



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

MISSÃO DE EXPLORAÇÃO A ASTEROIDE: DINÂMICA, TRAJETÓRIAS E MANOBRAS

**RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA
(PIBIC/CNPq/INPE)**

Gustavo Andrés Diaz (UNESP, Bolsista PIBIC/CNPq)
E-mail: diazlins@gmail.com

Elbert E. N. Macau (LAC/CTE/INPE, Orientador)
E-mail: elbert.macau@inpe.br

Julho de 2017



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

MISSÃO DE EXPLORAÇÃO A ASTEROIDE: DINÂMICA, TRAJETÓRIAS E MANOBRAS

Gustavo Andrés Diaz

Relatório final de Projeto de Iniciação
Científica (PIBIC/CNPq/INPE) orientado
pelo Dr. Elbert E. N. Macau.

INPE
São José dos Campos
2017

SUMÁRIO

1. Introdução	9
2. Objetivos do trabalho	10
3. Fundamentação Teórica	10
3.1. Iterações	10
3.2. Periodicidade de órbitas	10
3.3. Mapa logístico (logistic map) e bifurcações	10
3.4. Atratores	11
3.5. Oscilador Harmônico	11
3.6. Equação Logística	12
3.7. Expoente de Lyapunov	13
4. Métodos utilizados	14
4.1. Iterações com calculadora	14
4.2. Iterações computacionais	15
4.3. Bifurcações do logistic map	15
4.4. Atratores	15
5. Análises e Resultados:	16
5.1. Análise dos dados obtidos do diagrama de bifurcação	16
5.2. Atrator de Hénon	20
6. Conclusões	23
7. Bibliografia	23

LISTA DE ILUSTRAÇÕES

Figura 1 - Função $f(x)=2x$ e $g(x)=x/2$	14
Figura 2 - Diagrama de bifurcação do mapa logístico para $\Delta\lambda=0.05$	15
Figura 3 - Diagrama de bifurcação do mapa logístico para $\Delta\lambda=0.005$	15
Figura 4 - Diagrama de bifurcação do mapa logístico para $\Delta\lambda=0.001$	16
Figura 5 - Diagrama de bifurcação do mapa logístico para $\Delta\lambda=0.0001$	16
Figura 6 - Ampliação do diagrama da Figura 5	17
Figura 8 - Ampliação 3 consecutiva do diagrama da Figura 5	18
Figura 9 - Ampliação do diagrama da Figura 7	18
Figura 10 - Ampliação final do diagrama da Figura 5	19
Figura 11 - Atrator de Hénon	20
Figura 12 - Ampliação no atrator de Hénon	20
Figura 13 - Atratores de Hénon com valores iniciais diferentes	21
Figura 13 - Aumento na Figura 13	21
Figura 13 - Aumento 2 na Figura 13	22

LISTA DE SÍMBOLOS E ABREVIATURAS

a - constante;

b - constante;

λ - letra grega lâmbda, usada como uma variável no caso do diagrama de bifurcação e como definição do expoente de Lyapunov;

$\Delta\lambda$ - acréscimo do parâmetro lâmbda.

$\lim_{t \rightarrow \infty} \lim_{|\Delta x_0| \rightarrow 0} \frac{1}{t} \ln \left| \frac{\Delta x(X_0, t)}{\Delta x_0} \right|$ - limite definido como λ , o expoente de Lyapunov.

RESUMO

Ao estudar missões espaciais, em primeira instância, é preciso compreender como sistemas dinâmicos se comportam e, neste caso, o Modelo dos Três Corpos fornece uma gama de conceitos pertinentes para se estudar e analisar. O modelo possui características dinâmicas complexas, onde surgem regiões com comportamento periódico, quasi-periódico e caótico. Sendo assim, o trabalho iniciou-se com a compreensão básica de tais conceitos da dinâmica caótica, passando por tópicos como órbitas, pontos fixos e periódicos. Assim, quando um espaço de estado que possua tais regiões aonde aparecem de forma emaranhada e formam um cenário de apreciável complexidade surge, o devido entendimento de iterações numéricas e análise do comportamento de funções com variação de parâmetros é importante. Visto que a análise de algumas funções requer a análise de grande número de iterações, houve também uma introdução à programação para poder visualizar e obter conclusões graficamente, uma vez que tal cenário, se adequadamente explorado, pode servir para a realização de manobras orbitais que priorizam pelo uso consideravelmente reduzido de energia. Então a exploração de bifurcações e mapas que possuam atratores acabam sendo um foco para um ensaio preliminar da dinâmica caótica e inspeção de tais regiões para que o trabalho presente, que tem por objetivo explorar dinâmica caótica, identificar suas particularidades que subsequentemente poderão ser exploradas para suportar a realização de missões espaciais com gastos reduzidos de energia.

Palavras-chave: Sistemas dinâmicos. Manobras orbitais. Dinâmica caótica.

1. Introdução:

Missões espaciais, por vezes, precisam de uma otimização na trajetória para a melhora do rendimento e economia de combustível. Assim, é pertinente estudar sistemas dinâmicos, seus conceitos e comportamentos que, neste caso, o Modelo dos Três Corpos fornece. O modelo possui características complexas, com regiões de comportamento periódico, quasi-periódico e caótico.

Para ter-se uma base em relação à sistemas dinâmicos, foi necessário, primeiro, o estudo de alguns conceitos e propriedades, como por exemplo, órbitas, pontos fixos e periódicos; passou-se pelas iterações de algumas funções e seus comportamentos para diferentes parâmetros; adentrou-se mais no conhecimento de bifurcações e seus diagramas, assim, dando outra perspectiva ao comportamento caótico antes não compreendido.

Além da teoria envolvida foram criados programas computacionais. A necessidade de utilizar-se iterações computacionais exigiu o conhecimento e estudo de programação, no caso, a linguagem C. Durante o semestre aprendeu-se novas formas de utilizar, além da linguagem em C, comandos em Linux e a otimização de algoritmos de forma a ilustrar os eventos supracitados e aprimorar o estudo e as habilidades em programação, coleta de dados e plotagem dos mesmos.

2. Objetivos do trabalho:

Trazer a perspectiva do funcionamento de um sistema dinâmico ao aluno implementando rotinas computacionais e, em seguida, introduzir conceitos e comportamentos que explicam a dinâmica caótica e relacionar estes a manobras e trajetórias espaciais.

3. Fundamentação Teórica:

Alguns tópicos são necessários para que o aluno inicie a compreensão de sistemas dinâmicos. Onde os mais importantes estão destacados a seguir.

3.1. Iterações:

Iterações, de forma básica, são repetições da mesma função dado o seu valor final anterior como o próximo valor inicial, ou seja, a repetição de um processo (ou função) várias vezes.

3.2. Periodicidade de órbitas:

Órbitas periódicas ocorrem quando estudam-se funções e estas possuem regiões onde se comportam de forma “constante”, como foi o caso de $\cos(x)$ que possui uma órbita que gira em torno de $0.739085\dots$ para a “n”ésima iteração com $n \rightarrow \infty$; a órbita retornará, eventualmente, para um valor igual ao original ($f^n(x_0) = x_0$) gerando um ciclo.

Esse tipo de órbita possui algumas propriedades como:

- Se x_0 está em uma órbita que possui período n , então $x_0 = f(x_{n-1}) = f^n(x_0)$. O período se repete também para outros termos.
- Órbitas eventualmente fixas ocorrerão quando pontos não fixos/periódicos, entretanto, futuramente um ponto fixo é fixo/periódico.

