numérico (do tipo float) a resposta não é a que um aluno do Secundário estaria à espera. Por exemplo, se $a = \frac{39}{10}$, ao escrevermos:

```
> a:=39/10:

> pertres(g);
```

Obtemos:

$$[\frac{1}{39}\,\text{RootOf}(\%1,\ index = 1),\ \frac{1}{39}\,\text{RootOf}(\%1,\ index = 6),\ \frac{1}{39}\,\text{RootOf}(\%1,\ index = 5),$$
$$\frac{1}{39}\,\text{RootOf}(\%1,\ index = 2),\ \frac{1}{39}\,\text{RootOf}(\%1,\ index = 3),\ \frac{1}{39}\,\text{RootOf}(\%1,\ index = 4)]$$
$$\%1 := _Z^6 - 127\,_Z^5 + 6223\,_Z^4 - 148069\,_Z^3 + 1769680\,_Z^2 - 9853900\,_Z + 20110000$$

Como se pode observar pela sintaxe da resposta, as soluções são dadas em função de um símbolo, ou seja, esta não é um a resposta razoável para apresentar a um aluno do Secundário. Contudo, se em vez de escrevermos

```
> a:=39/10:
```

escrevermos

```
> a:=3.9:
```

O resultado obtido é:

[0.4487177532, 0.9512131776, 0.9647435115, 0.1326525271, 0.1809860065, 0.5780972805]

A diferença crucial de resposta ao problema de determinar os pontos periódicos de período três, reside no facto de se ter fixado o parâmetro a como sendo **numérico**. Assim, os comandos solve do procedimento **pertres** são executados como se tratasse do comando **fsolve**. Como já se viu na secção 1.8, internamente o comando **fsolve** possui algoritmos que permitem obter aproximações de todas as raízes de uma equação polinomial e por consequência o procedimento **pertres** também nos dá a aproximação numérica de todos os pontos periódicos de período três.

Comando fsolve

Para determinar numericamente os pontos periódicos de período três da aplicação logística, podemos usar o procedimento pertresf que já foi abordado na secção 2.3.2. Este procedimento tem como parâmetros de entrada a função f em estudo e o intervalo $[xmin, xmax]$ onde queremos estudar a existência de pontos periódicos de período três.

Escolhendo de novo o parâmetro $a = 3.9$, para determinar os pontos periódicos de período três no intervalo $[0, 1]$ e escrevendo:

```
> a:=3.9:
```

```
> pertresf(g,0,1);
```

obtemos

[0.1326525274, 0.1809860054, 0.4487177896, 0.5780971634, 0.7435899013, 0.9512126972, 0.9647439150]

Podemos verificar se este resultado é condizente com o obtido, recorrendo ao procedimento **listaorbita**. Comecemos por ver os cinco primeiros elementos da órbita do ponto 0.1326525274:

```
> listaorbita(f,0.1326525274,0,5);
```

[0.1326525274, 0.4487177541, 0.9647435119, 0.1326525258, 0.4487177495, 0.9647435100]

Os resultados obtidos indicam que $\{0.1326525274, 0.4487177541, 0.9647435119\}$ formam uma órbita periódica de período três.

Vejamos agora os cinco primeiros elementos da órbita do ponto 0.1809860054:

```
> listaorbita(f,0.1809860054,0,5);
```

[0.1809860054, 0.5780972779, 0.9512131793, 0.1809860006, 0.5780972659, 0.9512131865]

Os resultados indicam que $\{0.1809860054, 0.5780972779, 0.9512131793\}$ é uma órbita periódica de período três.

Os resultados obtidos parecem indicar que 0.7435899013 não é um ponto periódico de período três. De facto,

```
> listaorbita(f,0.7435899013,0,5);
```

[0.7435899013, 0.7435894439, 0.7435903129, 0.7435886618, 0.7435917990, 0.7435858383]

Olhando para as soluções apresentadas, facilmente se pode concluir que não estão correctas, uma vez que o número de pontos periódicos de período três é sempre um múltiplo de três e o Maple apresenta sete soluções. De facto, o procedimento **pertresf** recorre ao comando **fsolve** que trabalha, por defeito, com aproximações de números a dez casas decimais. Neste caso, esta aproximação mostra-se insuficiente para dar uma resposta correcta e, se fixarmos o número de casas decimais em quinze:

```
> Digits:=15:
```

ao executarmos de novo o procedimento **pertres** para determinar os pontos periódicos de período três de $f_{3.9}$

```
> a:=3.9:
> pertresf(g,0,1);
```

obtemos os seis pontos periódicos de período três de $f_{3.9}$.

[0.132652527098159, 0.180986006451061, 0.448717753195260, 0.578097280487834, 0.951213177643578, 0.964743511534365]

Para garantir a existência de pontos periódicos de período três, o professor deve ter o cuidado de previamente definir que os cálculos deverão ser efectuados com pelo menos quinze casas decimais. Note-se que se quisermos trabalhar com períodos maiores este número deverá ser aumentado em conformidade (recorrendo à experimentação).

4.5.2 Método gráfico

Vamos agora recorrer ao procedimento pertresplot para determinar graficamente os pontos periódicos de período três da logística. Este procedimento produz no mesmo sistema de eixos o gráfico de f, de f^3 e da identidade. Na secção anterior vimos que o comando **fsolve** nem sempre funciona da melhor forma da primeira vez que se usa. Assim, se não modificarmos os valores que o Maple apresenta por defeito, no início de cada sessão, nomeadamente o número de casas decimais, a identificação dos pontos periódicos de período três pode ter erros. Então, para obtermos, por exemplo para $a = 3.9$, devemos escrever

```
> f:=(a,x)->a*x*(1-x):
> g:=x->f(a,x):
> a:=3.9:
> Digits:=15:
> pertresplot(g,0,1);
```

Com estas instruções obtemos o gráfico da figura 4.10.

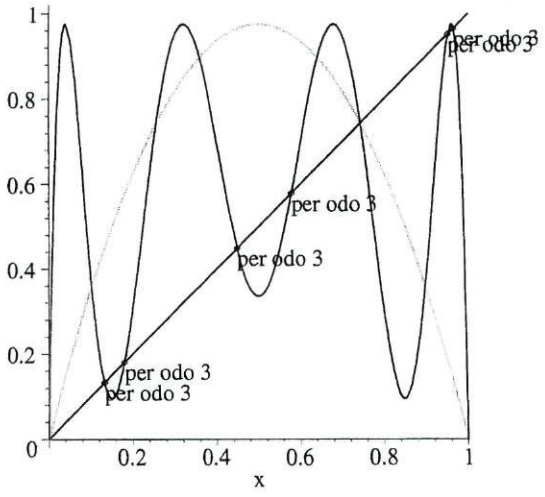


Figura 4.10: **pertresplot** de $f_{3.9}$ com $xmin = 0$ e $xmax = 1$

Pela análise deste gráfico, o aluno pode observar que existem seis pontos onde só se intersectam gráficos de cor preta. Como os gráficos que estão desenhados a preto são os de $f^3_{3.9}$ e da identidade, $f_{3.9}$ tem seis pontos periódicos de período três.

Tal como temos vindo a fazer, podemos usar uma animação para visualizar a variação dos pontos periódicos de período três com a variação do parâmetro a. Para tal, recorremos ao procedimento pertresanima. Os parâmetros de entrada deste procedimento são f que é a função em estudo, *amin* e *amax* que indicam, respectivamente os valores mínimo e máximo do parâmetro a e *npontos* que é número de pontos em que se divide o intervalo $[amin, amax]$.

Na tabela 4.10 apresentamos alguns dos gráficos intermédios que podem ser obtidos com o procedimento **pertresanima**. Para o usarmos temos de definir previamente a função $g(x) = f(a, x)$, o intervalo de variação do parâmetro a e o número de pontos a usar.

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):

> pertresanima(g,0,4,400);# exemplo em que 0<a<4 e npontos=400
```

Com base nos gráficos expressos na tabela 4.10, o aluno pode concluir que:

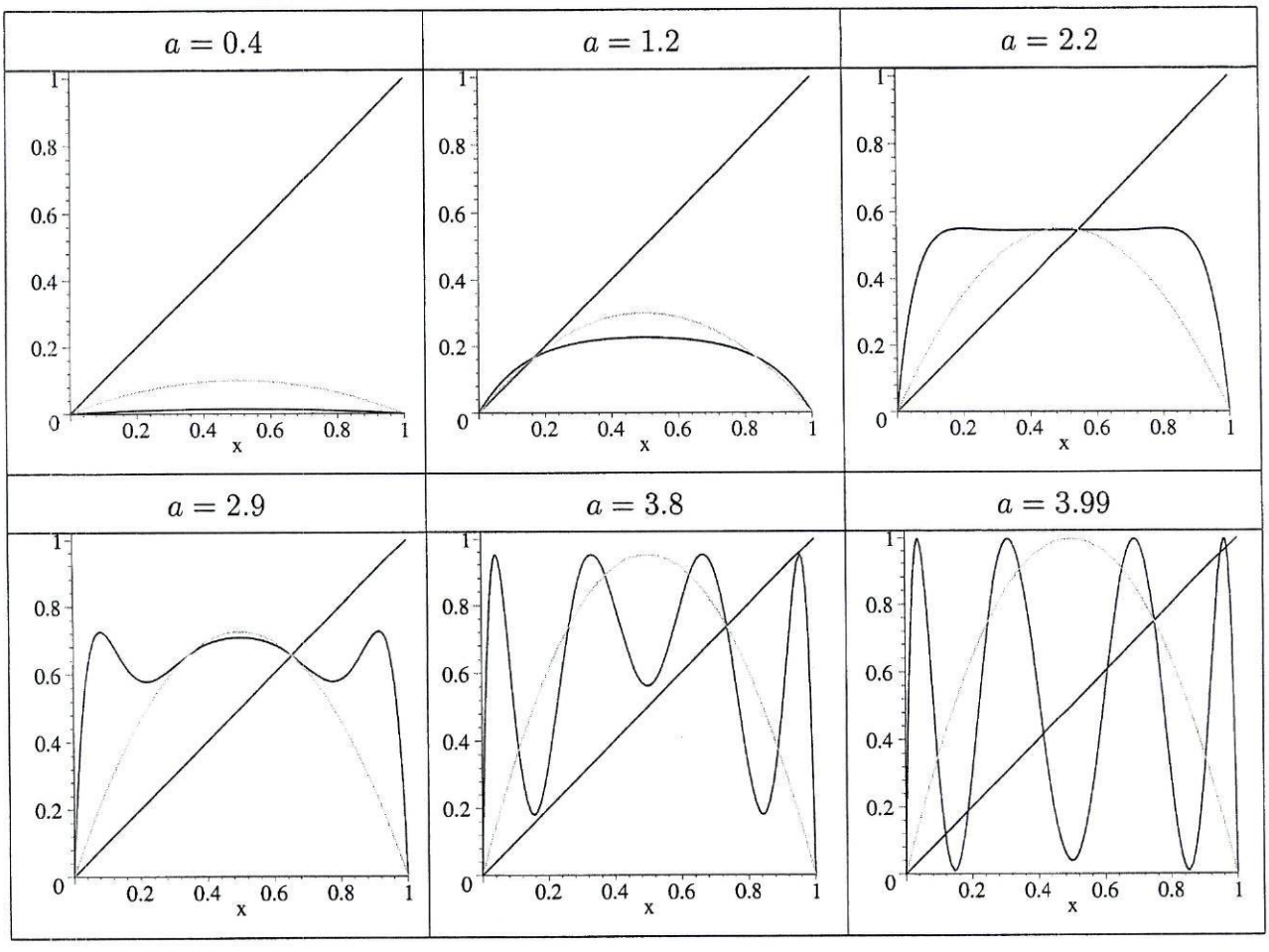


Tabela 4.10: Pontos periódicos de período 3 de f_a para diferentes valores de a.

- Para valores de $a < 3.8$ a aplicação logística não tem pontos periódicos de período três;

- Para $a > 3.8$ a aplicação tem seis pontos periódicos de período três.

4.6 Período arbitrário

Nesta secção, em face do que se viu na secção anterior, o método gráfico é o mais eficaz, pelo menos numa abordagem mais elementar, para determinar os pontos periódicos de período três. Assim, para determinar graficamente os pontos periódicos de período n da aplicação logística podemos usar o procedimento pernplot que apresentamos na

secção 2.3.2. Para determinarmos graficamente os pontos periódicos de período quatro da aplicação logística para, por exemplo $a = 3.75$, $xmin = 0$ e $xmax = 1$, devemos escrever

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):

> a:=3.75:

> pernplot(g,4,0,1);
```

Com estes comando obtemos o gráfico expresso na figura 4.11.

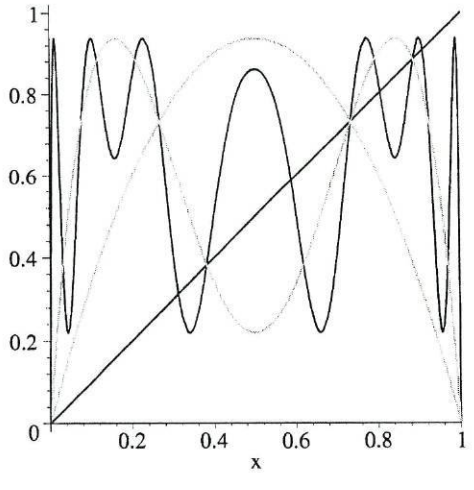


Figura 4.11: pontos periódicos de período 4 de $f_{3.75}$

Pela análise do gráfico da figura 4.11 o aluno conclui que $f^4_{3.75}$ tem quatro pontos periódicos de período quatro, que correspondem aos pontos onde só há intersecção dos gráficos representados a cor preta.

