



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

CONTROLE DE CAOS E DINAMICA NAO LINEAR

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

Carlos Henrique Dejavite Araujo (ITA, Bolsista PIBIC/CNPq)
E-mail: CARLOSHARAUJO@REDECASD.ITA.BR

Dr. Elbert Einstein Nehrer Macau (LIT/INPE, Orientador)
E-mail: ELBERT@LAC.INPE.BR

Julho de 2006

RESUMO

Este trabalho tem por objetivo o estudo da evolução de um dos sistemas dinâmicos mais simples: o pêndulo amortecido forçado. Primeiramente, são apresentados conceitos básicos para avaliação de sistemas dinâmicos. Estes conceitos são aplicados na dinâmica do pêndulo de modo a avaliar se esta dinâmica se processa para uma evolução periódica ou caótica de acordo com uma terna de parâmetros. Para este fim são desenvolvidos programas de computadores para este específico exemplo. Ao final, obtém-se os intervalos dos parâmetros em que o pêndulo apresenta um comportamento interessante.

SUMÁRIO

	<u>Pág.</u>
CAPÍTULO 1 - INTRODUÇÃO.....	7
1.1- Esboço Geral	<u>9</u>
CAPÍTULO 2 - CONCEITOS BÁSICOS.....	<u>11</u>
2.1 – Espaço de Fase.....	<u>11</u>
2.2 – Atratores	<u>12</u>
2.2 – Bacias de atração	<u>14</u>
2.2 – Secção de Poincaré	<u>14</u>
2.2 –Diagramas de Bifurcação	<u>14</u>
CAPÍTULO 3 - PÊNULO AMORTECIDO FORÇADO.....	<u>16</u>
CAPÍTULO 4 - RESULTADOS	<u>18</u>
CAPÍTULO 5 - CONCLUSÃO.....	<u>24</u>
REFERÊNCIAS BIBLIOGRÁFICAS	24
APÊNDICE A - PROGRAMAS EM AMBIENTE C UTILIZADOS PARA OBTER OS RESULTADOS DESCRITOS	<u>27</u>

CAPÍTULO 1

INTRODUÇÃO

O caos é intitulado àquilo em que o futuro não é previsto com base nos dados presentes, pois o caos não repete seus comportamentos passados. Ele está presente desde de sistemas dinâmicos simples como pendulo amortecido forçado ou vibrações em estruturas mecânicas até reações químicas ou mesmo o clima. Surgiu como nova ciência e vem se desenvolvendo nos últimos anos. Um dos cientistas que contribuiu com seus passos iniciais foi Edward Norton Lorenz com seus estudos sobre atratores estranhos (em forma de borboletas) aplicados ao estudo meteorológico. Uma de seus artigos pode caracterizar o caos: “Previsibilidade: O bater de asas de uma borboleta no Brasil provoca tornados no Texas?”. Onde mostra que pequenas mudanças em condições climáticas geram efeitos imprevisíveis. Comprovando que a previsão climática em longo prazo não é alcançável com a precisão atual de nossas medidas.

Uma das suas principais características é a sensibilidade a condições iniciais. Um sistema dinâmico não caótico apresenta para duas condições iniciais próximas um erro que varia linearmente com o tempo. Diferente do sistema caótico em que o erro varia exponencialmente com o tempo, assim não sendo mensurável o comportamento futuro a partir de mudanças das condições atuais. Este fenômeno foi estudado pelo matemático e astrônomo teórico Henri Poincaré (1854-1912).

Pode parecer que a sua irregularidade é devido às imprevisíveis influências externas, mas é devido ao sistema dinâmico em si. Pois o sistema dinâmico é regido por equações diferenciais e é devido às interações destas que todos os sistemas dinâmicos podem apresentar comportamento caótico. Isto vai depender apenas de duas condições: o sistema de equações dependa de mais de duas variáveis e apresente termos não lineares

que englobem grande parte das variáveis. Estas condições não garantem o caos, mas sua possível existência.

Isto é comprovado, pois um espaço de três dimensões é suficiente para garantir a divergência de trajetórias, confinamento das trajetórias em uma região finita do diagrama de fases e trajetória única. E os termos não lineares garantem que a solução periódica (a série de Fourier que substituí a solução da equação diferencial após o período de transiente) seja instável para a certos parâmetros.

Estes termos não lineares muitas vezes impedem de se obter uma solução analítica. Assim o único método de estudar a evolução de sistemas regidos por estas equações é desenvolver métodos de resolução numéricos. Que se resumem ao cálculo recursivo das variáveis, que definem o sistema dinâmico, a partir das equações diferenciais com uma variação muito pequena de tempo.

1.1 Esboço Geral

Este trabalho foi dividido em mais quatro capítulos, resumidos abaixo:

- **CAPÍTULO 2 – CONCEITOS BÁSICOS:** Neste capítulo são apresentados os conceitos importantes para análise de sistemas dinâmicos. Entre eles estão espaço de fase, Secção de Poincaré, Diagrama de bifurcação, Atratores,...
- **CAPÍTULO 3 – PÊNDULO AMORTECIDO FORÇADO:** Neste capítulo são apresentadas todas as características do pêndulo amortecido forçado (objeto de estudo). Além de derivar as equações diferenciais que governam o movimento do pêndulo amortecido forçado.

- **CAPÍTULO 4 – RESULTADOS:** Neste capítulo são apresentados os resultados obtidos do estudo para o pêndulo amortecido forçado. Mostrando o conjunto de valores para os quais o pêndulo amortecido forçado tem movimento periódico ou caótico.
- **CAPÍTULO 5 – CONCLUSÃO:** Conclusões e perspectiva de trabalho futuro com os métodos aprendidos.