3.3. Mapa logístico (logistic map) e bifurcações:

Após compreender algumas propriedades das funções, foi vista o mapa logístico que possui como função $x_{n+1} = \lambda x_n(1 - x_n)$. Para diferentes parâmetros de λ e valores iniciais x_0 percebeu-se que as bifurcações aumentam conforme λ diminui, implicando num período (da órbita) tendendo a infinito e a distâncias entre bifurcações sucessivas aproxima-se de zero. Isso para um dado parâmetro $\lambda < -1.4$.

O diagrama de bifurcação tem como ideia geral usar uma semente que é iterada um número finito de vezes com um valor inicial e parâmetro λ fixo. É feito um acréscimo $\Delta\lambda$ no parâmetro fixo e então a partir do resultado final se torna o valor inicial do novo conjunto de iterações. Assim, haverá vezes onde a função irá gerar mais de um valor para um parâmetro λ igual, gerando as bifurcações.

3.4. Atratores:

Os atratores seriam o estudos do comportamento análogo das funções para outras dimensões. Quando o tempo de iteração tende a infinito, estas possuem suas trajetórias invariantes. Um mapeamento de n-tuplas de números reais, ao ser iterado diversas vezes num espaço de dimensão “n”, irá gerar uma órbita nos pontos $p, f(p), f^2(p), \dots, f^n(p), \dots$ onde p é a semente.

De forma análoga às bifurcações, valores de x e y iniciais são iterados com os parâmetros a e b fixos e a cada novo valor de x e y obtidos a próxima iteração muda, devido à forma do mapa de Hénon.

$$\begin{aligned}x_{n+1} &= 1 - ax_n^2 + y_n \\y_{n+1} &= bx_n\end{aligned}$$

O par ordenado (x,y) acaba gerando pontos que pulam de um lugar para outro, já que um parâmetro depende do outro, porém, devido a a e b , qualquer valor inicial irá convergir para o atrator de forma diferente. Este mesmo par, mostra que estamos trabalhando com dados em dimensão 2. Para outras dimensões maiores, podem ser formados superfícies, volumes e seus análogos para tais dimensões.

Outros exemplos de atratores, agora em 3D, são: Chua (circuito eletrônico), Duffing (oscilador não-linear), Rössler(cinética química) e Lorenz(convecção atmosférica).

Perturbações extremamente pequenas num sistema podem resultar numa consequência tão grande quanto a diferença entre um tornado acontecendo ou não. O informalmente conhecido Efeito Borboleta é mais conhecido como dependência sensível nas condições iniciais, que foi amplamente estudado por Lorenz. É uma característica essencial de um sistema caótico.

3.5. Oscilador Harmônico:

Vimos três tipos de comportamento: fixo, periódico e caótico (ergódicos). Os dois primeiros, aparecem em vários sistemas dinâmicos contínuos descritos por equações diferenciais com soluções conhecidas e exatas. Comparando o oscilador harmônico (sistema periódico contínuo) com a equação logística (sistema periódico discreto), já que a natureza dos dois sistemas é análoga, podemos realizar análises paralelas do oscilador harmônico simples (OHS) e a equação logística. Deste estudo um novo assunto na análise dinâmica surge, um análogo para o oscilador harmônico forçado (OHF): a “equação logística forçada”. Uma ferramenta para quantificar o comportamento dos mapas iterados. O expoente de Lyapunov, como resultado indireto de imagens obtidas.

O oscilador harmônico possui alguns casos importantes de se ressaltar que serão precisos para as analogias: o oscilador harmônico simples, o oscilador harmônico amortecido (OHA) e o oscilador harmônico forçado (OHF).

3.6. Equação Logística:

A equação logística simples, serve para aproximar a evolução de uma população animal ao longo do tempo. Muitas espécies apresentam o período fértil apenas por um período breve do ano e os filhotes nascem numa outra temporada particular de modo que, quando aptos a comer comida sólida a mesma será abundante. Para tal período, o sistema será melhor descrito por uma equação diferencial discreta do que uma equação diferencial contínua.

Como nem todo animal existente irá se reproduzir (uma porção deles serão machos); nem todas as fêmeas apresentarão um período fértil; nem todo nascimento será bem sucedido, e nem toda gravidez irá até o final. Assim, aumento da população será uma fração da presente população. Então, se " A_n " é o número de animais este ano e " A_{n+1} " é o número no ano seguinte, então $A_{n+1} = rA_n$, onde " r " é a taxa de crescimento ou fecundidade, irá aproximar a taxa de reprodução bem sucedida.

Esse modelo produz um crescimento exponencial sem limite. Porém, toda população é ligada pelas limitações físicas do território, então algum limite deve ser colocado para restringir tal crescimento. Isso pode ser modelado multiplicando-se a população por um número que irá limitar como a população se aproxima da superpopulação ou da extinção. Se normalizarmos " A_n " para sua capacidade então o multiplicador $(1 - A_n)$ será suficiente e o resultado da equação logística se torna $A_{n+1} = rA_n(1 - A_n)$.

O comportamento do sistema é determinado pela órbita do valor inicial da semente. Todas as condições iniciais acabam tomando três diferentes formas de comportamento:

1. **Fixo:** A população se aproxima de um valor estável. Pode também ser por aproximação assintótica como um OHA ou assintoticamente dos dois lados como um oscilador superamortecido. Começando numa semente que é um ponto fixo é como começar um OHS na existência de eventuais pontos fixos.
2. **Periódico:** A população se alterna entre dois ou mais valores fixos. Como o caso, de uma aproximação assintótica em uma direção ou a partir de lados opostos de forma alternada. A natureza da periodicidade é mais abundante nas equações logísticas do que nos OHS. Órbitas periódicas podem ser tanto estáveis quanto instáveis, um OHS nunca se fixará em um estado periódico a não ser que seja forçado. A periodicidade é discreta, sem valores intermediários.

- 3. Caótico:** A população irá eventualmente visitar cada vizinho em um subintervalo de $(0, 1)$. “Aninhado” entre os pontos que visita, há um conjunto contável infinito de pontos fixos e pontos periódicos de cada período. Os pontos são amplamente instáveis, as órbitas caóticas exibem uma dependência sensível às condições iniciais; quaisquer dois pontos próximos irão eventualmente divergir em suas órbitas para qualquer separação arbitrária escolhida.

O comportamento da equação logística é mais complexo que um OHS. O tipo de órbita depende da taxa de crescimento. A melhor forma de visualizar este comportamento das órbitas em função da taxa de crescimento é um diagrama de bifurcação. Escolhendo um valor conveniente para a semente, gerando um grande número de iterações, descartando algumas das primeiras e plotando o resto como função do fator de crescimento teremos uma boa visualização para tal comportamento. Para valores dos parâmetros onde a órbita é fixa, o diagrama de bifurcação se reduzirá a uma única linha; para valores periódicos, uma série de linhas; e para valores caóticos, um conjunto de pontos como comentado anteriormente. Assim, relacionamos as iterações à sistemas discretos e, conseqüentemente, sistemas contínuos.

3.7. Expoente de Lyapunov:

O expoente de Lyapunov é uma medida simples que determina como a órbita irá progredir ao longo do tempo, da mesma maneira que os parâmetros do oscilador harmônico acabam determinando como irá se comportar.

Para isso, considerando dois pontos no espaço: X_0 e $X_0 + \Delta X_0$. Cada um gerará uma órbita no espaço usando alguma equação ou sistema de equações.