Para determinar com alguma precisão algum ou alguns dos pontos periódicos podemos usar o procedimento **pernplot** com o enquadramento para o eixo dos xx desejado. Se quisermos, por exemplo, aumentar a exactidão do ponto periódico de período quatro que está no intervalo $[0.3; 0.4]$, devemos escrever

```
> pernplot(g,4,0.3,0.4);
```

e obtemos o gráfico da figura 4.12

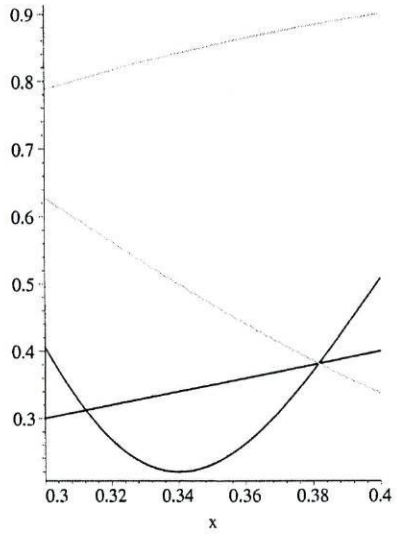


Figura 4.12: ponto periódico de período 4 de $f_{3.75}$ em $[0.3, 0.4]$

Analisando a figura 4.12 o aluno pode concluir que o ponto periódico se situa no intervalo $[0.31; 0.32]$.

Mais uma vez podemos constatar que o método gráfico é um método bastante eficaz na determinação de pontos periódicos e que é a ele que se deve dar preferência para iniciar o estudo de uma órbita.

4.7 Bifurcação das órbitas periódicas

Para mostrar o comportamento periódico, pelo menos para períodos baixos, vimos nas secções anteriores maneiras de o conseguir. Contudo, já quando se fez a análise de órbitas na secção 4.2.4, se concluiu que pelos resultados obtidos existiam casos em que estávamos em presença de órbitas não periódicas ou de período muito grande. Nestes casos, faz então sentido tentar aproximar essas órbitas, ou seja, aproximar o seu ω-limite. Na ajuda dessa aproximação do ω-limite, já construímos na secção 4.2.4 o procedimento omegalimite.

Nesta secção vamos apresentar um procedimento que nos mostra como varia o ω-limite com o parâmetro a. O aluno pode ser então confrontado com o procedimento conjlimite

que para um valor de a e x_0 fixados nos dá uma aproximação do ω-limite de x_0 para a função f_a. Este procedimento como parâmetros de entrada tem f, a, x_0, m e n, em que:

- f é a função a estudar;
- a é o parâmetro escolhido;
- x_0 é o ponto inicial;
- m é o tempo a partir do qual queremos visualizar os iterados;
- n é o último iterado a visualizar.

O procedimento é constituído pelas seguintes instruções:

```
> conjlimite:=proc(f,a,x0,m,n)
  local i,x:
  with(plots):
  x[0]:=x0:
  for i from 1 to n do
  x[i]:=f(x[i-1]):
  od:
  pointplot([seq([a,x[i]],i=m+1..n)],view=[a-1..a+1,0..1]);
  end proc:
```

Com este procedimento são calculados n elementos da órbita e representados graficamente os pares de pontos da forma (a, x_j) para j entre m e n, ou seja, este procedimento fornece uma aproximação gráfica da órbita de x_0 pela função f_a. Para esse efeito devemos escolher um valor elevado para m e um valor ainda maior para n.

Na tabela 4.11 exibimos os resultados obtidos com a aplicação logística para $x_0 = 0.2$, $m = 100$ e $n = 200$ para diferentes valores do parâmetro a. Para obter cada uma das representações gráficas desta tabela, devemos primeiro definir a função $g = f(a, x)$ (conforme justificado na secção 4.4).

```
> T:=(a,x)->a*x*(1-x):
```

```
> g:=x->T(a,x):
```

e em seguida escrever:

```
> conjlimite(g,a,0.2,100,200);#a é substituído pelo valor do parâmetro
a considerar
```

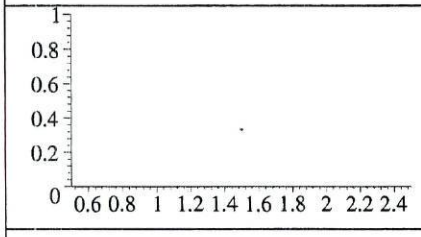
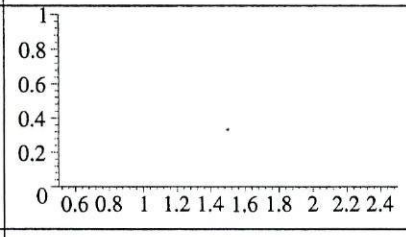
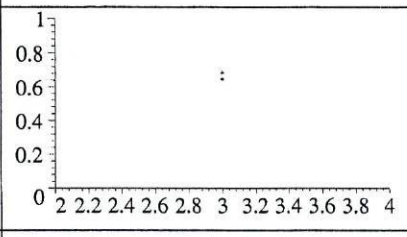
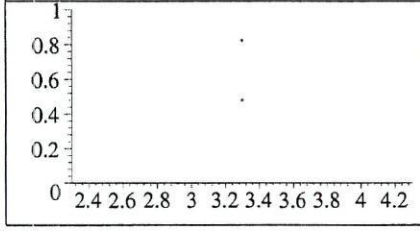
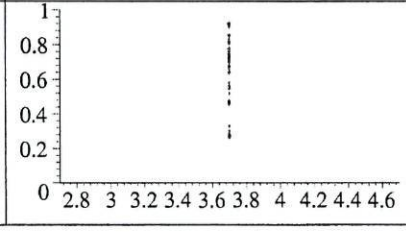
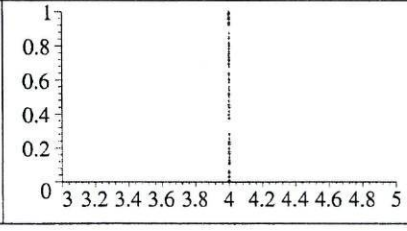
$a = 1.5$	$a = 2.3$	$a = 3$
		
$a = 3.3$	$a = 3.7$	$a = 4$
		

Tabela 4.11: ω-limite de várias órbitas para a entre 1 e 4

Uma vez que na secção 4.2.4 já vimos que se $a < 1$ o ω-limite da órbita de qualquer ponto inicial é zero, parece-nos que não há necessidade de exibir exemplos com $a < 1$. Com base na tabela 4.11, o aluno pode concluir que nos dois primeiros gráficos, x_0 converge para um ponto fixo e nos dois gráficos seguintes, x_0 converge para pontos periódicos de período dois. Além disso, também se pode pedir ao aluno que experimente atribuir outros valores a x_0. Ao executar o pretendido, aperceber-se-á que se obtém a mesma figura e assim concluirá que o ω-limite não depende do ponto inicial considerado. Testando mais valores para o parâmetro a, poderá concluir que:

- para $a < 3$ a órbita converge para um ponto fixo;
- para $a = 3$ e $a = 3.3$ a órbita converge para um ponto periódico de período dois.

Confrontando o aluno com as figuras obtidas para $a = 3.7$ e $a = 4$ e testando ainda o procedimento para diferentes valores de a e x_0, facilmente concluirá que o tipo de figuras obtido é similar ao apresentado e portanto conjectura que o ω-limite da órbita é o mesmo

e não depende do valor inicial.(Claro que os casos em que $x_0 = 0$ e $x_0 = 1$ são excepções.) Além disso, também podemos levá-lo a concluir que para $a > 3.7$, os resultados apontam para que não haja convergência para órbitas periódicas.

Depois do aluno estar familiarizado com esta aplicação, para diferentes valores de a e x_0, e de saber interpretar os resultados obtidos, distinguindo as órbitas periódicas e as não periódicas, podemos confrontá-lo com o clássico diagrama de bifurcação. Neste diagrama, fixando um valor para x_0 , fazemos a representação gráfica do ω-limite de x_0 para f_a em que a varia entre 0 e 4. Para o efeito, criamos o procedimento o bifurcplot que tem como parâmetros de entrada f, a_0, a_1, *npontos*, m, n, x_0, y_0 e y_1, em que:

- f é a função a estudar;
- a_0 é o valor mínimo do parâmetro a;
- a_1 é o valor máximo do parâmetro a;
- *npontos* é o número de pontos em que se divide o intervalo $[a0, a1]$;
- m é o tempo a partir do qual se quer visualizar a órbita;
- n é o iterado final para a visualização;
- x_0 é o ponto inicial;
- y_0 é o valor mínimo para o eixo dos yy;
- y_1 é o valor máximo para o eixo dos yy.

Para cada valor de a são calculados n elementos da órbita são desenhados, para cada valor de a, os pontos da forma (a, x_j) com $j = m..n$.

```
> bifurcplot:=proc(f,a0,a1,npontos,m,n,x0,y0,y1)
  local t,a,x,bifurc;
  bifurc:=NULL:
  with(plots):
  for a from a0 to a1 by (a1-a0)/npontos do
  x[0]:=x0;
  for t from 0 to n do
  x[t+1]:=f(a,x[t]):
  od:
  bifurc:=bifurc,plot([[a,x[s]]$s=m..n],a0..a1,y0..y1,color=black,
  style=point,symbol=POINT):
  end do:
  display(bifurc);
  end proc:
```

O primeiro bloco de instruções serve para determinar os primeiros n elementos da órbita de x_0.

```
> bifurcplot:=proc(f,a0,a1,npontos,m,n,x0,y0,y1)
  local t,a,x,bifurc;
  bifurc:=NULL:
  with(plots):
  for a from a0 to a1 by (a1-a0)/npontos do
  x[0]:=x0;
  for t from 0 to n do
  x[t+1]:=f(a,x[t]):
  od:
```

O bloco de instruções que finaliza o procedimento serve para esboçar os pontos da forma (a, x_j) com j entre m e n.

```
> bifurc:=bifurc,plot([[a,x[s]]$s=m..n],a0..a1,y0..y1,color=black,
  style=point,symbol=POINT):
  end do:
  display(bifurc);
  end proc:
```

Para vermos o resultado obtido com $a_0 = 0$, $a_1 = 4$, npontos=400, $m = 100$, $n = 500$,

$x_0 = 0.1$, $y_0 = 0$ e $y_1 = 1$, devemos escrever as seguintes instruções:

```
> f:=(a,x)->a*x*(1-x):

> bifurcplot(f,0,4,400,100,500,0.1,0,1);
```

O resultado é o que se exibe na figura 4.13.

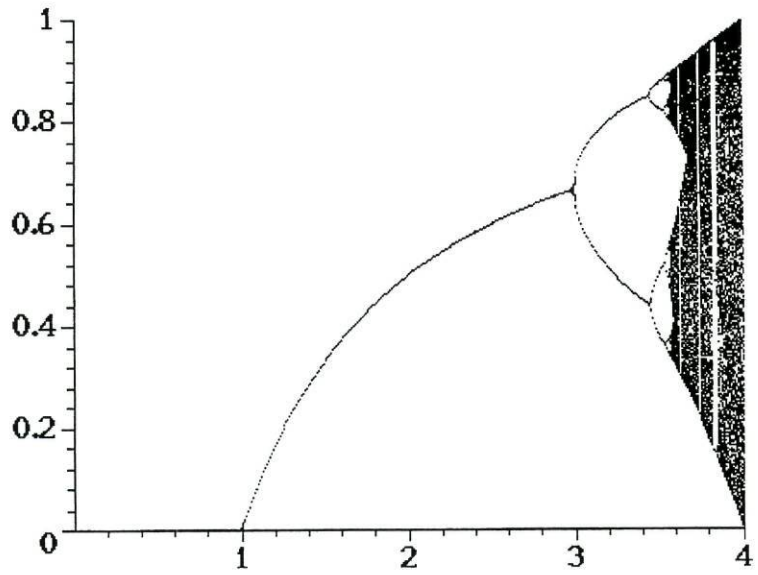


Figura 4.13: Diagrama de bifurcação da logística

Através deste diagrama o aluno pode observar que:

- Para $0 < a < 1$, as órbitas convergem para 0;
- Para $1 < a < 3$, as órbitas convergem para um ponto fixo que depende do parâmetro a considerado;
- Para $a = 3$, dá-se a primeira bifurcação, ou seja, há uma mudança de comportamento no ω-limite das órbitas: para $a < 3$ o ω-limite é um ponto fixo e para $a > 3$ é uma órbita periódica de período dois;
- Para a perto de 3.5 há uma nova bifurcação;

- Para valores de a superiores a 3.8 o diagrama não fica perceptível com este enquadramento para o parâmetro.

É de referir que executando o mesmo procedimento com a aplicação tenda

```
> T:=(a,x)->piecewise(0<=x and x<1/2,a*x,1/2<=x and x<=1,a*(1-x)):
> bifurcplot(T,0,2,200,400,500,0.1,0,1);
```

obtemos o diagrama da figura 2.5.