CAPÍTULO 2

CONCEITOS BÁSICOS

1.1 Espaço de Fase

Um sistema dinâmico é regido por equações determinísticas provenientes dos fundamentos da física clássica. Estas equações são equações diferenciais ordinárias (EDO) e podem ser organizadas de maneira a formar um sistema de equações diferenciais de primeira ordem como exposto abaixo:

$$\left\{ \begin{array}{l} \frac{dx_1}{dt} = F_1(x_1, x_2, \dots, x_n) \\ \frac{dx_2}{dt} = F_2(x_1, x_2, \dots, x_n) \\ \dots \\ \frac{dx_n}{dt} = F_n(x_1, x_2, \dots, x_n) \end{array} \right. , \text{ onde } n \geq 3 \text{ e } F_n \text{ apresenta termos não-lineares de maneira a garantir a possibilidade de caos como explicado na introdução.}$$

Sempre procurando rearrumar estas equações de forma a eliminar a dependência do tempo com relação as F_n . As soluções destes sistemas de equações geram função que dependem de certas constantes a serem determinadas pelas condições iniciais.

Plotando-se um ponto a cada instante em um sistema de coordenadas ortogonais, onde cada coordenada representa uma das variáveis que compõe o sistema, origina o espaço de fase do sistema. O conjunto destes pontos origina uma órbita ou trajetória de fase.

As trajetórias nunca se cruzam devido à unicidade de solução do sistema de EDO e à linearização do tempo. Para limitar o espaço de fase e facilitar o estudo costuma-se impor condições de fronteira periódica. Por exemplo, para um sistema que dependa do ângulo, pode-se limitá-lo a um intervalo de $[-\pi, \pi]$.

Quando um espaço de fase se contrai ao passar do tempo significa que o sistema dinâmico é dissipativo. Caso o espaço de fase permaneça constante, o sistema é conservativo. Isto pode ser avaliado pelo divergente:

$$\nabla \cdot F = \nabla \cdot (F_1, F_2, \dots, F_n) = \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \dots + \frac{\partial F_n}{\partial x_n} \Rightarrow \begin{cases} \nabla \cdot F < 0 (\text{Dissipativo}) \\ \nabla \cdot F = 0 (\text{Conservativo}) \end{cases}$$

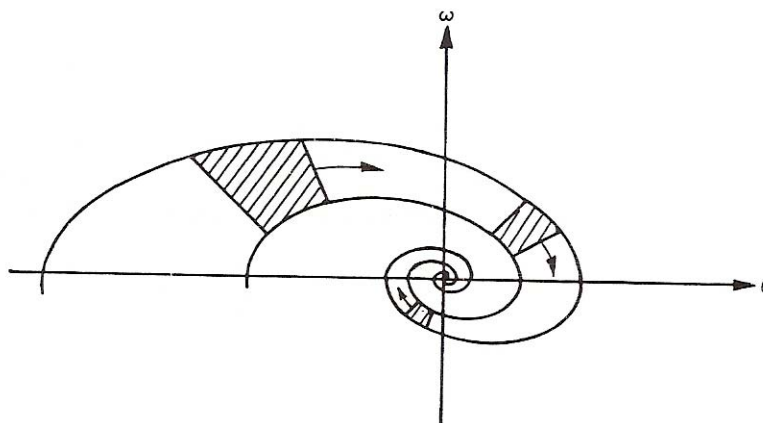


Fig. 1 – Espaço de Estado $\theta \times \omega$ com “encolhimento da área”
 FONTE: Baker

1.2 Atratores

São pontos ou curvas para os quais finitos pontos do espaço de fase convergem para ele. Nestes pontos $\frac{dx_i}{dt} = 0, \forall i \in [1, n]$. Quando os atratores são pontos (chamadas de pontos críticos), pode-se avaliar se as órbitas convergem (ponto estável) ou divergem (ponto instável) dependendo da direção.

Para esta avaliação, lineariza-se as equações de movimento nas regiões próximas do ponto crítico. Depois, avalia-se os autovalores (λ) das soluções ($e^{\lambda t}$): $\Re(\lambda) < 0 \Rightarrow$ ponto estável e $\Re(\lambda) > 0 \Rightarrow$ ponto instável.