Essas órbitas podem ser vistas como funções paramétricas de uma variável dependente do tempo. Se usamos uma das órbitas como referência, então a separação entre as duas órbitas será em função do tempo, pois a dependência sensível apenas surge em algumas porções de um sistema (como a equação logística), essa separação é também uma função dependente da posição inicial e tem a forma $\Delta x(X_0, t)$. Num sistema com pontos de atração fixos ou pontos de atração periódicos, $\Delta x(X_0, t)$ diminui assintoticamente com o tempo; se um sistema é instável, como pinos equilibrados em suas pontas, então as órbitas divergem exponencialmente por um tempo mas, eventualmente, se estabelecem. Para pontos caóticos, a função $\Delta x(X_0, t)$ irá se comportar erraticamente. É, portanto, útil estudar a taxa exponencial média de divergência de duas órbitas inicialmente próximas usando a fórmula:

$$\lambda = \lim_{t \rightarrow \infty} \lim_{|\Delta x_0| \rightarrow 0} \frac{1}{t} \ln \left| \frac{\Delta x(X_0, t)}{\Delta x_0} \right|$$

Esse número, chamado de expoente de Lyapunov “ λ ”, é útil para distinguir entre diversos tipos de órbitas. Funciona para sistemas discretos assim como para sistemas contínuos.

- $\lambda < 0$ A órbita é atraída para um ponto fixo estável ou uma órbita periódica estável. Expoentes de Lyapunov negativos são características de sistemas dissipativos ou não-conservativos (como o OHA, por exemplo). Tais sistemas exibem estabilidade assintótica; quanto mais negativo o expoente, maior a estabilidade. Pontos fixos super estáveis e pontos periódicos super estáveis têm um expoente de Lyapunov de $\lambda = -\infty$. Isto é algo parecido ao oscilador harmônico criticamente amortecido onde o sistema converge ao ponto de equilíbrio o mais rápido possível.
- $\lambda = 0$ A órbita é um ponto fixo neutro (ou, eventualmente, um ponto fixo). Um expoente de Lyapunov nulo indica que o sistema está numa espécie de estado estacionário, um sistema físico com este expoente é conservativo. Tais sistemas exibem estabilidade, como o caso de dois OHS com amplitudes diferentes, por causa da frequência ser independente da amplitude, a imagem gerada pela fase dos dois osciladores será um par de círculos concêntricos. As órbitas, nessa situação, irão manter uma separação constante.
- $\lambda > 0$ A órbita é instável e caótica. Pontos que estão próximos, não importa quão próximos, irão divergir para qualquer separação arbitrária. Todos os vizinhos no espaço de fase serão eventualmente visitados; estes pontos são ditos instáveis. Tal comportamento não impede que nenhuma organização padronizada surja, mesmo o sistema sendo determinístico, não há ordem para a órbita seguir.

4. Métodos utilizados:

4.1. Iterações com calculadora:

Para funções simples podemos iterar utilizando uma calculadora de forma a ilustrar o comportamento destas. Por exemplo, com a função $f(x) = x^2$ utilizando valores de $x > 1$ e iterando “n” vezes e fazer $n \rightarrow \infty$ o nosso $f^n(x) \rightarrow \infty$, onde $f^n(x)$ é a função iterada “n” vezes. Igualmente para valores de $x < 1$ e $n \rightarrow \infty$, $f^n(x) \rightarrow 0$.

Outro exemplo é a função $f(x) = \sqrt{x}$ que também possui comportamentos diferentes de acordo com intervalo escolhido para o valor de x. A “n”ésima raiz ($f^n(x)$) em intervalo onde $0 < x < 1$ e $n \rightarrow \infty$ tenderá a 1 ($f^n(x) \rightarrow 1$).

Além dos exemplos apresentados, foram realizados testes com outras funções como $sen(x)$ e $cos(x)$, as quais introduziram o conceito de periodicidade.

4.2. Iterações computacionais:

As iterações computacionais permitem o processamento de valores (dados) mais rápido, precisão maior e a capacidade de plotar tais dados de forma a analisar o comportamento da função.

Tendo em vista as propriedades que os gráficos têm, a análise gráfica traz o conceito de pontos fixos. Estes podem atrair ou repelir órbitas. Ao fixar-se 0 como ponto fixo para as duas funções, podemos ver que nas proximidades do mesmo, o comportamento das duas funções é diferente. Para $g(x)$, 0 atrai as órbitas de todos os pontos; enquanto que para $f(x)$ todas as órbitas não-nulas se comportam de forma diferente, sendo repelidas pelo ponto fixo.

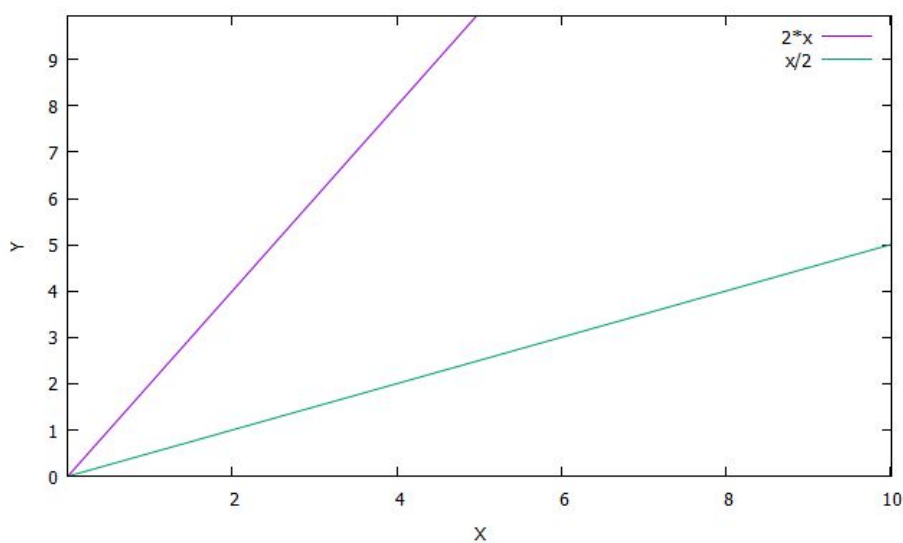


Figura 1 - Função $f(x) = 2x$ e $g(x) = x/2$

4.3. Bifurcações do logistic map:

Para obter o diagrama de bifurcação do mapa logístico programou-se um programa em C e o software gnuplot. De forma que fizeram-se, inicialmente, 500 iterações com o parâmetro $\lambda = 2.0$, $x_0 = 0.5$ e incrementos de $\Delta\lambda = 0.05$, onde λ vai de 2.0 até 4.0 descartando-se as 100 primeiras iterações.

Em seguida, para melhora do diagrama, fizeram-se iterações apenas mudando o valor de incremento de $\Delta\lambda$ sendo estes $\Delta\lambda = 0.005$, $\Delta\lambda = 0.001$ e $\Delta\lambda = 0.0001$. Algumas das rotinas criadas estão em anexo no Anexo I.

4.4. Atratores:

Para tal ilustração foi utilizado o mapa de Hénon com parâmetros a e b fixos sendo $a = 1.4$ e $b = 0.3$ já que para estes valores a semente não escapa e o mapa iterativo não colapsa que possui a forma:

$$x_{n+1} = 1 - ax_n^2 + y_n$$

$$y_{n+1} = bx_n$$

Para obter o atrator, foi feito um programa em C, contido no Anexo I, que fizesse essas iterações 10000000 de vezes usando o valor inicial de $x = 0.5$ e de $y = -0.1$, para em seguida plotar os pares de dados obtidos.