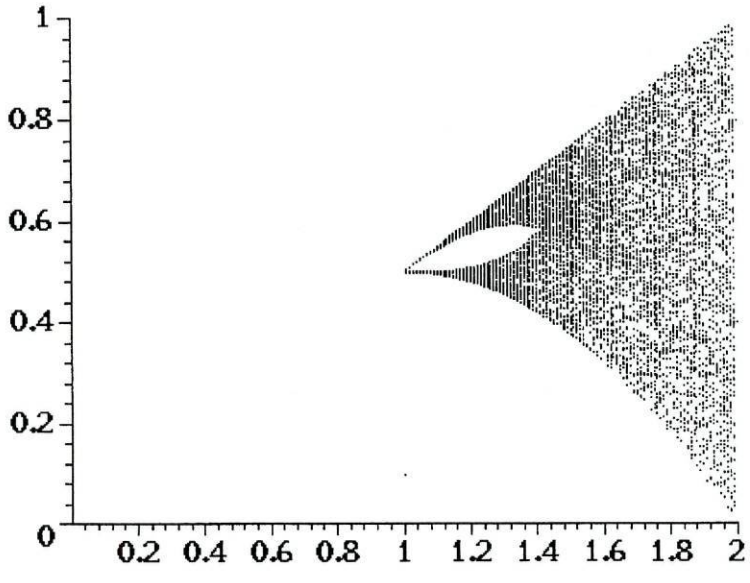


Figura 4.14: Diagrama de bifurcação da tenda

Este diagrama não se assemelha ao diagrama da bifurcação da logística. De facto, para os casos em que $a > 1$ $|T'_a(x)| = a$, $\forall x$ é sempre maior do que um, o que significa que há sensibilidade às condições iniciais. Na aplicação tenda, as órbitas periódicas são sempre do tipo repulsor, contrariamente ao que acontece na logística. Assim, considerando um elevado número de iterados, $|T^n_a(x)|$ cresce exponencialmente, ou seja, qualquer que seja a escolha de dois valores iniciais próximos (x e y), se iterarmos muitas vezes, $T^n_a(x)$ e $T^n_a(y)$ estarão sempre afastados.

4.8 Constante de Feigenbaum

Feingenbaum em [3] ao aplicar iterativamente a função logística $x_{n+1} = ax_n - ax_n^2$ para diferentes valores de a, mostrou que para a aplicação $f_a(x) = ax(1-x)$ existe uma sequência de valores $\{a_1, a_2, \ldots, a_n, a_{n+1}\}$ tal que que se $0 < a < a_1$, f_a tem um ponto periódico de período um que é o ω-limite de todo o ponto $x \in]0,1[$, se $a_1 < a < a_2$, f_a tem uma órbita periódica de período dois que é o ω-limite de todo o ponto $x \in]0,1[$ e que se $a_n < a < a_{n+1}$, f_a tem uma órbita periódica de período 2^n que é o ω-limite de todo o ponto $x \in]0,1[$. Ao determinar o valor de $\frac{a_{n-1}-a_{n-2}}{a_n-a_{n-1}}$, concluiu que à medida que n aumentava, esse quociente se aproximava cada vez mais de 4,669201609, valor que ficou conhecido como **constante de Feigenbaum**.

Para conduzir o aluno a obter numericamente os primeiros termos a_n da sequência dos parâmetros onde se dão as bifurcações (e assim, verificar a convergência daquele quociente para a constante de Feigenbaum), o professor deve proceder a uma análise analítica e gráfica desses parâmetros.

Como já vimos anteriormente, para $a = 3$, f_a tem um ponto fixo. Para $a > 3$ surgem dois novos pontos fixos de f^2 que atraem todas as órbitas (Teorema de Singer, ver [1]. Para determinar analiticamente uma aproximação para o valor do parâmetro a a partir do qual passam a existir estes novos pontos fixos de f^2, devemos começar por definir a função f_a.

```
> f:=(a,x)->a*x*(1-x):
```

Depois de definida a função f_a, determinamos as expressões algébricas de $f_a(x)$ e da derivada de $f_a(x)$. Para isso escrevemos:

```
> r1:=expand(f(x));#Expressão de f
```

```
> r2:=diff(r1,x);#Derivada de f
```

Obtemos

$$r1 := a\,x - a\,x^2$$

$$r2 := a - 2\,a\,x$$

Pela observação do diagrama de bifurcação, o aluno percebe que a bifurcação se dá

perto de $a = 3$, que é o valor exacto. Assim, para determinar o valor do parâmetro onde ocorre a bifurcação, apenas observando o diagrama de bifurcação basta testar valores de a entre 2.9 e 3.1. Quando se dá a bifurcação, a derivada de f_a é -1. Portanto, ao escrevermos:

```
> fsolve({r1=x,r2=-1},{a,x},{a=2.9..3.1,x=0.5..1});#Determinação
  dos pontos fixos de f que têm derivada -1
```

Obtemos o valor do parâmetro a a partir do qual se dá a primeira bifurcação.

$$\{a = 3.000000000,\ x = 0.6666666667\}$$

O aluno obtém, então que a partir de $a = 3$ existe uma órbita periódica de período dois.

Com o procedimento **bifurcplot** é possível visualizar com algum pormenor essa bifurcação. Para isso, é necessário fazer o devido enquadramento para os eixos vertical e horizontal. Na figura 4.15 podemos visualizar a bifurcação que se dá para $a = 3$. Esta figura foi obtida com a instrução:

```
> bifurcplot(f,2.9,3.1,160,800,1200,0.1,0.5,1);
```

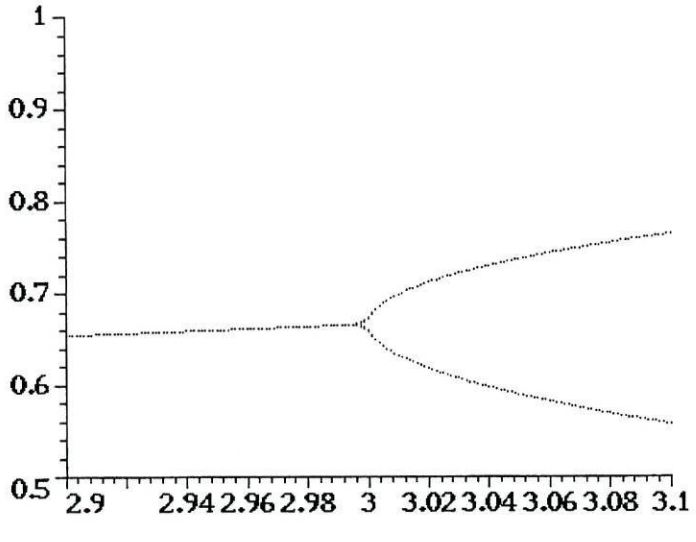


Figura 4.15: Bifurcação em $a = 3$ obtida com o procedimento **bifurcplot**

Com o procedimento bifurcplot, o aluno pode testar diferentes valores para os parâmetros

de entrada para identificar graficamente a bifurcação em $a = 3$.

Mas, até quando é que se verifica este comportamento? Ou seja, até que parâmetro a, f_a tem dois pontos periódicos (com órbita periódica de período dois) atractores? Pela análise do diagrama de bifurcação, o aluno facilmente indica que para a perto de 3.5 essa órbita periódica deixa de existir, pois dão-se duas novas bifurcações. Na tabela da figura 4.12 exibimos representações gráficas dos pontos periódicos de período quatro para valores de a entre 3.4 e 3.6. Para obtermos cada uma das imagens, recorremos ao procedimento pernplot, escrevendo as seguintes instruções:

```
> f:=(a,x)->a*x*(1-x):
> g:=x->f(a,x):
> a:=#valor do parâmetro considerado
> pernplot(g,4,0,1);
```

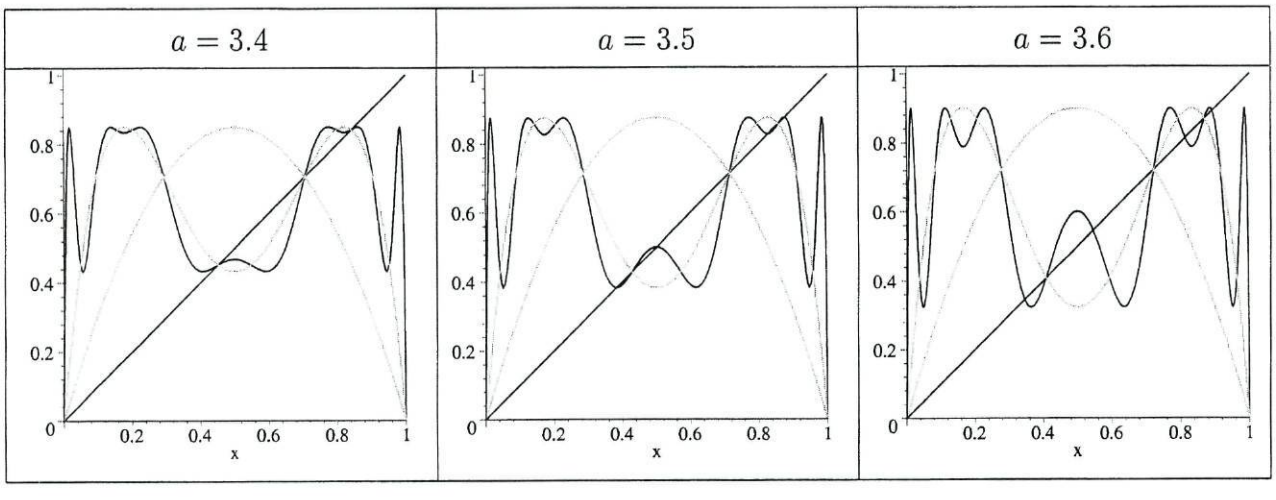


Tabela 4.12: Pontos periódicos de período quatro de f_a com a entre 3.4 e 3.6

Os gráficos apresentados na tabela 4.12 podem servir para levar o aluno a concluir que entre os valores considerados para o parâmetro a, surgem novos pontos fixos de f^4, que não são pontos fixos de f nem de f^2 e que portanto há uma nova bifurcação para pontos periódicos de período quatro.

Para determinar analiticamente o valor do parâmetro a, a partir do qual se verifica tal situação, procedemos de modo análogo ao que fizemos para a primeira bifurcação. Depois

de definida a função f_a, determinamos as expressões algébricas de $f_a^2(x)$ e da derivada de $f_a^2(x)$, escrevendo:

```
> r1:=expand((f@@2)(x));#Expressão de f^2

> r2:=diff(r1,x);#Derivada de f^2
```

Com estas instruções, obtemos:

$$r1 := a^2\,x - a^3\,x^2 + 2\,a^3\,x^3 - a^2\,x^2 - a^3\,x^4$$

$$r2 := a^2 - 2\,a^3\,x + 6\,a^3\,x^2 - 2\,a^2\,x - 4\,a^3\,x^3$$

Agora só falta determinar o valor do parâmetro onde ocorre a bifurcação. Como a bifurcação ocorre entre 3.4 e 3.6, só nos interessa resolver a equação $r_1(a,x) = x$ para valores de a entre esses valores.

```
> fsolve({r1=x,r2=-1},{a,x},{a=3.4..3.6,x=0.3..0.5});#Determinação
  dos pontos fixos de f^2 que têm derivada -1
```

$$\{a = 3.449489743,\ x = 0.4399601690\}$$

De facto, para $a = 3.449489743$ surge uma órbita atractora de período quatro.

Uma vez que é possível determinar analiticamente o valor do parâmetro a para o qual se dá a bifurcação, podemos pedir ao aluno que determine, com o auxílio do procedimento **bifurcplot**, que encontre uma boa visualização desta duplicação do período. Pela análise do diagrama de bifurcação, o aluno pode observar a ocorrência de duas bifurcações: uma para x entre 0.4 e 0.5 e outra para x entre 0.8 e 0.9. Para obtermos o parâmetro que corresponde a essas bifurcações, vamos observar somente uma delas em pormenor, por exemplo a que se encontra entre 0.4 e 0.5. Para isso, devemos fazer um enquadramento apropriado para os eixos horizontal e vertical que nos permita visualizar o parâmetro com algumas casas decimais. Na figura 4.16 exibimos o pedaço do diagrama de bifurcação que se obtém ao escrevermos a instrução:

```
> bifurcplot(f,3.446,3.45,500,700,900,0.1,0.4,0.5);
```

obtemos o gráfico que se exibe na figura 4.16. Pela observação deste gráfico, o aluno pode verificar que a bifurcação ocorre antes de 3.449, o que não está de acordo com os resultados analíticos obtidos! Assim, para evitarmos erros deste género, devemos informar o aluno

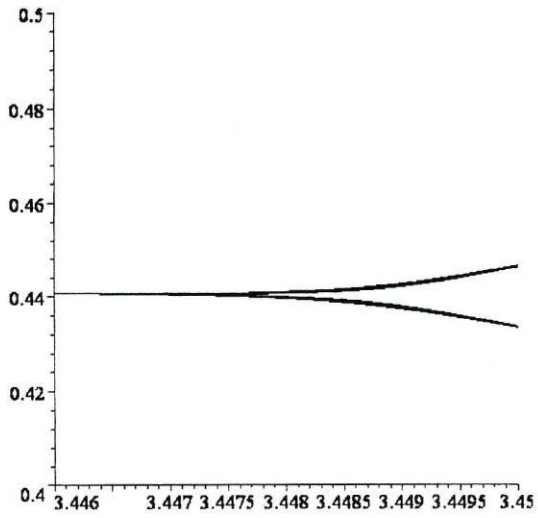


Figura 4.16: Bifurcação em $a = 3.449490$ obtida com o procedimento **bifurcplot**

que deve introduzir um número suficientemente grande de iterações. No caso do exemplo anterior, se escrevermos a instrução

```
> bifurcplot(f,3.446,3.45,500,5000,5200,0.1,0.4,0.5);
```

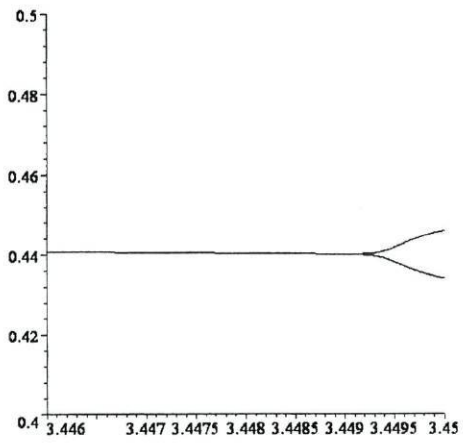


Figura 4.17: Bifurcação em $a = 3.449490$ obtida com o procedimento **bifurcplot**

obtemos o gráfico da figura 4.17 que tem uma resolução bastante melhor e que oferece um resultado mais próximo do obtido através do método analítico.