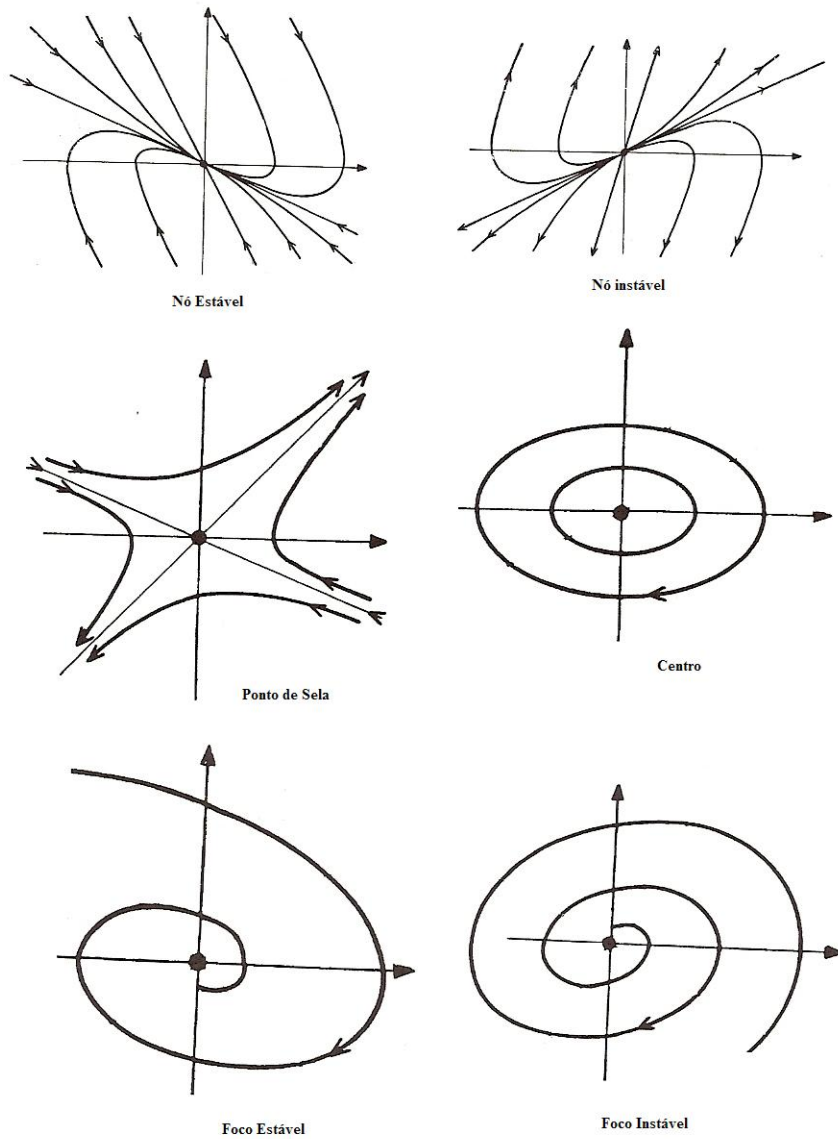


Fig. 2 – Principais tipos de Atratores

1.3 Bacia de atração

Faixa de valores para os quais os pontos de um espaço de fase convergem para um atratores. Duas bacias de atração são separadas pela *separatrix*.

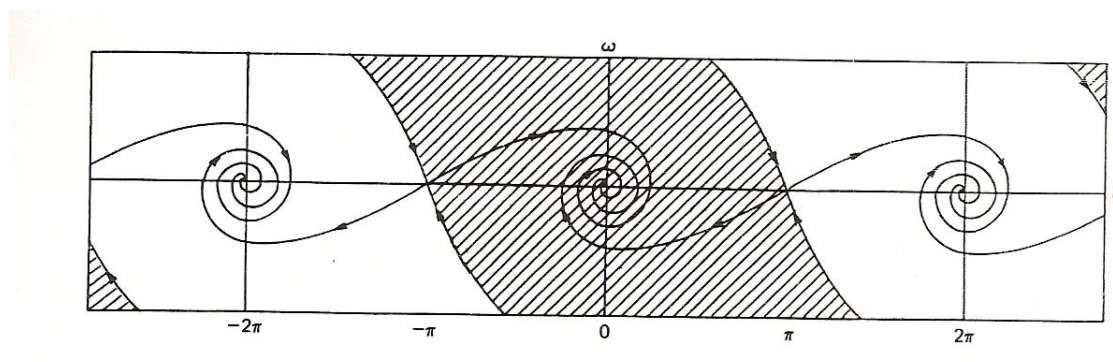


Fig. 3- Bacia de Atração é a área Hachurada para uma condição de fronteira periódica
FONTE: Baker

1.4 Secção de Poincaré

São os pontos de um espaço de fase plotados apenas em intervalos de tempo iguais. Em outras palavras, os pontos de um espaço de fase só são plotados no sistema de coordenadas a cada ciclo predefinido. Enquanto o ciclo não for completado, nenhum ponto é plotado. A frequência com que estes pontos são plotados é a frequência estroboscópica (ω_s).

1.5 Diagrama de Bifurcação

As equações que regem sistemas dinâmicos dependem de certos parâmetros. Dependendo da escolha destes parâmetros, o sistema pode apresentar comportamento periódico ou caótico. Para observar para que faixa de valores de um parâmetro isto ocorre, fixa-se os outros parâmetros e faz-se variar o parâmetro de escolha. Plota-se para cada escolha, uma das variáveis do sistema dinâmico. Observada segundo um ciclo predefinido.

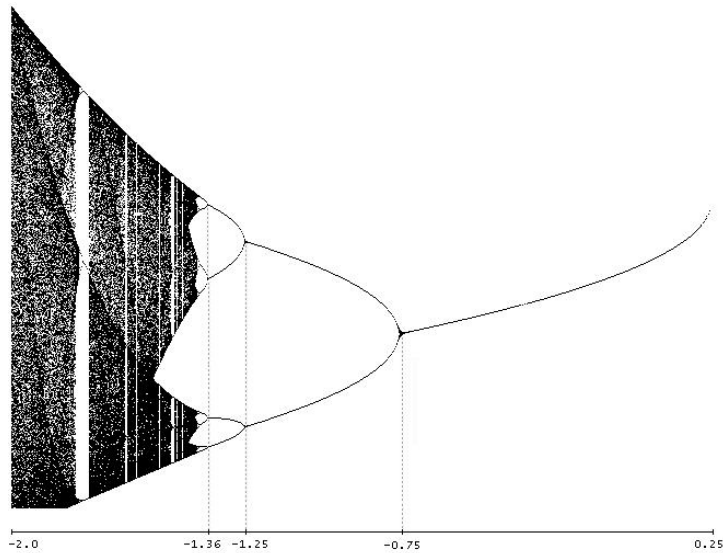


Fig. 4– Diagrama de Bifurcação para a função $f_c(x) = x^2 + c$

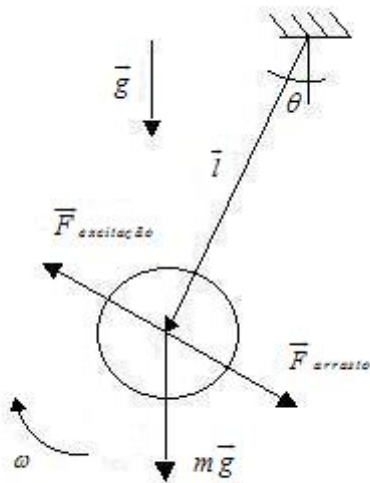
Assim se para um dado valor do parâmetro, existe apenas um ponto, isto implica que o movimento é periódico de período simples. Se existem dois pontos, isto implica que é um movimento periódico de período dois. Se existem infinitos pontos, isto implica que o movimento é caótico. A passagem de um movimento de período simples para um de período dois é chamada de **duplicação de período**.

Existem dentro das regiões caóticas, janelas ou buracos. Nestes pequenos intervalos, o sistema dinâmico volta a ser periódico. Sobre ampliação destas regiões, verifica-se a periodicidade do movimento.