5. Análises e Resultados:

5.1. Análise dos dados obtidos do diagrama de bifurcação:

Após obtenção de dados e plotá-los, obtiveram-se os seguintes gráficos:

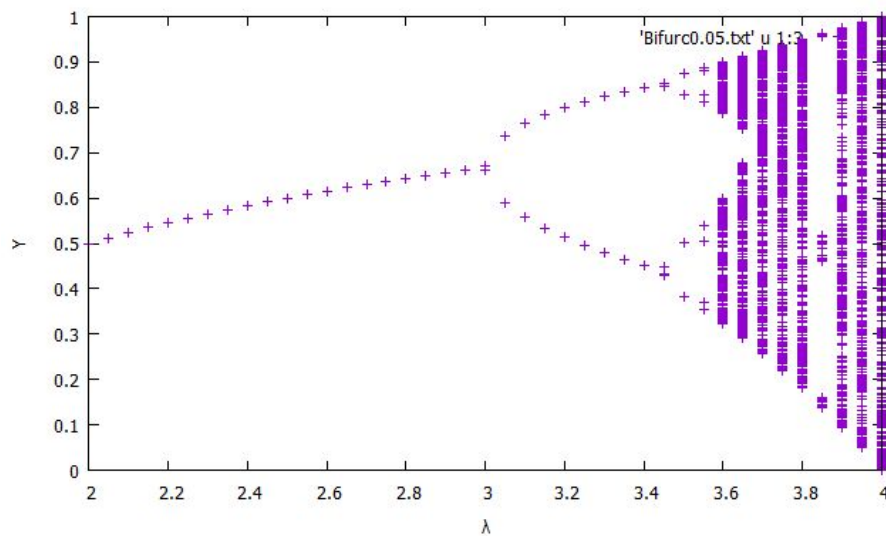


Figura 2 - Diagrama de bifurcação do mapa logístico para $\Delta\lambda = 0.05$.

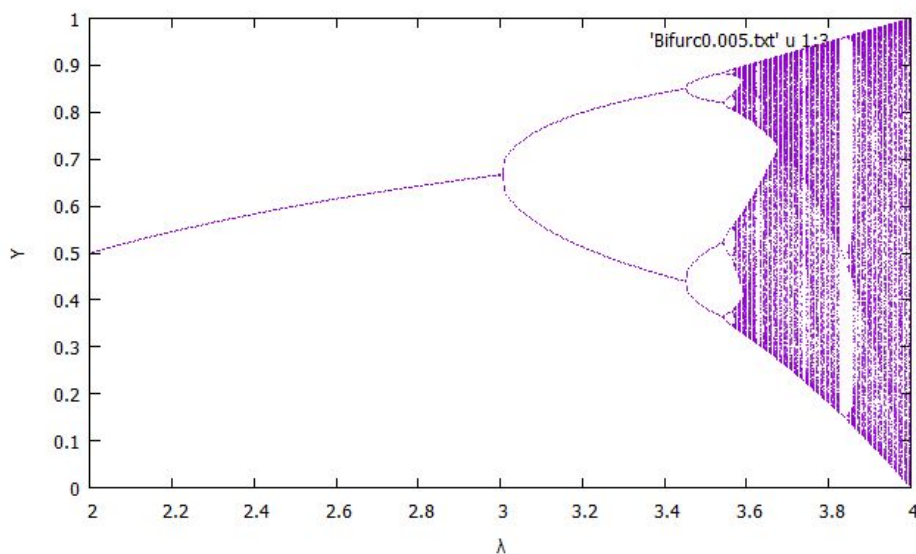


Figura 3 - Diagrama de bifurcação do mapa logístico para $\Delta\lambda = 0.005$.

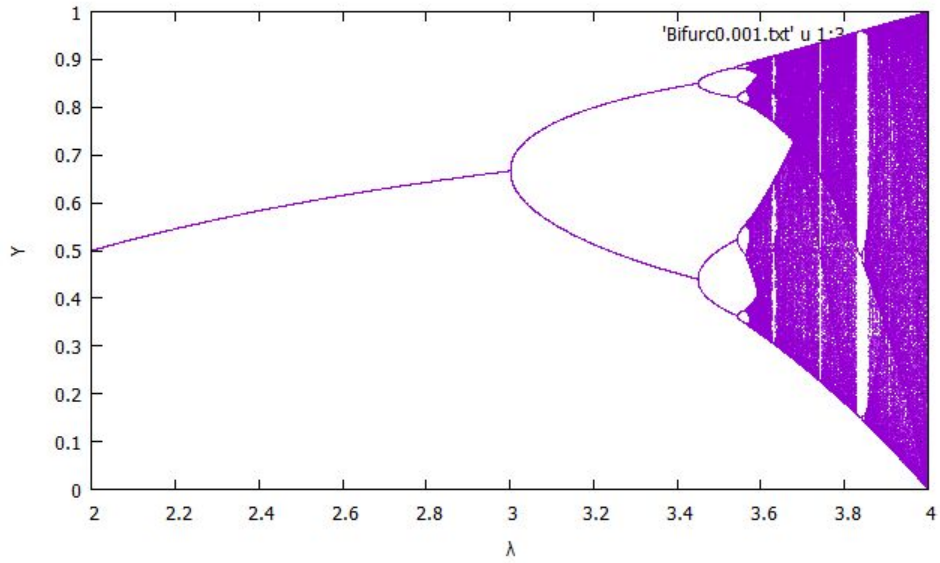


Figura 4 - Diagrama de bifurcação do mapa logístico para $\Delta\lambda = 0.001$.

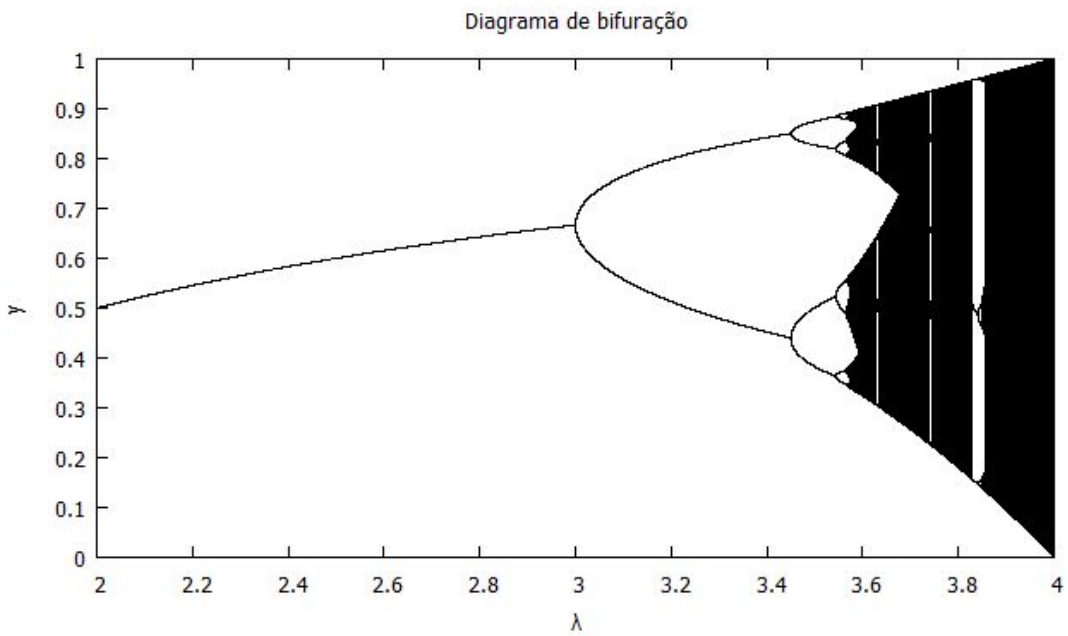


Figura 5 - Diagrama de bifurcação do mapa logístico para $\Delta\lambda = 0.0001$.