Pelo diagrama de bifurcação tudo leva a crer que pouco depois de 3.5 esta órbita

deixa de ser atractora. Será que a seguir aparece uma órbita atractora periódica de período oito? De modo idêntico ao que fizemos para a órbita periódica de período quatro, podemos pedir ao aluno que recorra ao procedimento **pernplot** para obter alguns gráficos, para diferentes valores de a a fim de averiguar se existem pontos periódicos de período oito para $a > 3.449489743$. Na tabela 4.13 exibimos alguns dos gráficos que podem ser obtidos pelos alunos, para valores de a entre 3.5 e 3.6.

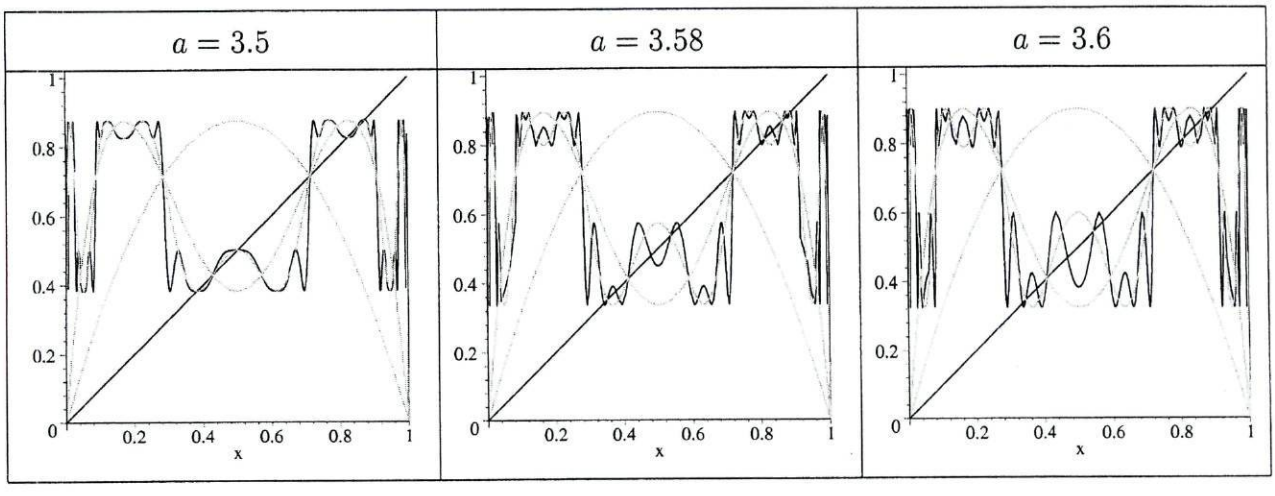


Tabela 4.13: Pontos periódicos de período oito de f_a com a entre 3.5 e 3.6

De facto, nos gráficos obtidos é clara a existência de pontos periódicos de período oito entre 0.3 e 0.5.

Para determinar analiticamente o valor de a a partir do qual se dá a bifurcação em pontos periódicos de período oito, escrevemos as seguintes instruções:

```
> f:=x->a*x*(1-x):

> r1:=expand((f@@4)(x)):#Expressão de f^4

> r2:=diff(r1,x):#Derivada de f^4
> fsolve({r1=x,r2=-1},{a,x},{a=3.5..3.6,x=0.3..0.5});#Determinação
  dos pontos fixos de f^4 que têm derivada -1
```

O resultado obtido é:

$$\{x = 0.3632903374,\ a = 3.544090360\}$$

Então, para $a > 3.544090360$ a órbita periódica de período quatro deixa de ser atractora e

dá-se uma bifurcação em pontos periódicos de período oito. Para obtermos graficamente o valor do parâmetro a onde ocorre esta bifurcação, recorremos ao procedimento **bifurcplot**. Como já referimos atrás, devemos escolher um número de iterações suficientemente grande para determinar o valor do parâmetro com algum rigor. O gráfico da figura 4.18 foi obtido com as seguintes instruções:

```
> bifurcplot(f,3.54,3.55,500,2000,2200,0.1,0.3,0.45);
```

obtemos o gráfico que se exibe na figura 4.18. Neste gráfico é bem visível a bifurcação para a muito perto de 3.544.

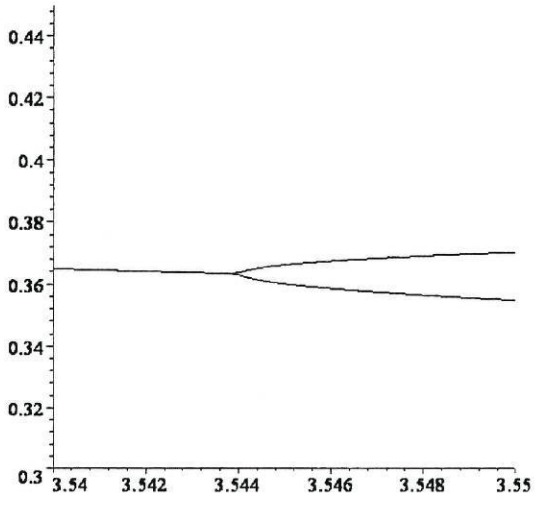


Figura 4.18: Bifurcação em $a = 3.544090$ obtida com o procedimento **bifurcplot**

O professor pode pedir ao aluno que tente determinar graficamente o intervalo do parâmetro a onde se dá a duplicação para o período dezasseis. Na tabela 4.14 exibimos alguns exemplos de gráficos que podem ser obtidos com o procedimento **pernplot**.

Embora, com alguma dificuldade, no gráfico que corresponde ao parâmetro $a = 3.57$, verifica-se que existem pontos periódicos de período dezasseis entre 0.4 e 0.6. A determinação analítica do valor do parâmetro a a partir do qual se verifica a duplicação em pontos periódicos de período dezasseis não é possível com o Maple. Ao escrevermos a instrução:

```
> r1:=expand((f@@16)(x)):#Expressão de f^16
```

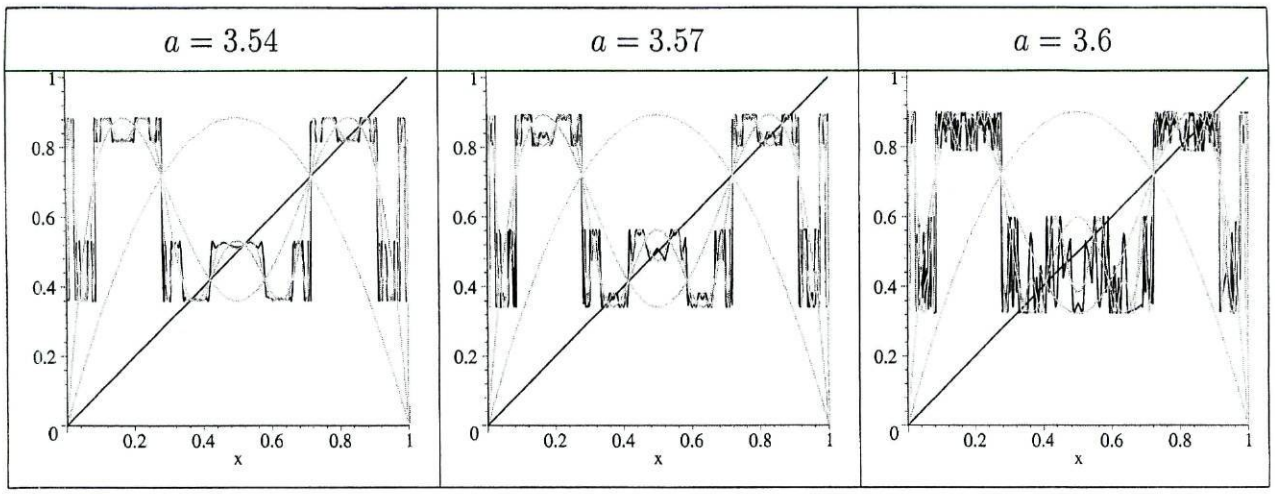


Tabela 4.14: Pontos periódicos de período dezasseis de f_a com a entre 3.54 e 3.6

não conseguimos obter resposta nos primeiros dez minutos! Assim, a determinação do parâmetro a para o qual há bifurcação em pontos periódicos de período dezasseis, por este método, torna-se impraticável para poder ser apresentada a um nível tão elementar. Deste modo, o aluno fica só com a ideia geométrica que de facto existe essa bifurcação.

Ao escrevermos a instrução

```
> bifurcplot(f,3.56,3.57,500,800,1000,0.1,0.3,0.4);
```

obtemos o gráfico que se mostra na figura 4.19. Através deste gráfico, o aluno observa que

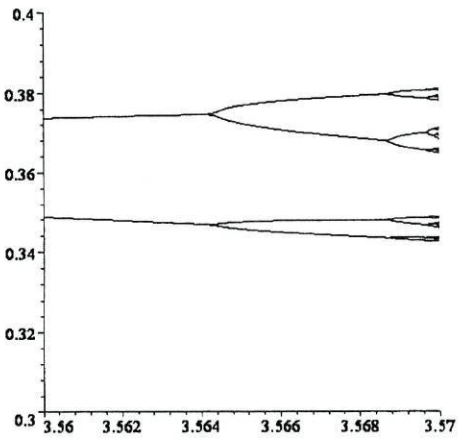


Figura 4.19: Bifurcação em $a = 3.5644073$ obtida com o procedimento **bifurcplot**

entre 0.3 e 0.4 existem duas bifurcações. Para obtermos uma delas (por exemplo, a que está entre 0.37 e 0.38) com maior precisão quer para o eixo vertical quer para o horizontal, podemos escrever a seguinte instrução:

```
> bifurcplot(f,3.562,3.568,500,1000,1200,0.1,0.37,0.38);
```

obtemos o gráfico que se mostra na figura 4.20.

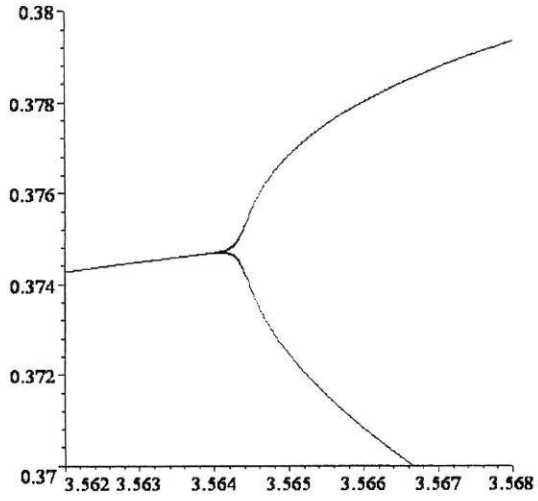


Figura 4.20: Bifurcação em $a = 3.5644073$ obtida com o procedimento **bifurcplot**

Determinar analiticamente o valor do parâmetro a para o qual se dão as próximas bifurcações é inviável, como já referimos para o período dezasseis. Assim, podemos fornecer ao aluno uma tabela como a da tabela 4.15, (que foi extraída de [1] página 504) para que o aluno tente encontrar graficamente (através do procedimento **bifurcplot**) os parâmetros aí referidos.

Período	16	32	64	128	256
Parâmetro a	3.5644073	3.5687594	3.5696916	3.5698913	3.5699340

Tabela 4.15: Valores do parâmetro a de duplicação do período

Apesar de não haver grande manipulação do procedimento, é surpreendente que para determinar graficamente o parâmetro a correspondente ao período 32 tenhamos que exe-

cutar um elevado número de iterados, o que leva a que o tempo de espera pelo resultado seja muito mais elevado que nos casos anteriores. Ao escrevermos a instrução:

```
> bifurcplot(f,3.5686,3.5688,500,5000,5200,0.1,0.365,0.375);
```

o Maple demora cerca de seis minutos a fornecer o gráfico da figura 4.21.

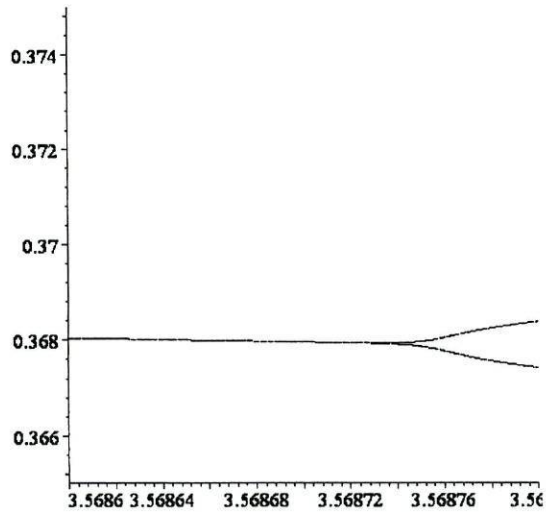


Figura 4.21: Bifurcação em $a = 3.5687594$ obtida com o procedimento **bifurcplot**

Como é fácil de perceber, imagens que demorem tanto tempo a serem produzidas não têm grande interesse para serem apresentadas numa aula. Contudo, é natural que o aluno nutra uma certa curiosidade em saber o que se passa para outros valores de a. O professor deverá incentivar o aluno a experimentar o procedimento **bifurcplot** com outros valores para os parâmetros de entrada a fim de estes conseguirem obter graficamente valores próximos dos da tabela 4.15.