CAPÍTULO 3

PÊNDULO AMORTECIDO FORÇADO

1.6 A equação do pêndulo amortecido forçado



Para o pêndulo simples temos pela segunda

Lei de Newton aplicada a rotação:

$$ml^2 \frac{d^2\theta}{dt^2} = \vec{l} \times m\vec{g} + l \times \vec{F}_{arrasto} + l \times \vec{F}_{excitação}$$

$$ml^2 \frac{d^2\theta}{dt^2} = -mgl \sin \theta - l\gamma\omega + l g_d \cos \Phi, \text{ onde } g_d \text{ e } \gamma \text{ são constantes.}$$

Fig 5 - Esquema do pêndulo amortecido forçado

Tomando $m = g = l = 1$ sem perdas de generalidade:

$$\frac{d^2\theta}{dt^2} = -\sin \theta - \left(\frac{1}{q}\right) \frac{d\theta}{dt} + g_d \cos(\omega_d t)$$

Rearrmando de modo a formar um sistema de EDOs de primeira ordem:

$$\begin{cases} \frac{d\omega}{dt} = -\frac{\omega}{q} - \sin \theta + g_d \cos \Phi & (1) \\ \frac{d\theta}{dt} = \omega & (2) \\ \frac{d\Phi}{dt} = \omega_d & (3) \end{cases}$$

, onde g_d é a força de excitação, q é o fator de amortecimento e ω_d é a frequência de excitação

Percebe-se que as condições para possibilidade de caos foram verificadas: equação (1) não é linear e existem três variáveis (θ, Φ, ω) .

Assim as evoluções deste sistema dinâmico (movimento caótico ou periódico) dependem além das condições iniciais da terna de parâmetros (g_d, q, ω_d) . O estado de fase depende da terna de variável (θ, Φ, ω) , sendo imposta as condições de fronteira periódica $\theta \in [-\pi, \pi]$ e $\Phi \in [0, 2\pi]$.

O estudo deste sistema só importa quando atingido o “steady-state”, quando o tempo é suficientemente grande para passar o transiente.

Serão observados fases de estado 2D $(\theta \times \omega)$. As fases de estado que caracterizam movimento periódico são as fases que apresentam curvas fechadas. Isto mostra que os estados se repetem, caracterizando um movimento periódico. As fases de estado que caracterizam um movimento caótico são infinitas linhas. Isto mostra que o movimento não repete estados e sim se enrola em torno de si. Formando atratores caóticos de infinitas camadas. Assim ao se observar a fase sobre o plano $\theta \times \omega$ parece que as linhas se sobrepõem.

A secção de Poincaré desta dinâmica é o plano $(\theta \times \omega)$. Quando a secção de Poincaré apresenta pontos isolados, isto indica que o movimento é periódico com o período caracterizado pelo número de pontos. Quando a secção de Poincaré apresenta infinitos pontos, caracteriza um movimento caótico (pode-se pensar como um movimento de período infinito).

Os diagramas de bifurcação são obtidos plotando ω a cada ciclo em função de um dos parâmetros (g_d, q, ω_d) para determinada condição inicial.

CAPÍTULO 4

RESULTADOS

Os resultados foram obtidos através da utilização dos programas de computadores em ambientação C no Apêndice A. Um deles obtém o estado de fase, outro a secção de Poincaré, outro o diagrama de bifurcação e o último simula a oscilação do pendulo para os diferentes parâmetros escolhidos. Todos eles utilizaram o método de Runge-Kutta de 4ª ordem para a resolução do sistema de EDO's.

Alguns resultados interessantes foram obtidos através da utilização dos programas mencionados no Apêndice A, com o tempo mínimo 1000, tempo máximo 10000, ângulo de análise da Secção de Poincaré 0 e $\omega_s = 1$:

- Pode se observar pela utilização do diagrama de bifurcação que os valores interessantes para ω_d são aqueles próximos a 1, utilizou-se principalmente 2/3.

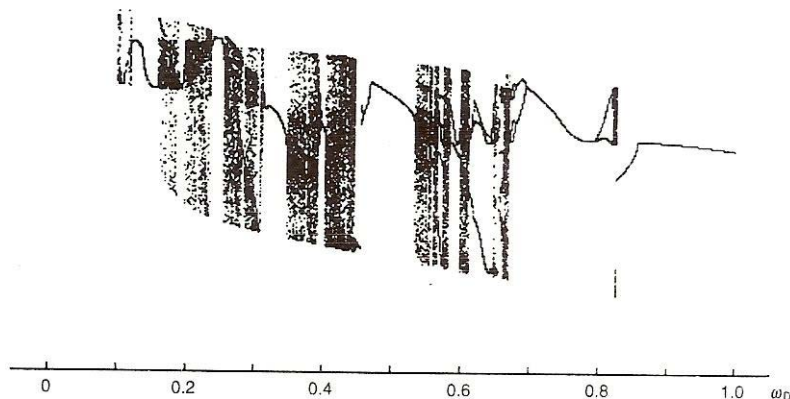


Fig. 7 – Diagrama de bifurcação para $g = 1,5$ e $q = 2$ em função de ω_d

- Ao se modificar os valores iniciais do ângulo para análise da secção de Poincaré, são obtidos figuras semelhantes, mas que parecem se “mover” sobre o diagrama de fase.