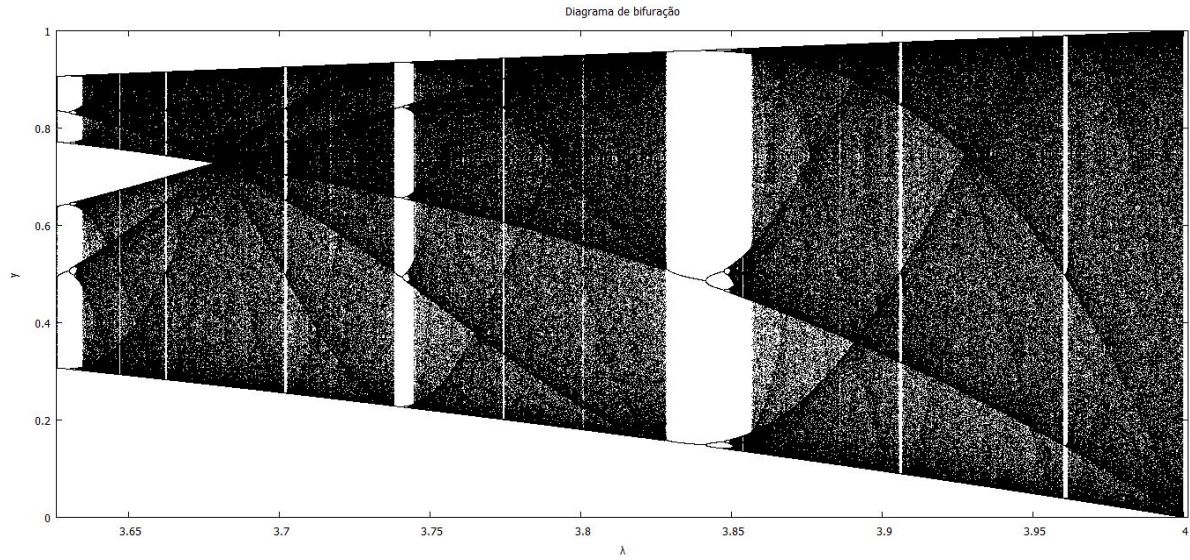


Figura 6 - Ampliação do diagrama da Figura 5.

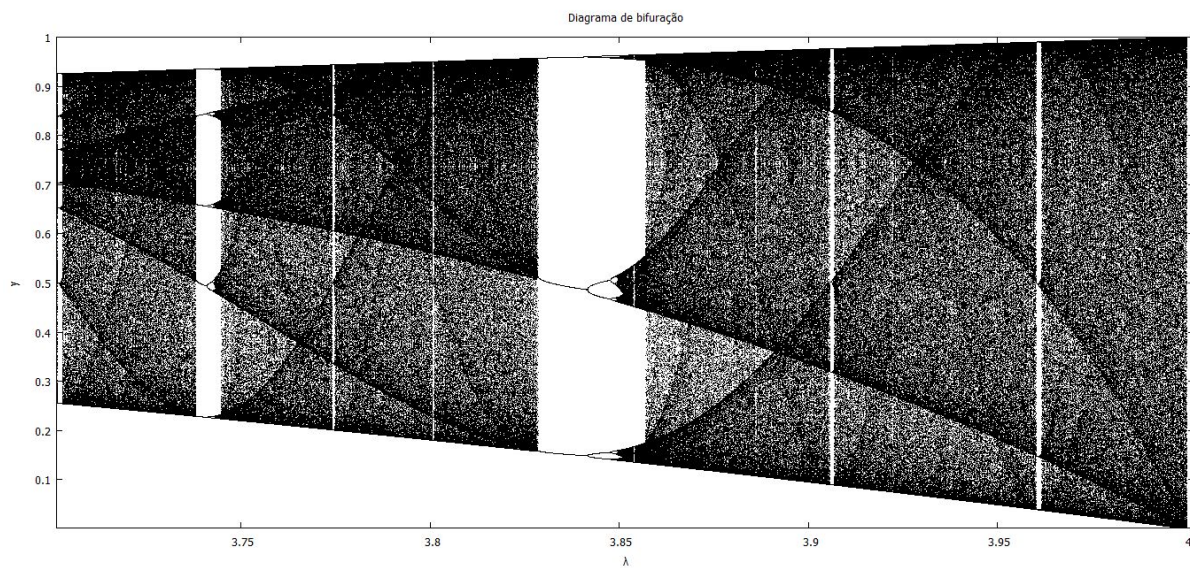


Figura 7 - Ampliação 2 consecutiva do diagrama da Figura 5.

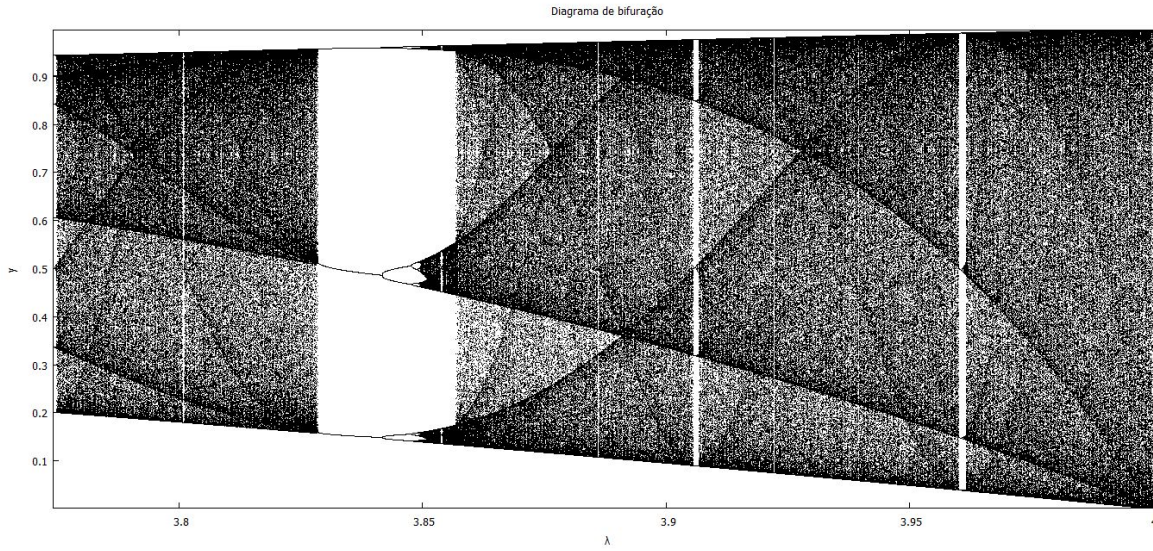


Figura 8 - Ampliação 3 consecutiva do diagrama da Figura 5.