Depois do aluno obter pelo menos os seis primeiros valores dos parâmetros onde ocorrem as bifurcações, o professor pode pedir que seja apresentada uma tabela como a da figura 4.16. Nesta tabela, exibe-se o valor dos parâmetros para os quais há duplicação do período e determina-se o quociente que aproxima a constante de Feigenbaum.

Para terminar este estudo, o professor pode pedir ao aluno que analise esta tabela, e levá-lo a conjecturar que existe um valor para o parâmetro a (aproximadamente 3.570), a que podemos chamar a_∞, a partir do qual existem órbitas de todos os períodos da forma

Período	Parâmetro a	Razão
2	3.000000	
4	3.4494896	
8	3.5440903	4.7514
16	3.5644073	4.6562
32	3.5687594	4.6683
64	3.5696916	4.6686
128	3.5698913	4.6692
256	3.5699340	4.6694

Tabela 4.16: Parâmetros de duplicação do período da logística e quociente que aproxima a constante de Feigenbaum

$2^k, k = 1, 2, \ldots$.

Sabe-se que para $a > a_\infty$, surgem órbitas periódicas de outros períodos. Por exemplo, para $a > 3.678573510$ surgem órbitas periódicas de período ímpar. Recorrendo de novo ao procedimento **pernplot**, é possível exibir aos alunos alguns exemplos dos pontos periódicos de período três entre 3.8 e 3.9. Para isso temos de escrever:

```
> f:=(a,x)->a*x*(1-x):
> g:=x->f(a,x):
> a:=#valor do parâmetro considerado
> pernplot(g,3,0,1);
```

Na tabela 4.17 podemos observar o "nascimento"de uma órbita periódica de período três, para um parâmetro a entre 3.8 e 3.9.

Através desta tabela é possível perceber que para $a > 3.8$ surge uma órbita de período três. De facto, existem duas órbitas de período três: uma atractora e outra repulsora. Vamos ver para que valores de a existem essas órbitas de período três e quando é que uma delas se torna atractora.

```
> f:=x->a*x*(1-x):
```

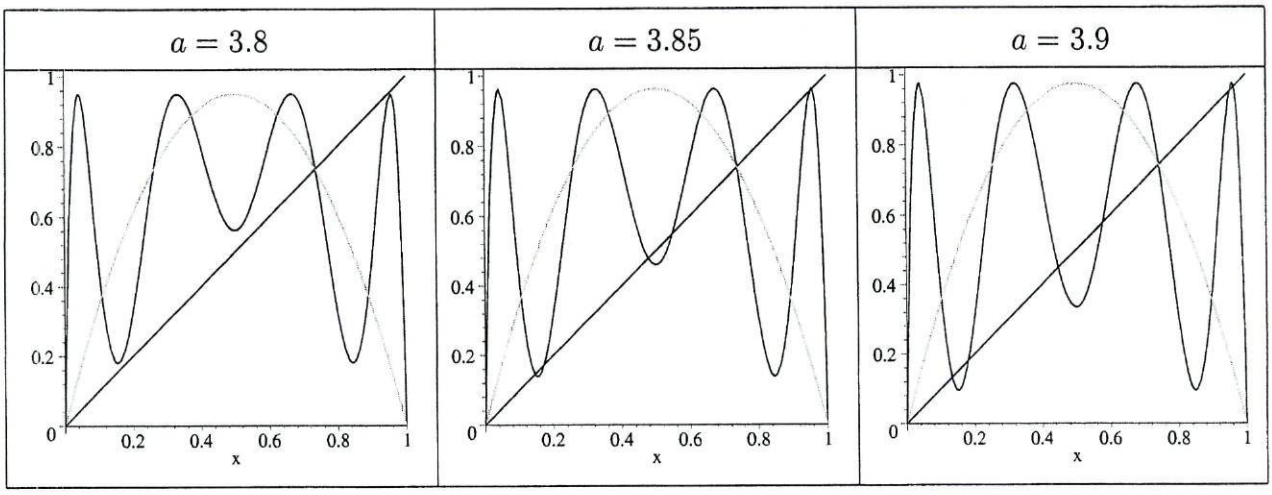


Tabela 4.17: Nascimento de pontos periódicos de período três de f_a com a entre 3.8 e 3.9

```
> r1:=expand((f@@3)(x)):#Expressão de f^3

> r2:=diff(r1,x):#Derivada de f^3
> fsolve({r1=x,r2=1},{a,x},{a=3.8..3.9,x=0.4..0.6});#Determinação dos
  pontos fixos de f^3 que têm derivada 1
```

$$\{a = 3.828427125,\ x = 0.5143552771\}$$

Para $a = 3.828427125$ surge uma órbita atractora periódica de período três.

4.9 Análise estatística de órbitas

Como já referimos no capítulo anterior, é possível analisar uma órbita recorrendo a métodos estatísticos. É usual recorrer-se a estes métodos quando a representação gráfica de uma órbita não nos fornece informação suficiente sobre a mesma.

4.9.1 Diagrama de frequências

Recorrendo ao procedimento diagramafreqs apresentado na secção 2.4 podemos analisar estatisticamente órbitas para diferentes valores do parâmetro da família logística. Este procedimento permite identificar numericamente regiões do intervalo que sejam mais significativas na evolução do sistema, ou seja, que correspondam a passagens mais frequen-

tes da órbita. Para podermos usar este procedimento, temos de inserir alguns parâmetros tais como:

- a função f (que se supõe definida no intervalo $[a, b]$);
- a que é o extremo esquerdo do intervalo;
- b que é o extremo direito do intervalo ;
- N que é o número pontos em que se divide o intervalo $[a, b]$, ;
- x_0 que é o ponto inicial;
- n que é o número de iterados.

Executando o comando

```
> diagramafreqs(f,a,b,N,x0,n);
```

obtém-se um histograma referente ao pedaço de órbita até ao instante n. Mais precisamente, obtemos um gráfico de barras em que a área de cada barra representa a frequência de visitas da órbita ao subintervalo determinado pela base do referido rectângulo.

Com várias experimentações, o aluno poderá ser confrontado com uma grande variedade de situações do género das que se apresentam na tabela 4.18. Os diagramas aí expostos foram obtidos com o procedimento **diagramafreqs**, considerando:

- diferentes valores do parâmetro a entre 0 e 4;
- o intervalo $[0, 1]$ (com 100 subintervalos);
- $x_0 = \frac{1}{3}$;
- 10000 iterações.

Para obtermos cada um deles, devemos escrever as seguintes instruções:

```
> f:=(a,x)->a*x*(1-x):
> g:=x->f(a,x):
```

```
> a:=#substituir pelo parâmetro

> diagramafreqs(g,0,1,100,1/3,10000);
```

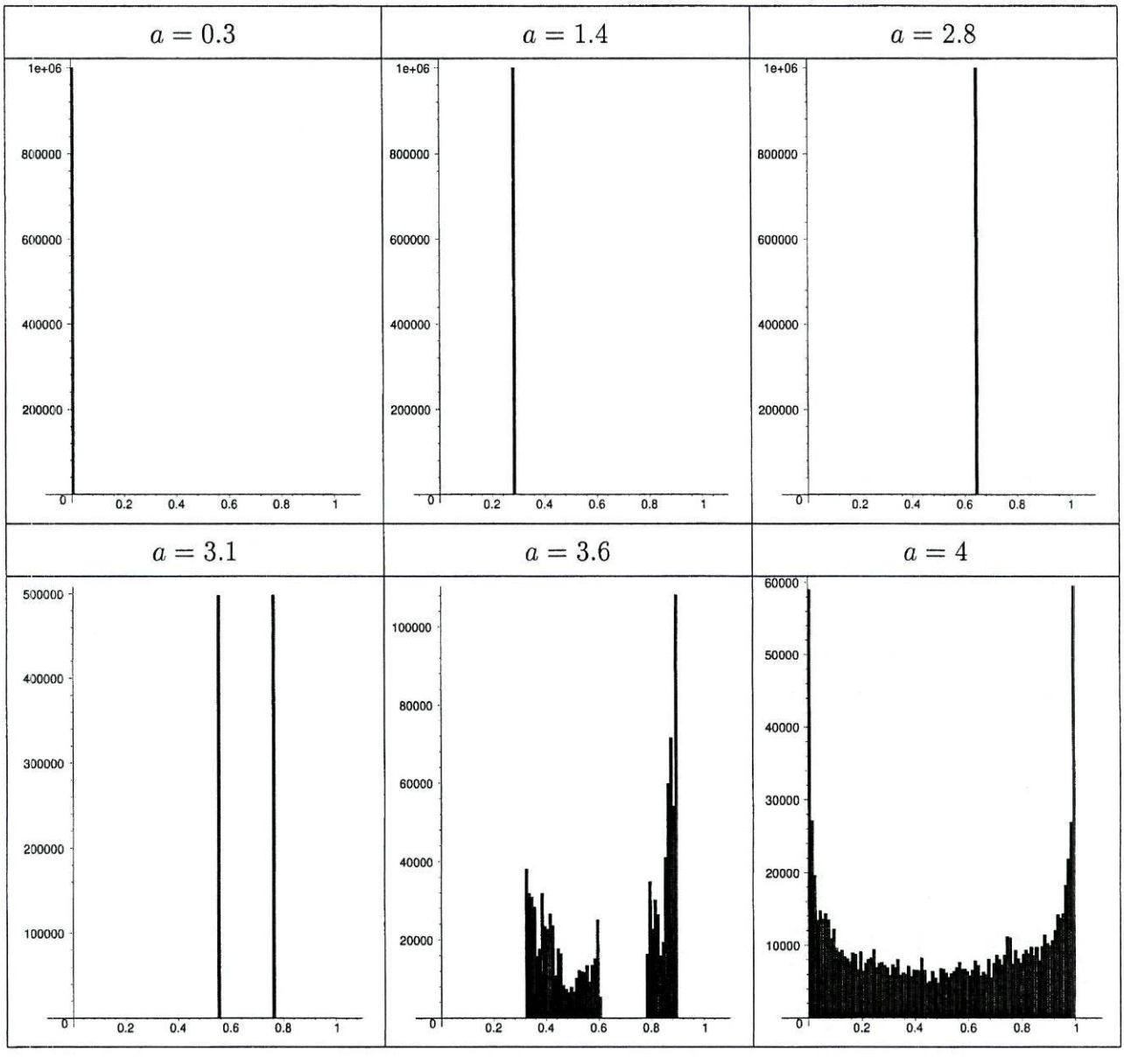


Tabela 4.18: Diagramas de frequências para f_a com $x_0 = \frac{1}{3}$

Com os dados desta tabela, o aluno pode tirar algumas conclusões.

- Para valores de $a < 3$, o diagrama obtido é uma só barra, porque, como já vimos anteriormente, o ω-limite de qualquer ponto da órbita é um ponto fixo;

- Para valores de $3 < a < 3.6$, o diagrama de frequências resume-se a duas barras;
- Para $a = 3.6$ nota-se que a órbita se distribui por alguns dos subintervalos;
- Para $a = 4$, a órbita distribui-se de uma forma uniforme por todos os subintervalos considerados, sendo, no entanto, que o primeiro e o último têm uma maior frequência.

4.9.2 Expoentes de Lyapunov

Pela definição de expoente de Lyapunov dada em 2.4.1, facilmente se pode verificar que para pontos periódicos atractores, eles são negativos e se determinarmos o expoente de Lyapunov nos valores de bifurcação, eles são zero (uma vez que $|f'_a(x)| = 1$ nestes pontos). Para que o aluno tome contacto com estes resultados, devemos pedir-lhes que testem o procedimento **Lyapunov** com diferentes valores do parâmetro a. Se testarem, por exemplo, o procedimento com $a = 0.3$, $a = 1.5$, $a = 3$, $a = 3.5$, $a = 3.5644073$, $a = 3.85$, $a = 3.92$ e $a = 4$, devem previamente definir a função com que estamos a trabalhar.

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):
```

De seguida é necessário definir o ponto inicial, o número de iterações e o parâmetro a a considerar.