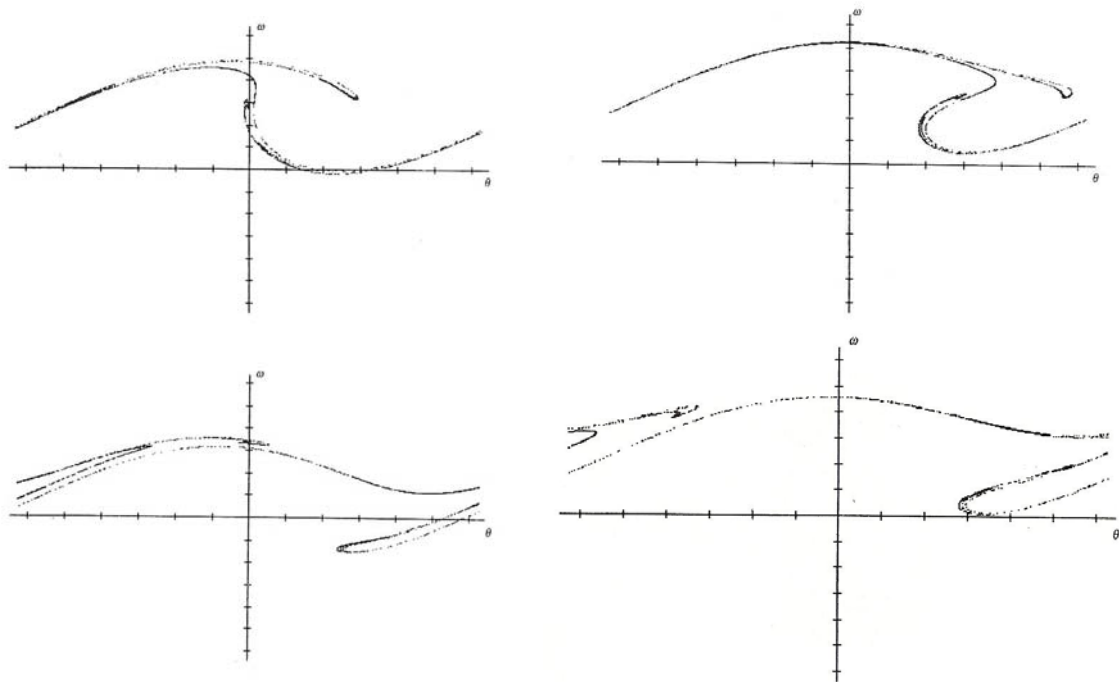


Fig. 8 – Secção de Poincaré tomada a ângulos de análise diferentes para $g = 1,5$; $q = 2$ e $\omega_d = 2/3$: $\Phi = 0$, $\Phi = 0,628319$, $\Phi = 1,25664$ e $\Phi = 1,88496$, respectivamente.

- Ao se observar as secções de Poincaré para evoluções caóticas, os atratores caóticos apresentam a propriedade “self-similarity” (semelhança sobre ampliação) e dimensões não inteiras (fractais).
- Ao se aumentar o fator de amortecimento; o atrator caótico fica mais abrangente.

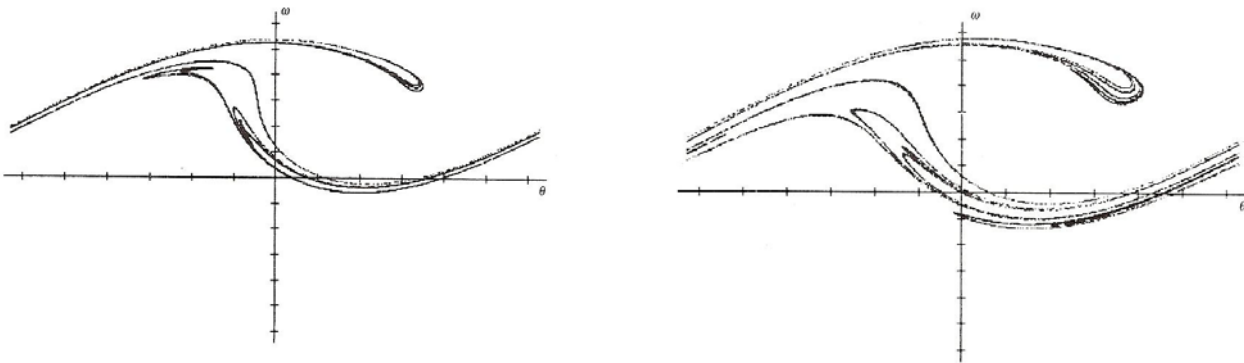


Fig. 9 – Secção de Poincaré para $g=1,5$ e $\omega_d = 2/3$ com $q = 2,8$ e $q = 4$, respectivamente.

- Pode-se observar que a secção de Poincaré está contida inteiramente em uma das bacias de atração em movimentos periódicos. Enquanto para movimentos caóticos, a bacia de atração se espalha pelo atrator caótico, mostrando que suas fronteiras são fractais e que duas condições iniciais próximas por estarem no atrator apresentam trajetórias incertas.

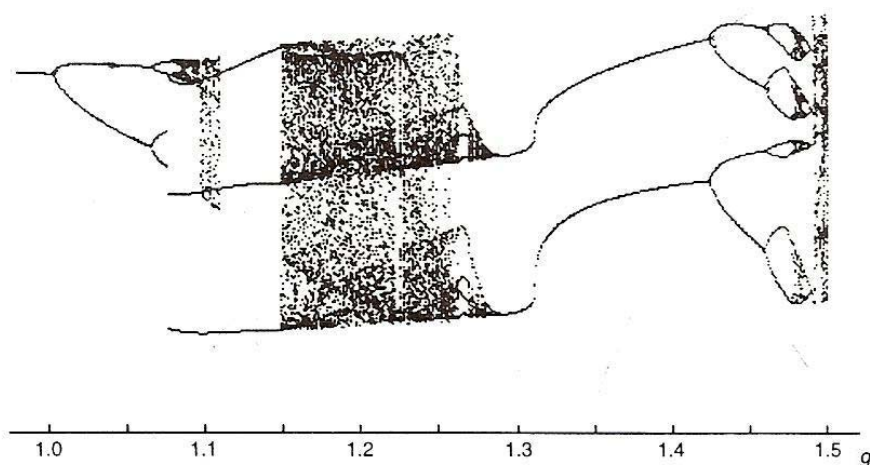
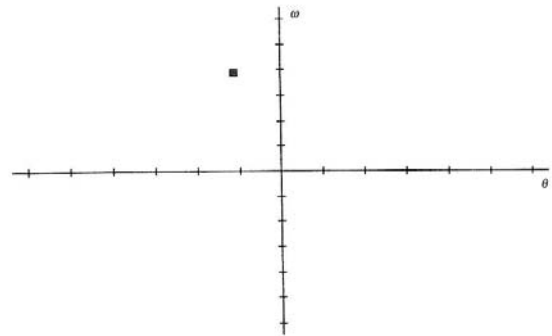
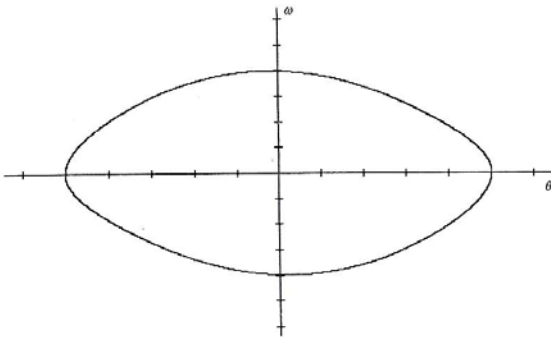