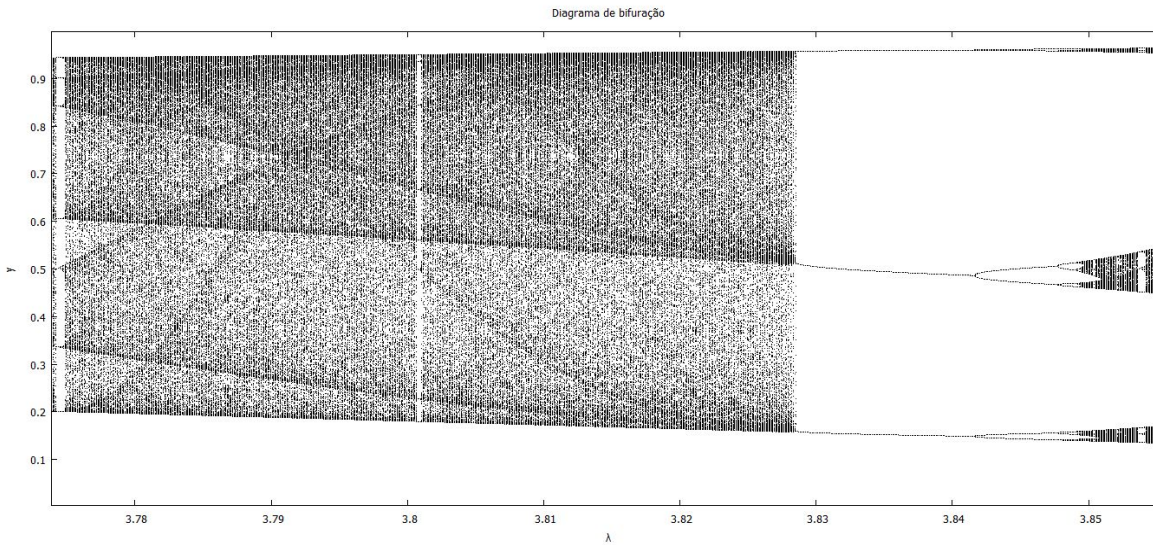


Figura 9 - Ampliação do diagrama da Figura 7.

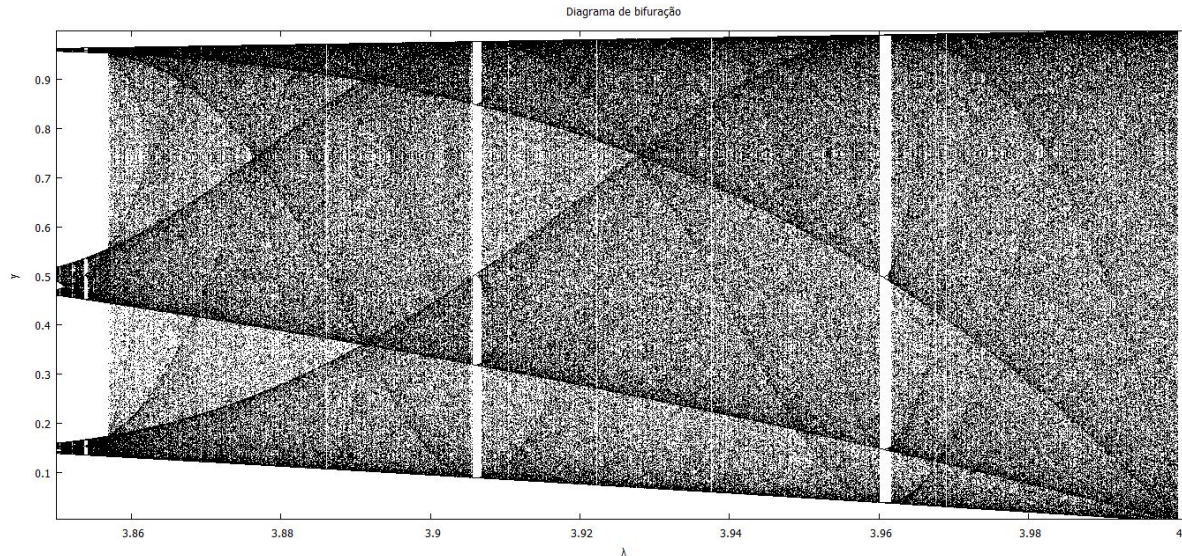


Figura 10 - Ampliação final do diagrama da Figura 5.

Tendo em vista o comportamento observado nas imagens, se confirma o que foi visto nos conceitos teóricos, onde, para valores iguais de λ mais de um valor para a função é obtido, pontos que, a partir deste, começa a bifurcação e repetição do comportamento.

Além disso, após a análise das imagens, foi possível estudar a universalidade desse tipo de diagrama. Há regiões onde o padrão se repete, mesmo para valores de λ diferentes, que seria o intervalo antes comentado que tende a zero conforme o número de iterações tende ao infinito. Este valor é o número de Feigenbaum dado por:

$$\delta = \lim_{n \rightarrow \infty} \frac{\lambda_n - \lambda_{n-1}}{\lambda_{n+1} - \lambda_n}$$

Onde λ_n é o “n”ésimo ponto de bifurcação, λ_{n+1} o ponto de bifurcação que sucede λ_n enquanto λ_{n-1} o ponto de bifurcação que precede λ_n . δ vale 4.6692016091... e aparece repetidamente em figuras semelhantes.

5.2. Atrator de Hénon:

Após os dados serem gerados pelo programa em C e plotados, o resultado obtido é o atrator de Hénon.

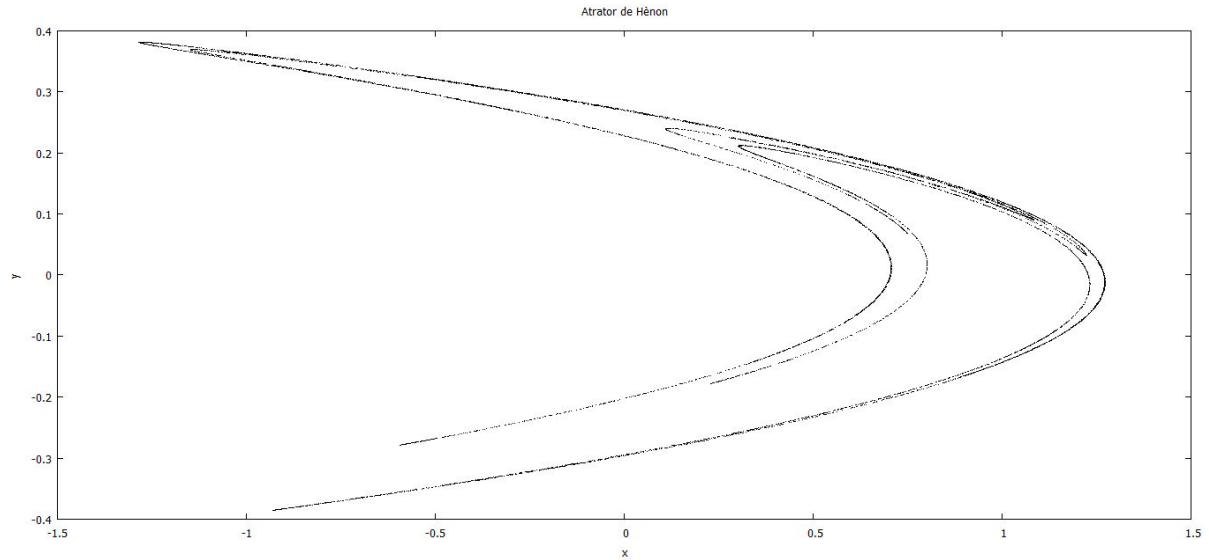


Figura 11 - Atrator de Hénon.