```
> x0:=0.01:

> imax:=50000:
> a:=0.3:# este valor deve ser substituído pelo valor do parâmetro a
  considerar
```

Tendo definido estes parâmetros, escrevemos:

```
> Lyapunov(g,imax,x0);
```

e obtemos, neste caso, o resultado para $a = 0.3$.

$$-1.203995148$$

Na tabela 4.19 exibimos os resultados obtidos com os parâmetros atrás indicados

Parâmetro a	Expoente Lyapunov
0.3	-1.203995148
1.5	-0.6929586778
3	-.0002231502008
3.5	-0.8721448502
3.5644073	-.00007432440348
3.85	0.005993952012
3.92	0.5324347656
4	0.6931900696

Tabela 4.19: Expoentes de Lyapunov com $x_0 = 0.01$ e 50000 iterações

Com base nos exemplos desta tabela, ou noutros análogos, o aluno pode verificar que, de facto:

- para valores do parâmetro a onde ocorrem bifurcações ($a = 3$ e $a = 3.5644073$) o expoente de Lyapunov é quase nulo;
- para valores de a inferiores a 3 o expoente de Lyapunov é negativo;
- a partir do parâmetro 3.5, o expoente de Lyapunov é sempre positivo;
- para $a = 4$ o expoente de Lyapunov é $\ln 2$.

Para que o aluno tenha uma visão global da variação dos expoentes de Lyapunov com a variação do parâmetro a, podemos usar o procedimento **lyaplot** descrito em pormenor na secção 3.7.2. Como parâmetros de entrada tem g, *amin*, *amax*, *npontos*, *imax* e x_0, em que:

- g é a função em estudo;
- *amin* é o valor mínimo para o parâmetro a;
- *amax* é o valor máximo para o parâmetro a;

- *npontos* é o número de pontos em que se divide o intervalo $[amin, amax]$;
- *imax* é o número de iterações;
- x_0 é o ponto inicial.

Para visualizarmos o diagrama dos expoentes de Lyapunov para um valor inicial de 0.1 e valores do parâmetro a entre 2 e 4 (com 400 subintervalos), com 500 iterações, devemos escrever:

```
> lyaplot(g,2,4,400,500,0.1);
```

e obtemos o diagrama da figura 4.22.

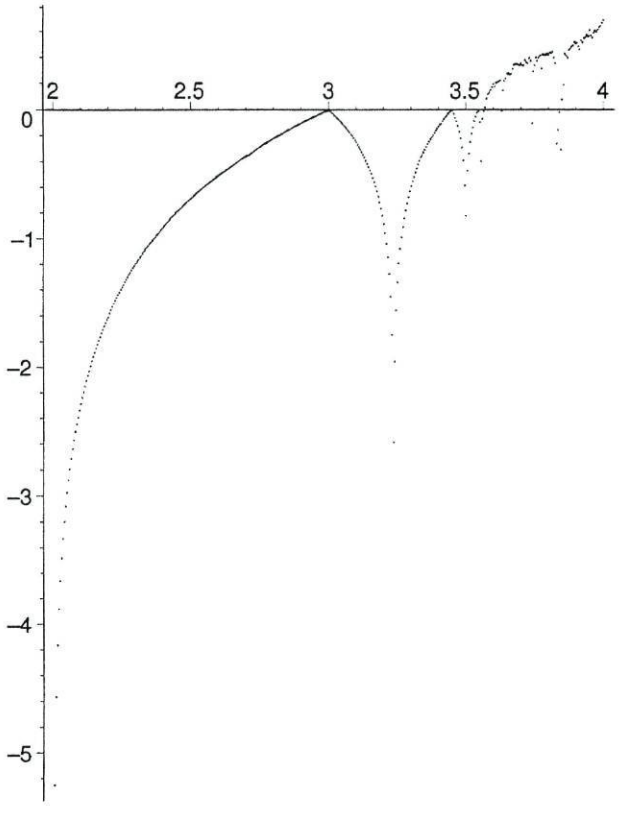


Figura 4.22: Diagrama dos expoentes de Lyapunov

Tal como já referimos para o diagrama da bifurcação, o valor do ponto inicial não interfere com o resultado final obtido. Na aula, podemos pedir ao aluno que teste o

procedimento com vários valores para o parâmetro e x_0 e que conclua que o diagrama obtido é o mesmo em todos os casos.

4.10 Codificação

Nesta secção, vamos começar por ver diferentes itinerários para vários valores do parâmetro a. Para obtermos o itinerário associado a uma órbita, vamos usar o procedimento **itinerario** descrito na secção 2.5. Este procedimento dá-nos uma lista de símbolos, em que cada símbolo está associado a determinado subintervalo de $[a, b]$. Os parâmetros de entrada deste procedimento são a função f, a partição L do intervalo $[a, b]$, o ponto inicial x_0 e o número de iterados pretendido, n. Aqui, na logística, a partição (L) que vamos considerar é a induzida pelo ponto crítico, ou seja, $L = [0, \frac{1}{2}, 1]$. Assim, ao intervalo $[0, \frac{1}{2}]$ corresponde o símbolo 0 e ao intervalo $[\frac{1}{2}, 1]$ corresponde o símbolo 1. Dado um valor inicial x_0, vamos construir o seu itinerário, listando os subintervalos 0 ou 1, conforme contenham ou não x_0 e os seus iterados.

Nos exemplos que se seguem demos como parâmetros de entrada a aplicação logística e a partição $L = [0, \frac{1}{2}, 1]$.

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):

> L:=[0,1/2,1]:
```

Foram consideradas cem iterações para diferentes valores de x_0 e diferentes elementos da família logística.

O professor deve começar por pedir ao aluno que obtenha vários itinerários com este procedimento com valores de $a < 2$ e para diferentes valores iniciais.

- Para experimentarmos, por exemplo, $a = 0.8$ e $x_0 = \frac{4}{5}$, devemos escrever

```
> a:=0.8:

> itinerario(g,L,evalf(4/5),100);
```

o resultado obtido é:

[1, 0, 0, 0

considerado não interfere com o resultado obtido. Na realidade, isto deve-se ao facto que para $a < 2$ as órbitas convergem todas para zero para qualquer que seja o ponto inicial considerado.

De seguida, o professor deve pedir aos alunos que façam experiências com valores de $a > 2$.

- Começamos com o caso em que $a = 2.2$ e $x_0 = \frac{1}{3}$. Como já explicamos, escrevemos

```
> a:=2.2:
> itinerario(g,L,evalf(1/3),100);
```

e obtemos

[0, 0, 1, 1, 1

- Para $a = 3.8$ e para $x_0 = \frac{3}{4}$ escrevemos

```
> a:=3.8:

> itinerario(g,L,evalf(3/4),100);
```

e obtemos

[1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0,
1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0,
1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1]

- Para $a = 4$ e para $x_0 = \frac{1}{5}$ escrevemos

```
> a:=4:

> itinerario(g,L,evalf(1/5),100);
```

e obtemos

[0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0,
0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1,
1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0]

Para valores de $a > 2$, o aluno deverá conjecturar que pode obter uma grande diversidade de itinerários e que nem sempre é fácil, pelos resultados obtidos, perceber se é o intervalo da esquerda ou o da direita que é mais visitado pela órbita. Assim, o professor deve auxiliar o aluno nesta tarefa, propondo-lhe que utilize o procedimento diagramafreq, referenciado na secção 2.5. Ao usar este procedimento o aluno obtém um diagrama de frequências associado a um determinado itinerário. Obtido este diagrama, o aluno facilmente pode tirar conclusões mais fácil e rapidamente acerca da função em estudo.

Para usar este procedimento, o aluno tem de definir a função em estudo, a partição L a usar e o parâmetro a.

```
> f:=(a,x)->a*x*(1-x):

> g:=x->f(a,x):

> L:=[0,1/2,1]:
```

```
> a:=0.8:#Substituir pelo valor pretendido
```

Na tabela 4.20 exibimos os diagramas obtidos com este procedimento com $x_0 = \frac{1}{3}$ e para diferentes valores do parâmetro $a > 2$.

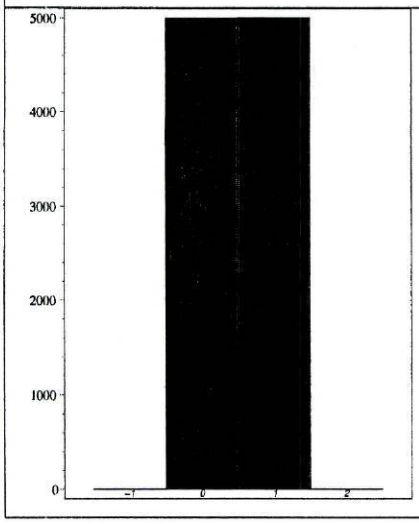
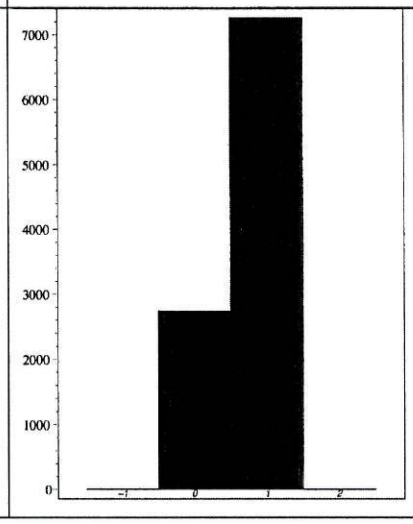
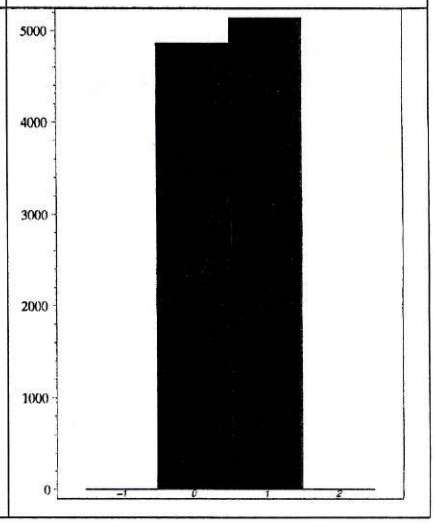
$a = 3.4$	$a = 3.8$	$a = 4$
		

Tabela 4.20: Frequência relativa dos símbolos com $x_0 = \frac{1}{3}$ com 10000 iterações

Com os diagramas obtidos, o aluno facilmente conclui que para $a = 3.4$ e para $a = 4$ o intervalo esquerdo é tão visitado como o intervalo direito e que para $a = 3.8$ o intervalo direito é muito mais visitado do que o esquerdo. O professor deve incentivar o aluno a testar vários exemplos para ver se consegue tirar algumas conclusões.

Capítulo 5

Maplets

Como já foi referido, o Maple é um dos programas de manipulação numérica e gráfica mais conhecido e utilizado. Com a sua versão 8 e introdução da tecnologia das Maplets, este programa passou a poder ser utilizado por alunos e professores sem que estes tenham a necessidade de conhecer a linguagem Maple. O interesse das Maplets é facilitar o uso do Maple, mesmo para quem não sabe o que é o Maple ou nunca ouviu falar deste programa. De facto, uma Maplet permite através de uma janela, onde aparecem botões e caixas de texto (para introduzir os dados a tratar), apresentar o resultado na mesma janela, por um processo invisível para o utilizador.

Neste capítulo vamos apresentar três Maplets: **Pontos fixos de f^n**, **Plotorbita**(Cobweb) e **Bifurcação**. Uma das Maplets que ainda se pode ver neste capítulo, é uma Maplet que permite estudar funções de $\mathbb{R}^2$ em $\mathbb{R}^2$.

5.1 Pontos fixos de f^n

A Maplet que apresentamos na figura 5.1 serve para visualizar os pontos fixos de uma função cuja expressão pode depender de um parâmetro r. É possível ver uma animação dos gráficos de f_r em função do parâmetro r que varia num intervalo pré-definido.

Esta janela é constituída por duas "zonas". A da esquerda onde se exibe a animação e a da direita onde se encontram caixas de texto e botões que permitem controlar a

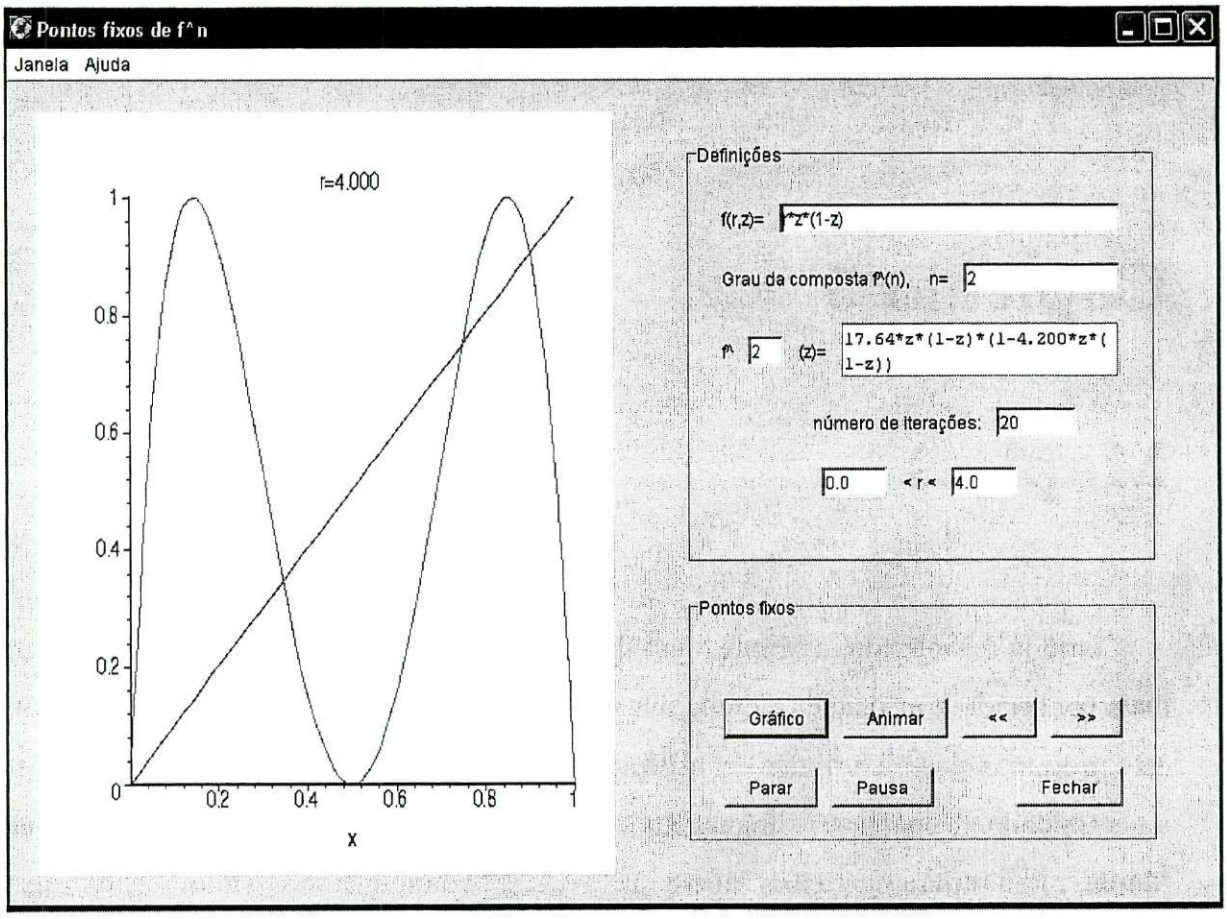


Figura 5.1: Maplet dos pontos fixos de f^n

animação. Nesta última zona, temos uma parte que diz respeito às definições a usar na animação. Nas caixas de texto que a compõem podemos inserir a função que queremos estudar, indicar o grau da composta, o número de iterações e indicar os valores entre os quais pretendemos que o parâmetro r varie. Depois de preencher estas caixas, ao clicarmos no botão 'Gráfico' e em seguida no botão 'Animar', obtemos a animação pretendida. Os pontos fixos da função podem ser visualizados quando o gráfico da identidade (cor azul) se intersecta com o gráfico da função (cor vermelha).