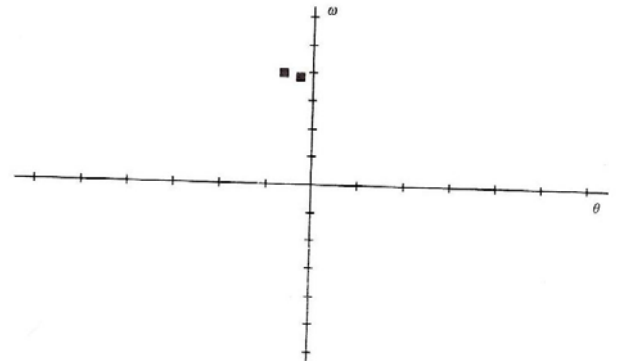
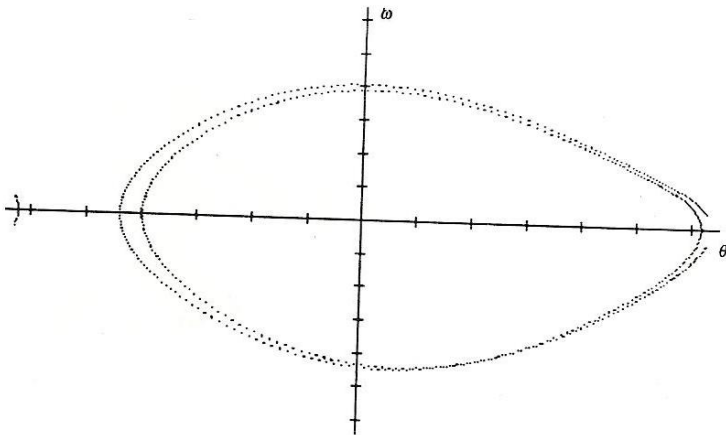
Fig. 10 – Diagrama de bifurcação para $q = 2$ e $\omega_d = 2/3$ em função de g .

Baseado neste diagrama de bifurcação, obtemos alguns exemplos de evoluções interessantes que podem ser comprovadas pelo programa que simula o movimento do pêndulo. Abaixo os diagramas de fase e as seções de Poincaré para:

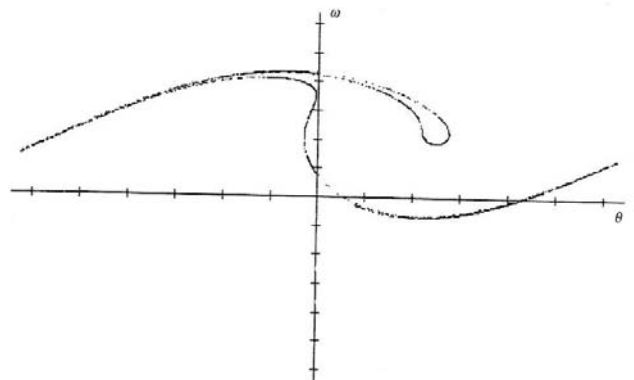
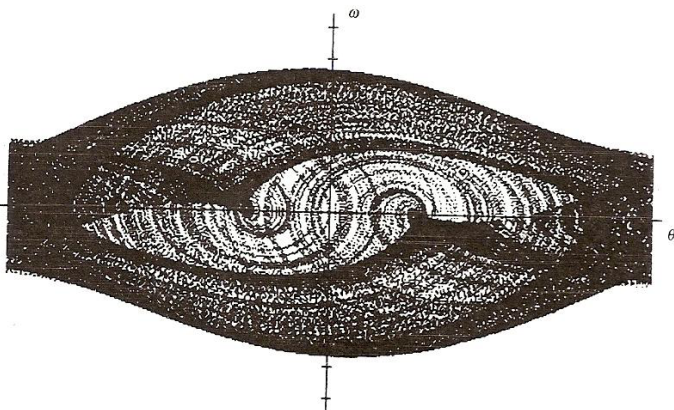
$g = 0,9$:



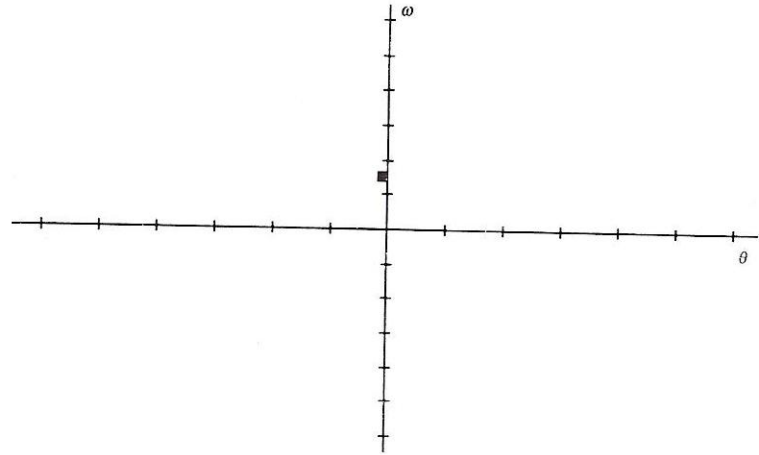
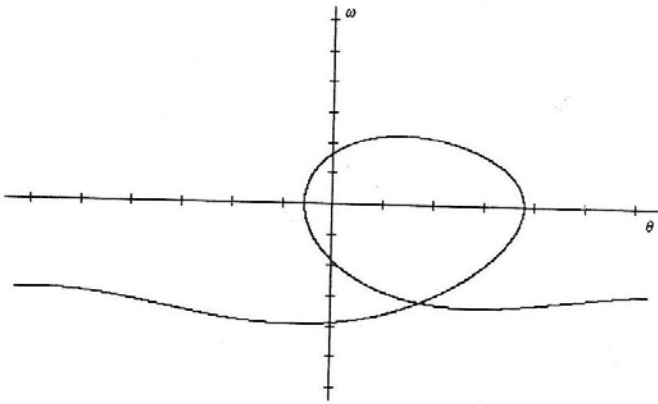
$g = 1.07$:



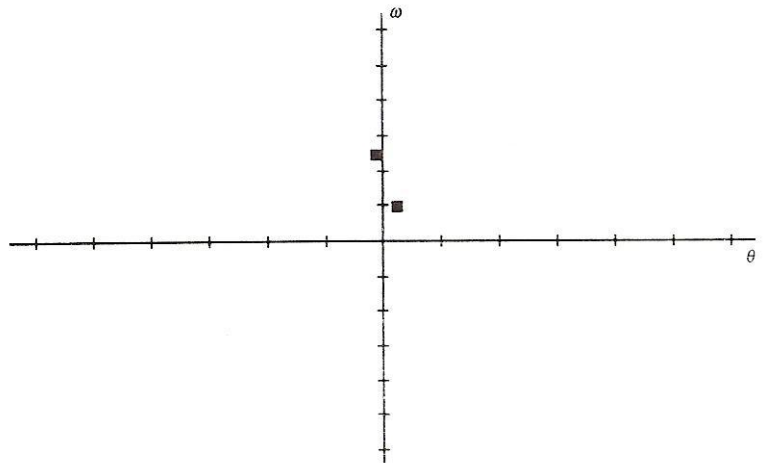
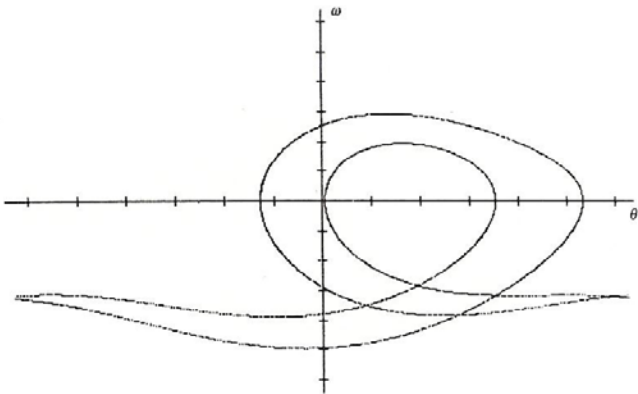
$g = 1.15$:



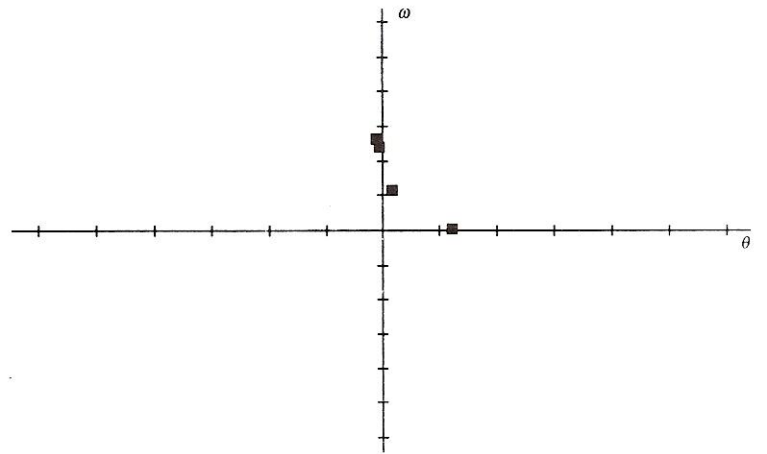
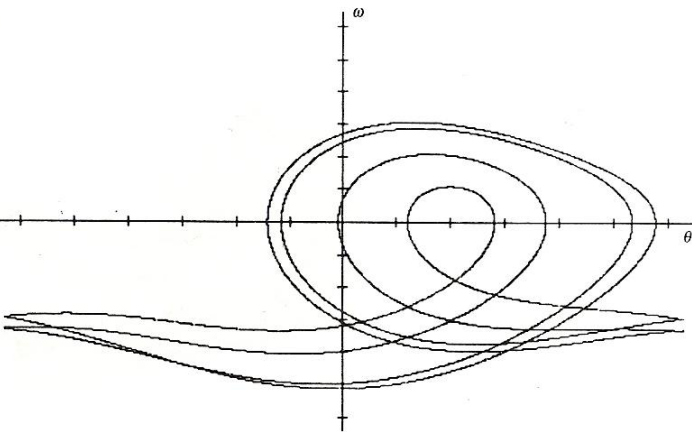
$g = 1,35:$



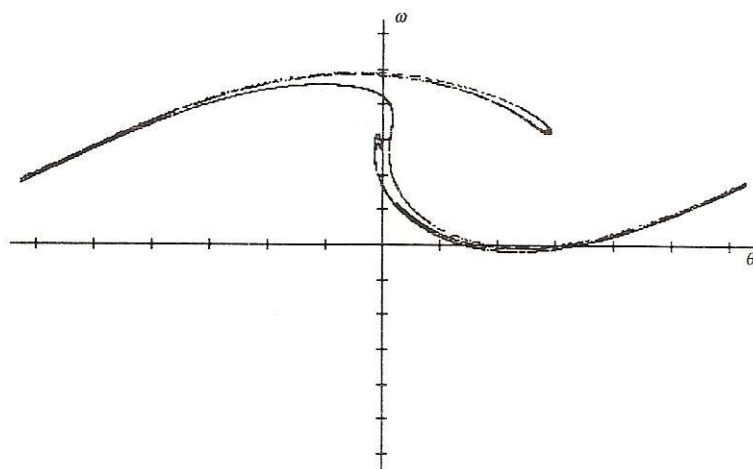
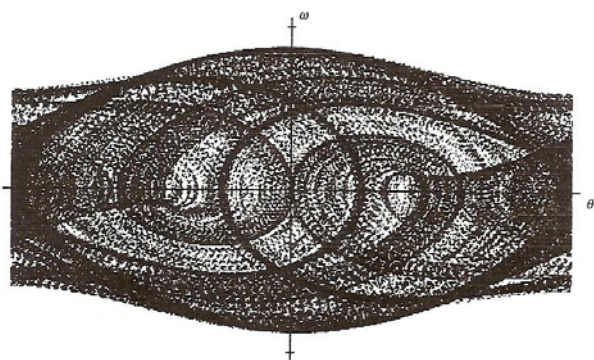
$g = 1,45:$