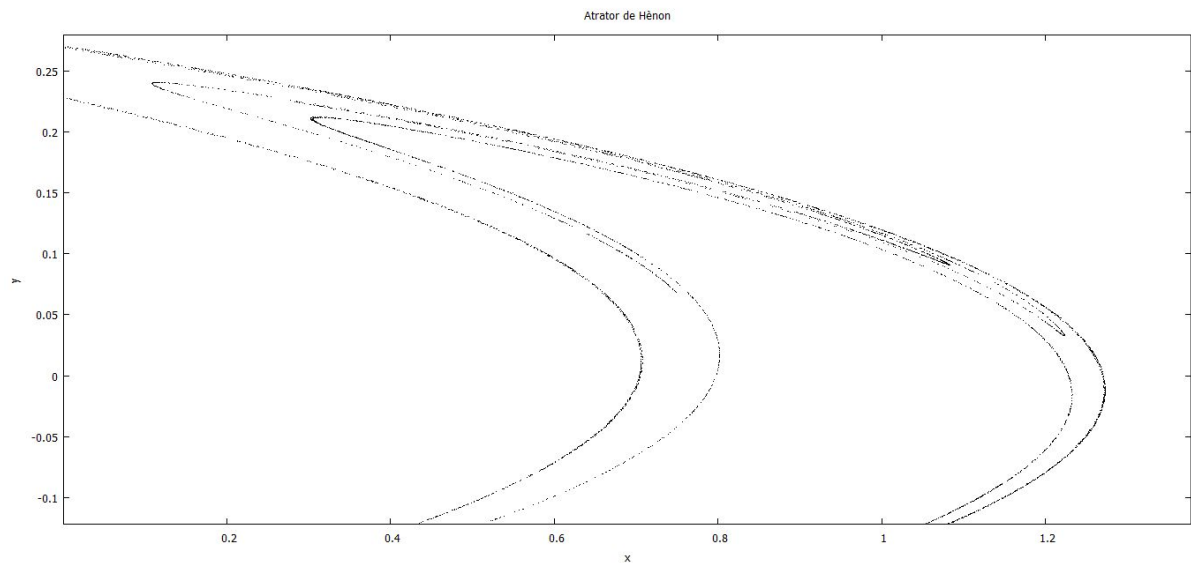


Figura 12 - Ampliação no atrator de Hénon.

O atrator de Hénon também nos mostra uma grande quantidade de “estruturas finas” que podem ser vistas ampliando a imagem.

Ademais, confirmou-se a “trajetória forçada” ao utilizar valores iniciais diferentes obtendo-se o seguinte conjunto de atratores:

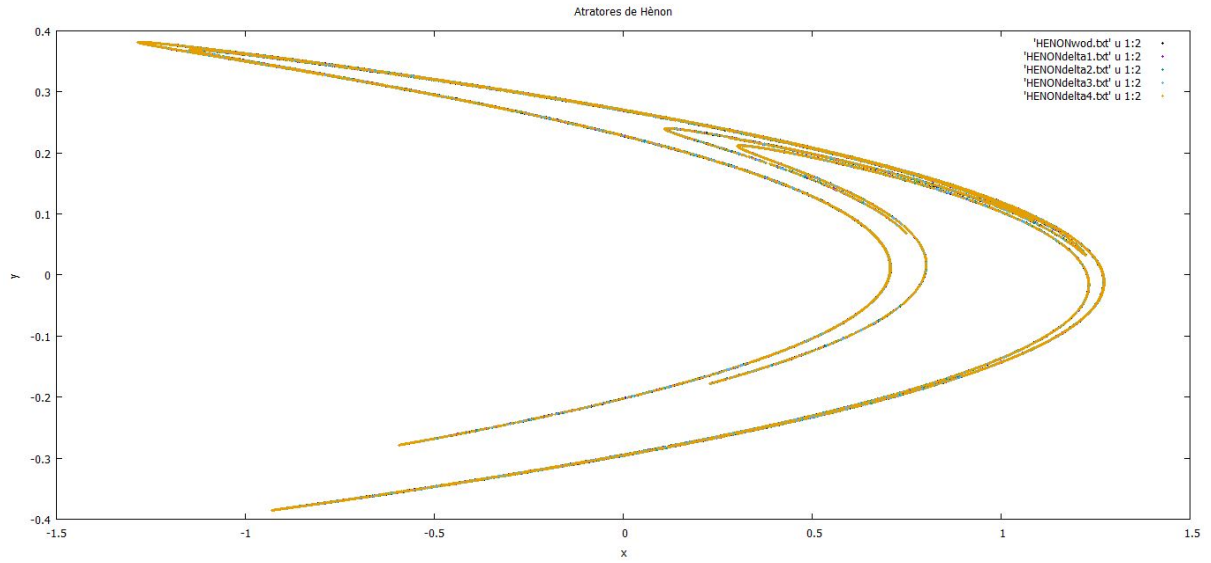


Figura 13 - Atratores de Hénon com valores iniciais diferentes.

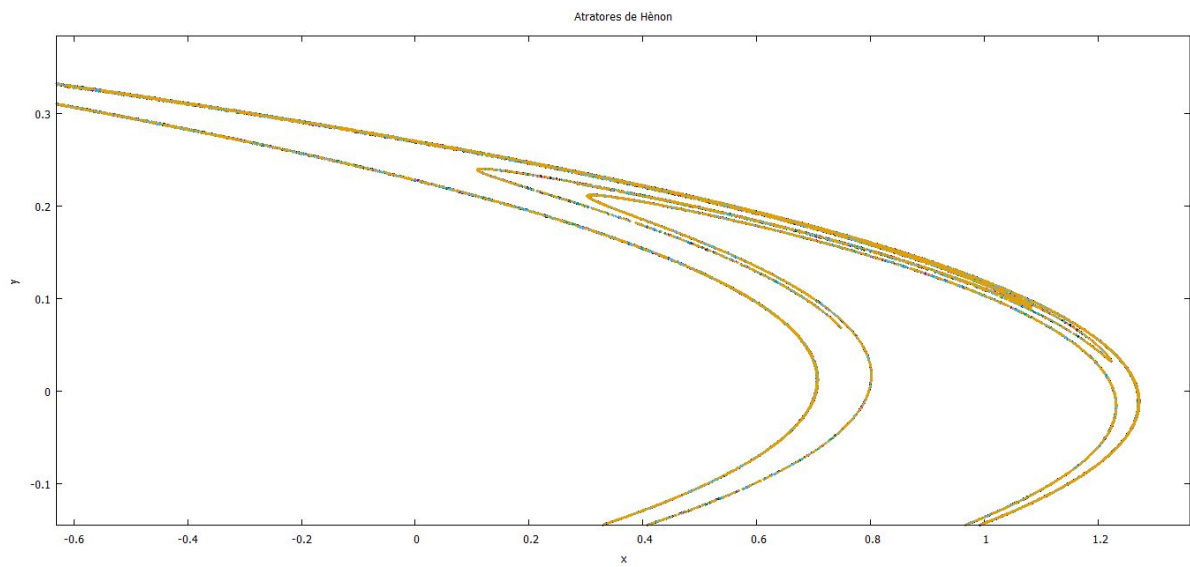


Figura 13 - Aumento na Figura 13.