Durante a animação é possível parar a mesma, clicando no botão 'Pausa' e ver imagens intermédias para a frente ou para trás, clicando nos respectivos botões. Ao clicar no botão 'Fechar' o Maplet encerra.

Na figura 5.2 exibimos o resultado obtido ao usar esta Maplet para determinar os

pontos fixos de $f_a^4(x)$ com a entre 0 e 4.

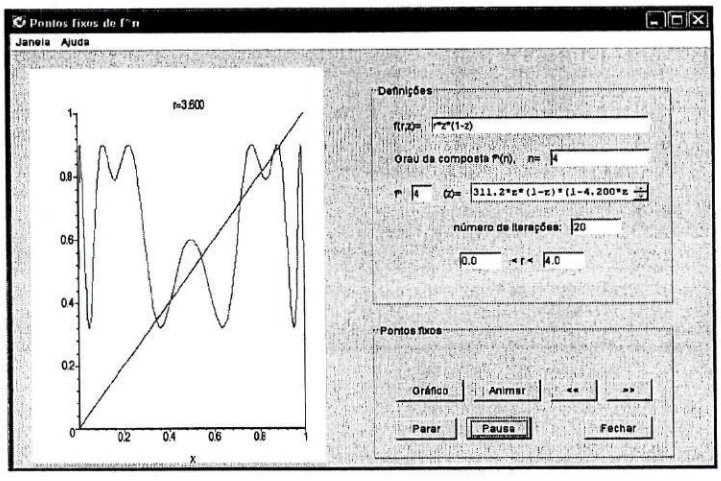


Figura 5.2: Pontos fixos de $f_a^3(x)$ com $a = 3.6$

5.2 Plotorbita

A Maplet que apresentamos na figura 5.3 serve para visualizar a representação gráfica da órbita de um ponto. Depois de definirmos a função, o número de iterações e o ponto inicial clicamos no botão 'Iniciar' para executar a Maplet. Durante a animação podemos clicar no botão 'Parar' e de seguida escolher outra velocidade com que se quer ver os vários gráficos ou então ver a animação passo a passo (ou para a frente ou para trás).

Nas figuras 5.4 e 5.5 que se seguem, exibimos na primeira, a órbita de $T_{1.8}(0.2)$ e na segunda, a órbita de $f_{3.85}(0.23)$.

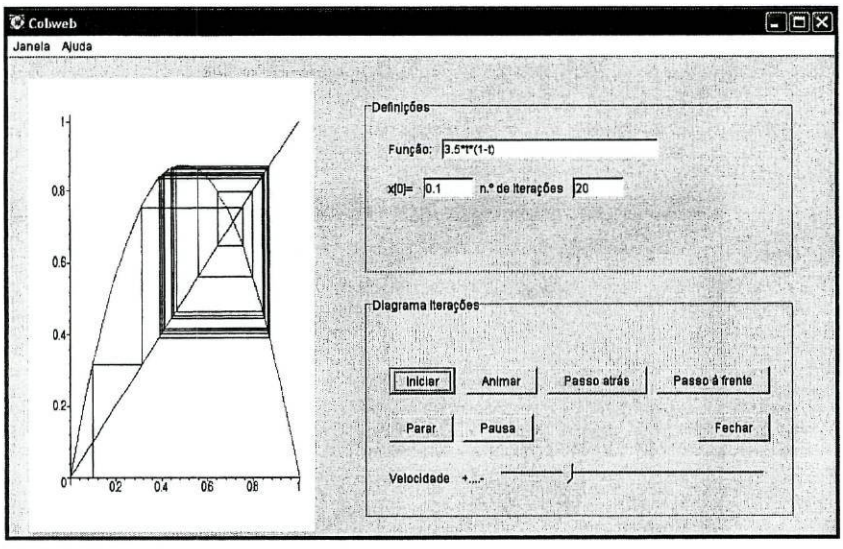


Figura 5.3: Maplet da representação gráfica da órbita de um ponto

5.3 Bifurcação

A Maplet da figura 5.6 serve para visualizar o diagrama de bifurcação da aplicação logística.

É possível, depois de se fazer o devido enquadramento para os eixos horizontal e vertical, visualizar os pontos de bifurcação. Esta Maplet é constituído por um conjunto de três caixas de texto: uma onde se define a função, outra onde se define o ponto inicial e outra onde se indica o número de pontos a considerar para cada valor do parâmetro a. Dadas estas instruções, ao clicar no botão 'Diagrama' obtemos o diagrama de bifurcação. Se não procedermos a qualquer alteração dos valores que são apresentados, por defeito, obtemos o diagrama da figura 5.6. Neste diagrama, o eixo horizontal é visualizado entre 2 e 4 e o eixo vertical entre 0 e 1. São considerados 20 valores diferentes para o parâmetro a entre 2 e 4. Ao clicamos no botão 'Opções' abre-se uma janela como a que se mostra na figura 5.7. Nessa janela aparecem várias caixas de texto com valores que podem ser alterados, tais como: o número de pontos, os iterados entre os quais queremos que os pontos sejam esboçados, o intervalo de variação do parâmetro a e o intervalo de visualização do eixo dos yy. Depois de fazermos alguma alteração nas caixas de texto, devemos clicar no botão

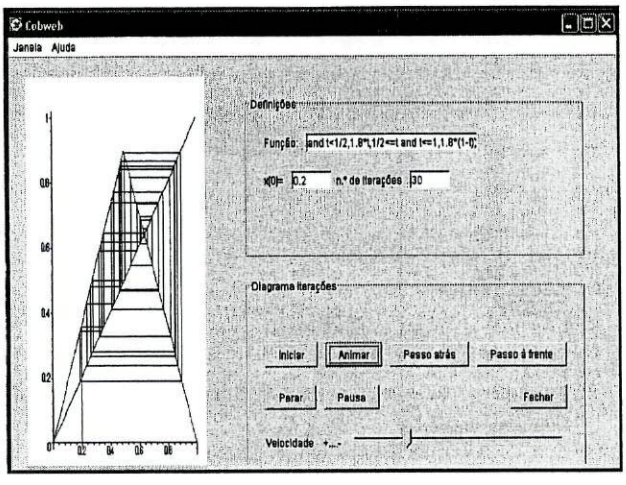


Figura 5.4: Maplet da representação gráfica da órbita de um ponto pela aplicação tenda

'Aplicar' para que as novas definições sejam validadas. Neste menu, existe também o botão 'Duplicar pontos' que ao ser pressionado duplica o número de pontos considerados anteriormente para o parâmetro a. Clicando neste botão obtemos uma resposta em menos tempo do que a anterior, pois o Maple já tem guardado na memória todos os dados necessários à concretização do diagrama e assim não há "perda de tempo"a ir buscar os valores às caixas de texto.

Depois de o aluno estar familiarizado com esta Maplet e estar consciente da poderosa ferramenta que tem ao seu dispor, o professor deve pedir ao aluno que recorra a esta Maplet para determinar os pontos de bifurcação. Deste modo, o aluno será conduzido à verificação da convergência dos valores do parâmetro a onde ocorrem bifurcações para o número de Feigenbaum.

Uma vez que este estudo de determinar graficamente os pontos de bifurcação da logística, já foi feito na secção 4.7, neste capítulo vamos mostrar com um exemplo o que o aluno poderá fazer.

Na secção 4.8 exibimos a tabela 4.19 que indica os parâmetros onde ocorrem as bifurcações e os períodos que lhes correspondem. Para determinarmos graficamente, com esta Maplet a duplicação dos pontos periódicos de período 16 em período 32, vamos começar por usar os seguintes parâmetros:

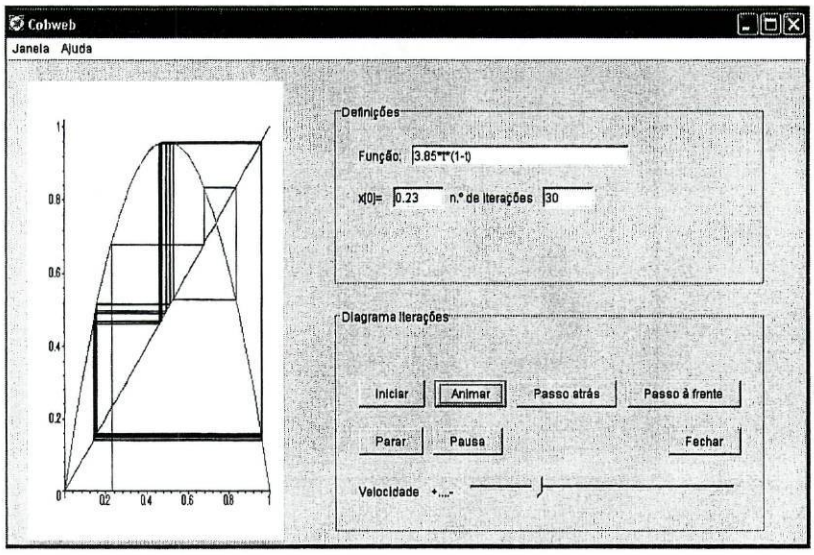


Figura 5.5: Maplet da representação gráfica da órbita de um ponto pela aplicação logística

Número de pontos : 80;

tmin e tmax : 200 e 400, respectivamente;

amin e amax : 3.568 e 3.569, respectivamente;

ymin e ymax : 0.365 e 0.375, respectivamente.

Com estas definições obtemos o resultado que se mostra na figura 5.8. Observando esta figura, o aluno verifica que tem de testar outros parâmetros, para obter a bifurcação com melhor qualidade. Para além de não estar bem definida, no resultado apresentado a bifurcação ocorre antes (perto de 3.5684) do valor que já é conhecido (3.5687594). Para evitar estes erros vamos ter de aumentar o número de iterados e o número de pontos. Ao testar a Maplet com os parâmetros:

Número de pontos : 80;

tmin e tmax : 2000 e 2200, respectivamente;

amin e amax : 3.568 e 3.569, respectivamente;

ymin e ymax : 0.365 e 0.375, respectivamente.

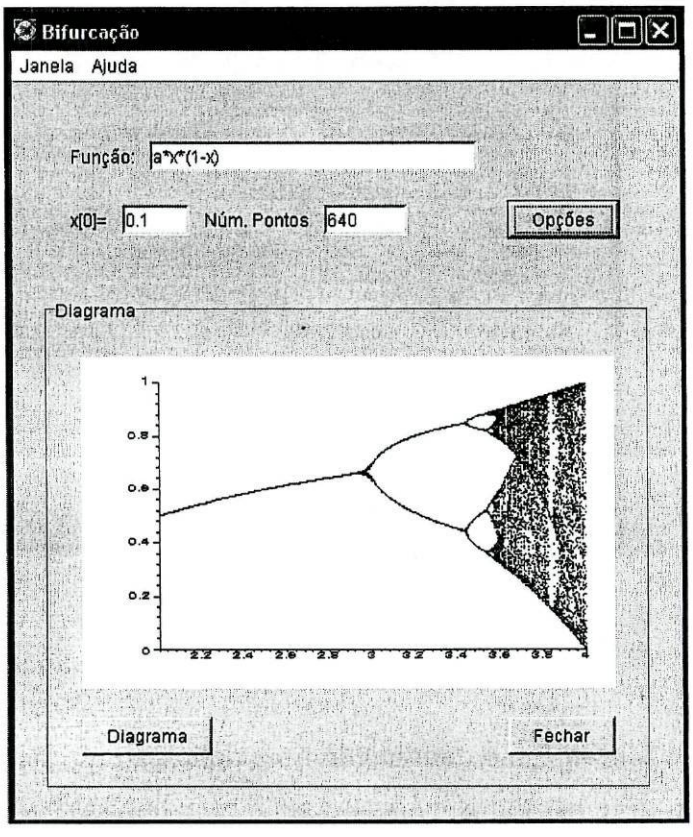


Figura 5.6: Maplet da bifurcação

O professor deve deixar que o aluno experimente outros valores para os parâmetros e que vá fazendo as suas conjecturas. Contudo, deve lembrar ao aluno que se escolher um número de pontos muito elevado e/ou um número de iterações também muito elevado correrá o risco que o seu computador fique com a memória virtual do *windows* bastante baixa e o computador não permita obter os resultados esperados.

5.4 Aplicações bidimensionais

No Maple também é possível estudar, para além das aplicações uni-dimensionais, aplicações de $\mathbb{R}^2$ ou $\mathbb{R}^3$. A Maplet que a seguir exibimos permite visualizar através das cores dispostas no domínio como é que uma função de duas variáveis, caracterizada por um rectângulo, é transformada através de uma aplicação num outro domínio do plano.

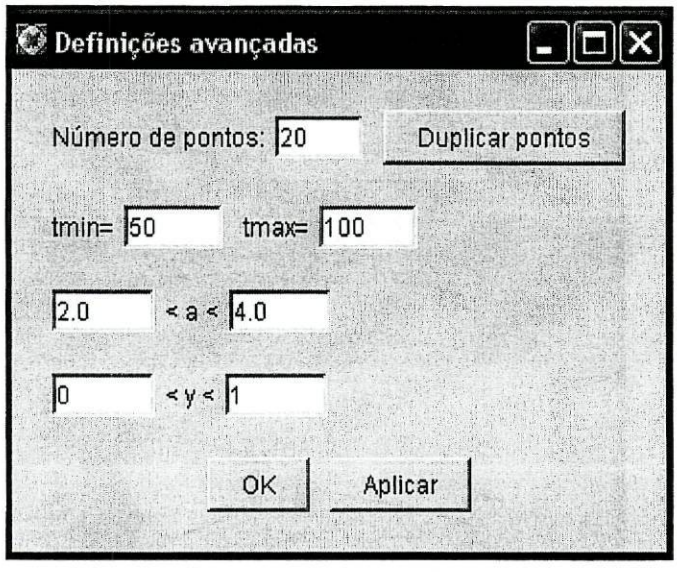


Figura 5.7: Menu Opções da Maplet bifurcação

Nesta Maplet tem-se como domínio de uma aplicação um rectângulo preenchido com várias cores dispostas em linhas horizontais ou em colunas verticais. O utilizador deve inserir os dados relativamente ao domínio, escolher a orientação das cores: horizontal ou vertical, especificar qual a definição de cores que pretende e inserir a função de $\mathbb{R}^2$ em $\mathbb{R}^2$ que vai definir a aplicação. Depois deve clicar no botão 'Domínio de cores' para ver o plano do domínio e a seguir clicar no botão 'Aplicação por F' para ver a região resultante pela aplicação da função. O botão 'Fechar' fecha esta janela.

Um exemplo simples de uma aplicação a duas dimensões é a aplicação de Hénon, que tem duas variáveis, x e y, e dois parâmetros, a e b. A aplicação é iterada através das equações:

$$\begin{aligned} x_{n+1} &= 1 + y_n - ax_n^2 \\ y_{n+1} &= bx_n, \end{aligned}$$

com $a > 0$ e $|b| < 1$.

Esta aplicação para valores de a perto de 2 e para valores de b perto de 0 deixa a região $[-1,1] \times [-1,1]$ invariante. Com a Maplet obtém-se a imagem que se mostra na figura 5.10.

Também é possível fazer um estudo assimptótico desta aplicação. Através do procedi-

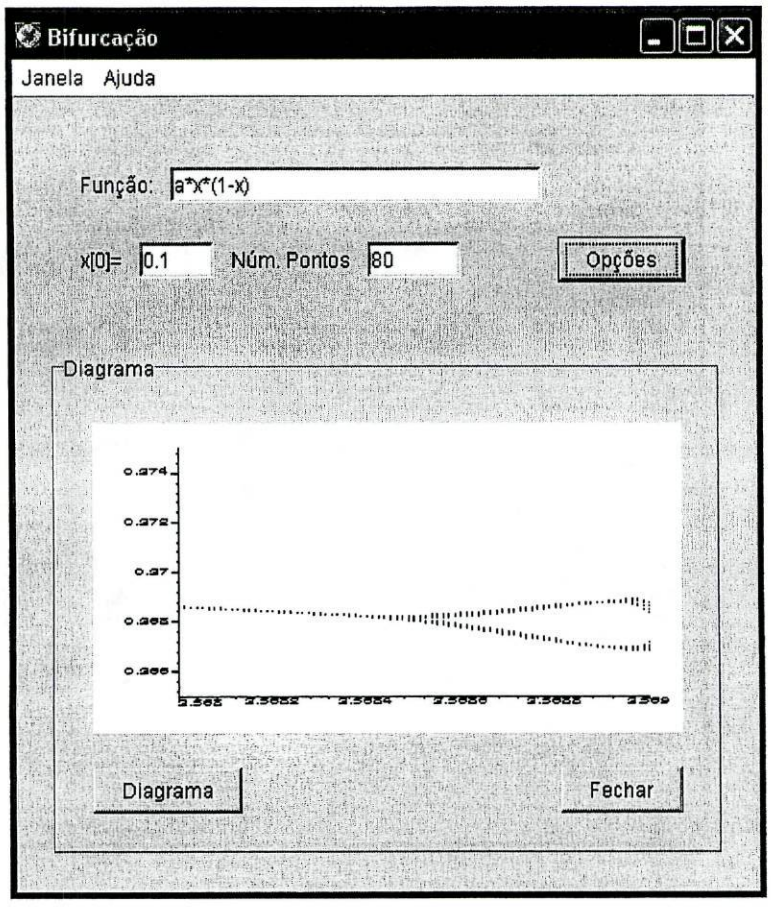


Figura 5.8: Bifurcação em $a = 3.5687594$ obtida com a Maplet **bifurcação**

mento **henon** é possível construir uma Maplet que nos dê a imagem do que é conhecido como atractor de Hénon.

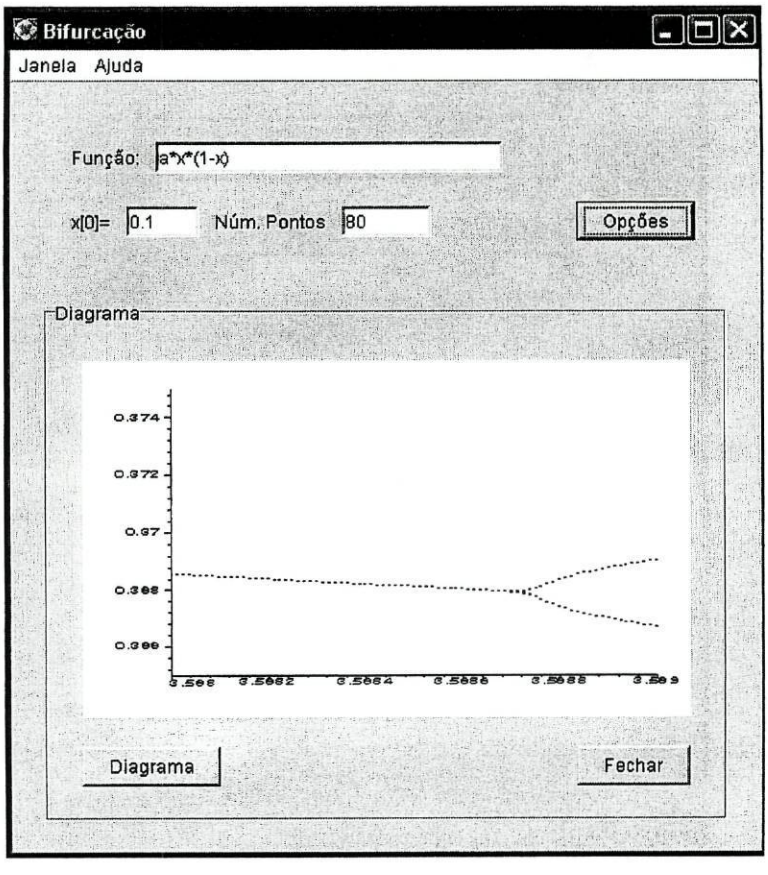


Figura 5.9: Bifurcação em $a = 3.5687594$ obtida com a Maplet **bifurcação**

```
> henon:=proc(a,b,n0,n1,x0,y0)
  local t,i,y,n,pontos,x;
  with(plots):
  pontos:=NULL:
  x[0]:=x0:
  y[0]:=y0:
  for i from 0 to n1 do
  x[i+1],y[i+1]:=f(x[i],y[i]):
  od:
  for n from n0 to n1 do
  pontos:=pontos,pointplot([[x[n],y[n]]],style=point,symbol=point,color=red);
  end do:
  display(pontos):
  end proc:
```

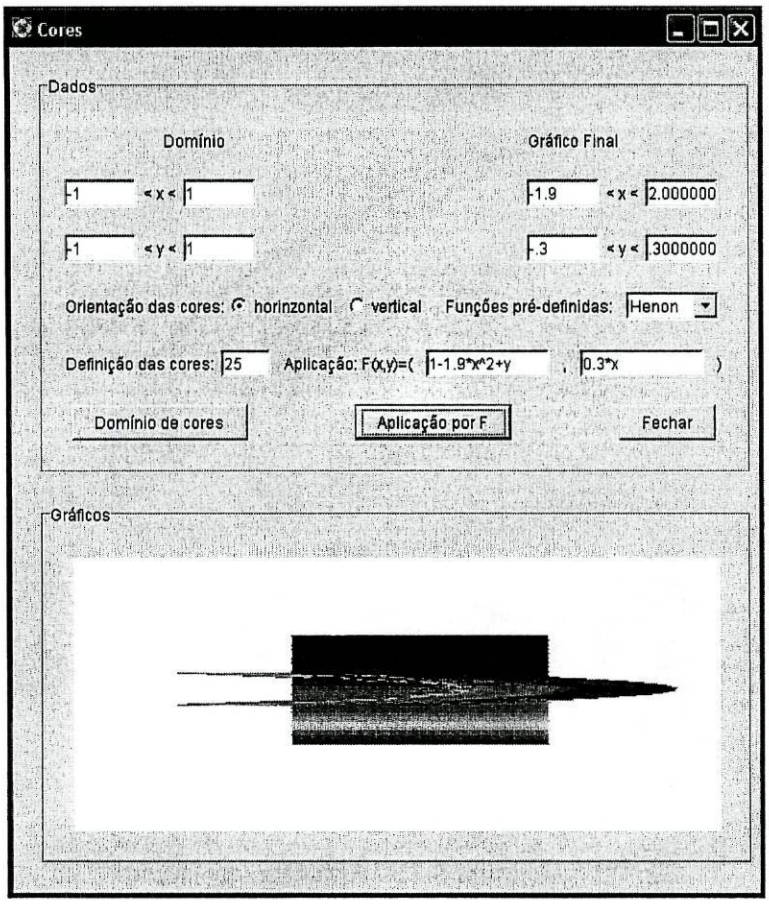


Figura 5.10: Região $[-1,1] \times [-1,1]$ invariante por $F(x,y)$

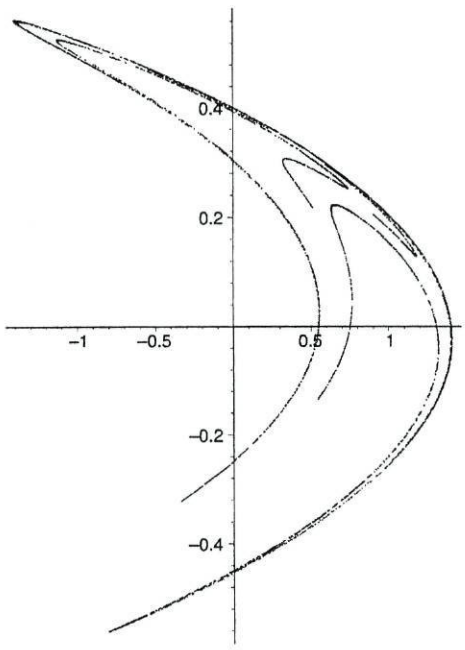


Figura 5.11: Atractor de Hénon

Bibliografia

[1] ALLIGOOD, KATHLEEN T.& SAUER, TIM D.& YORKE, JAMES A.: *Chaos - An Introduction to Dynamical Systems*, Springer-Verlag, 1996.

[2] DEVANEY, ROBERT L.: *An Introduction to Chaotic Dynamical Systems*, Addison-Wesley, 1987.

[3] FEIGENBAUM, MITCHELL. J.: *Universal Behavior in Nonlinear Systems*, Los Alamos Science, 1980.

[4] LI,TIEN; YORKE, JAMES A.: *Period Three implies chaos*, American Mathematical Monthly 82, 982-992, 1975.

[5] LYNCH, STEPHEN: *Dynamical Systems with applications using Maple*, Birkhäuser, 2001.

[6] MAPLE8 - LEARNING GUIDE.

[7] MAPLE8 - GETTING STARTED GUIDE .

[8] MAPLE8 - INTRODUCTORY PROGRAMMING GUIDE .

[9] SARKOVSKII, A. N.: *Coexistence of Cycles of a continuos map of a line into itself*, Ukrainian Math 16, 61-71, 1964.

[10] SOUSA, AVELINO JOSÉ: *Maplets - Modelos Interactivos no Ensino da Matemática*, pré-publicação da dissertação de Mestrado, F.C.U.P., 2003.

[11] WWW.WOLFRAM.COM.

[12] WWW.MAPLESOFT.COM.

[13] WWW.MAPLEAPPS.COM.

[14] WWW.MAPLE4STUDENTS.COM.

[15] WWW.ME.ROCHESTER.EDU/COURSES/ME406/.