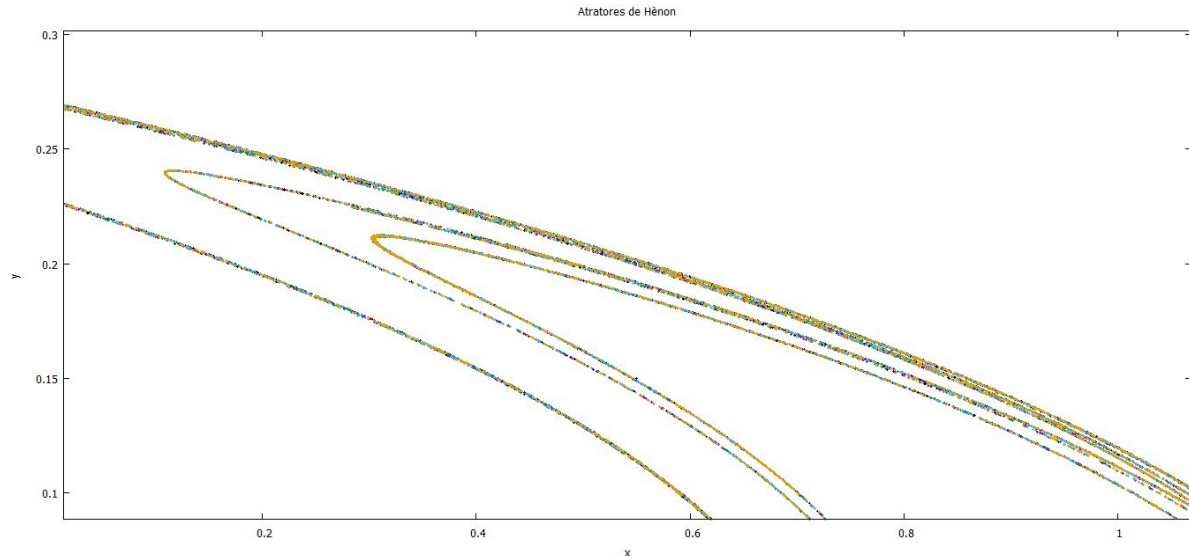


Figura 13 - Aumento 2 na Figura 13.

6. Conclusões:

Após o contato com estas definições, comportamentos e o ambiente computacional, o bolsista pôde ver algumas aplicações tanto no âmbito da programação ao realizar iterações, otimização de rotinas já criadas, plotagem de dados obtidos e sua disposição/apresentação nos gráficos, quanto no ponto de vista físico ao interpretar as imagens obtidas e outras que tiver contato futuramente; sem contar que é possível fazer analogias para outros sistemas físicos a partir das características que alguns sistemas dinâmicos apresentam.

Avançado com os estudos, o bolsista se deparou com o expoente de Lyapunov e buscou-se encontrar a aproximação do valor do mesmo, entretanto faltou a otimização do mesmo.

Sendo assim, conhecer as características e os parâmetros que definem um sistema e relacionar os ao comportamento geral, e acrescentado da análise do expoente de Lyapunov que rege as órbitas possíveis do sistema, se pode classificar esta como dissipativa, conservativa ou instável e caótica.

Portanto a implementação e aprofundamento deste estudo, no caso espacial, de manobras e trajetórias beneficia o rendimento destes tipos de missões, já que seria possível utilizar uma órbita que melhor relacione o tempo e o combustível disponível.

7. Bibliografia:

[1] “*Chaos, Fractals and Dynamics: Computer Experiments in Mathematics*”, Robert L. Devaney, Addison-Wesley, 1990.

[2] “*The Chaos Hypertextbook*”, Glenn Elert. (<http://hypertextbook.com/chaos/> último acesso em 19/07/2017).

ANEXO I

Programas criados:

→ Diagrama de bifurcação:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

int main()

{
    FILE *arq;
    int i, j, m, n;
    int result;
    float c, x0, x;
    char r;

    arq = fopen("Bifurc0.0001.txt", "a+");

    if(arq == NULL)
    {
        printf("Arquivo nao encontrado R.I.P");
        return(0);
    }

    printf("Insira o valor da constante c: ");
    scanf("%f", &c);
    printf("\nInsira o valor inicial x0: ");
    scanf("%f", &x0);
    printf("\nInsira o numero de iteracoes: ");
    scanf("%d", &j);

    fprintf(arq, "Iteracoes para:\tc=%f e x0=%f\n", c, x0);

    for(i=1; i<=j; i++)
    {
        x= c*x0*(1-x0);
        printf("\nx= %f", x);
        if (i>=100)
            fprintf(arq, "%f\t%d\t%f\n", c, i, x);
    }
}
```

```

        x0= x;
    }
    printf("\nDeseja iterar mais vezes? i/n\n");
    while((r= getchar())!= '\0')
    {
        switch (r)
        {
            case 'i': case 'I':      c=c+0.0001;
            printf("c=%f", c);
            fprintf(arq, "Iteracoes para:\tc=%f e x0=%f\n", c, x0);
        }
        while (c<4)
        {
            for(i=1; i<=j; i++)
            {
                x= c*x0*(1-x0);
                printf("\nx= %f", x);
                if (i>=100)
                    fprintf(arq, "%f\t%d\t%f\n", c, i, x);

                x0= x;
            }
            c=c+0.0001;
            printf("\nNovos valores - c:%f e x0:%f", c, x0);
        }

        printf("\nDeseja iterar mais vezes? i/n\n"); break;

        case 'n': case 'N': fclose(arq);

        system("PAUSE");
        return(0);
    }
}
}
}

```

→ Atrator de Hénon:

```

#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>

```

```

int main()

{
FILE *arq;
    int i, j, m, n;
    int result;
    float a, b, x0, x, y0, y;
    char r;

arq = fopen("HENON.txt", "a+");

if(arq == NULL)
{
printf("Arquivo nao encontrado R.I.P");
return(0);
}

printf("Insira o valor da constante a: ");
    scanf("%f", &a);
    printf("\nInsira o valor da constante b: ");
    scanf("%f", &b);
    printf("Insira o valor inicial de x: ");
    scanf("%f", &x0);
    printf("\nInsira o valor inicial de y: ");
    scanf("%f", &y0);
    printf("\nInsira o numero de iteracoes: ");
    scanf("%d", &j);

fprintf(arq, "Iteracoes para:\ta=%f\tb=%f\te x0=%f\ty0=%f\n", a, b,
x0, y0);

    for(i=1; i<=j; i++)
    {
        x = 1-(a*pow(x0,2))+y0;
        y = b*x0;
        printf("\nx= %f\ty=%f", x, y);
if (i>=1000)
fprintf(arq, "%f\t%f\t%d\n", x, y, i);
        x0= x;
        y0= y;
    }
printf("\nDeseja iterar mais vezes? s/n/i\n");
while((r= getchar())!= '\0')
{
switch (r)

```



```

{
case 's': case 'S': printf("Insira o valor da constante a: ");
scanf("%f", &a);
printf("\nInsira o valor da constante b: ");
scanf("%f", &b);
printf("Insira o valor inicial de x: ");
scanf("%f", &x0);
printf("\nInsira o valor inicial de y: ");
scanf("%f", &y0);
printf("\nInsira o numero de iteracoes: ");
scanf("%d", &j);

fprintf(arq, "Iteracoes para:\ta=%f\tb=%f\tx0=%f\ty0=%f\n", a, b,
x0, y0);

for(i=1; i<=j; i++)
{
x= b*y0;
y= 1+x-(a*pow(y0,2));
printf("\nx= %f\ty=%f", x, y);
if (i>=100)
fprintf(arq, "%f\t%f\t%d\n", x, y, i);

x0= x;
y0= y;
}
printf("\nDeseja iterar mais vezes? s/n/i\n"); break;

case 'n': case 'N': fclose(arq);

system("PAUSE");
return(0);
}
}
}

```