$g = 1,47 :$



$g = 1,50:$



CAPÍTULO 5

CONCLUSÃO

Concluimos que essa iniciação forneceu as ferramentas necessárias para a análise de qualquer sistema dinâmico. Desde de a formulação do problema e sua modelação segundo parâmetros. Como métodos gráficos e computacionais de análise a partir dos parâmetros que governam a evolução do sistema. Obtendo os valores de parâmetros que mais interessam de forma a obter um movimento caótico ou periódico.

Este trabalho pode se estender para diversas áreas já que sistemas dinâmicos vão desde de um simples pêndulo ao mais complexo veículo aeroespacial.

REFERÊNCIAS BIBLIOGRÁFICAS

Baker, G. L.; Gollub, J. P. **Chaotic Dynamics**: an introduction. Cambridge: University Press 1996. 255p.

Gleick, James; **Chaos**: Making a new science. Penguin 1988. 368p.

APÊNDICE A

PROGRAMAS EM AMBIENTE C UTILIZADOS PARA OBTER OS RESULTADOS DESCRITOS. PARA UTILIZAR OS PROGRAMAS DEVE-SE TER O BORLAND C++ OU CRIAR UMA PASTA C:\\BC5 E NELA INSERIR A BIBLIOTECA BGI PARA GRAFICOS DO DOS.

```
/* PROGRAMA PARA O MOVIMENTO DO PENDULO*/

#   i   n   c   l   u   d   e       <   s   t   d   i   o   .   h   >
#include <graphics.h>
#include <conio.h>
#include <math.h>
#define dt 0.0050000000000000

long double teta, w, g, q, wd, t, tdelay;

long double acel_pend (long double teta,long double w,long double g,long double q,long double wd,long
double t){
return (-sinl(teta)-(w/q)+g*cosl(wd*t));
}

void dinamica (){
    long double dxa,dxb,dxc,dxd,dwa,dwb,dwc,dwd;
    dxa = dt*w;
    dwa = dt*acel_pend(teta,w,g,q,wd,t);
    dxb = dt*(w+dwa/2);
    dwb = dt*acel_pend(teta+dxa/2,w+dwa/2,g,q,wd,t+dt/2);
    dxc = dt*(w+dwb/2);
    dwc = dt*acel_pend(teta+dxb/2,w+dwb/2,g,q,wd,t+dt/2);
    dxd = dt*(w+dwc);
    dwd = dt*acel_pend(teta+dxc,w+dwc,g,q,wd,t+dt);
    teta = teta + (dxa +2*dxb +2*dxc +dxd)/6;
    w = w + (dwa +2*dwb+ 2*dwc +dwd)/6;
    t = t + dt;
}

void desenha_pend (){

    cleardevice();
    for(long double i = 0; i<= tdelay ; i++);
    circle(319+200*sinl(teta),220+200*cosl(teta),20);
    line(319+200*sinl(teta),220+200*cosl(teta),319,220);
}

void main(void)
{
int g_driver, g_mode;
int left, top, bottom, right;

detectgraph(&g_driver, &g_mode);
initgraph(&g_driver, &g_mode, "C:\\BC5\\BGI");
left = 0; right = 638; top = 0; bottom = 478;

printf("***** Animacao do Pendulo *****");
printf("\n\n\tOBS: Pressione qualquer tecla para sair da animacao apos iniciada.");
printf("\n\n\tDigite:");
```

```

printf("\n\tValor inicial do angulo de inclinacao do pendulo(radianos): ");
scanf("%Lf",&teta);
printf("\n\tValor inicial da velocidade angular do pendulo(radianos/s): ");
scanf("%Lf",&w);
printf("\n\tValor da forca de excitacao: ");
scanf("%Lf",&g);
printf("\n\tValor do fator de amortecimento: ");
scanf("%Lf",&q);
printf("\n\tValor da velocidade angular de excitacao: ");
scanf("%Lf",&wd);
printf("\n\tValor do tempo de ajuste da animacao: ");
scanf("%Lf",&tdelay);
t = 0;

while(!kbhit()){

    desenha_pend();
    dinamica();

};

getch();
closegraph();
return;
}

```

```
/* PROGRAMA PARA FASE DE ESTADOS*/
```

```
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
#include <math.h>
#define dt 0.0050000000000000

long double teta, w, g, q, wd, t, tmin, tmax;

long double acel_pend (long double teta,long double w,long double g,long double q,long
double wd,long double t){
    return (-sinl(teta)-(w/q)+g*cosl(wd*t));
}

void dinamica (){
    long double dxa,dxb,dxc,dxd,dwa,dwb,dwc,dwd;
    dxa = dt*w;
    dwa = dt*acel_pend(teta,w,g,q,wd,t);
    dxb = dt*(w+dwa/2);
    dwb = dt*acel_pend(teta+dxa/2,w+dwa/2,g,q,wd,t+dt/2);
    dxc = dt*(w+dwb/2);
    dwc = dt*acel_pend(teta+dxb/2,w+dwb/2,g,q,wd,t+dt/2);
    dxd = dt*(w+dwc);
    dwd = dt*acel_pend(teta+dxc,w+dwc,g,q,wd,t+dt);
    teta = teta + (dxa +2*dxb +2*dxc +dxd)/6;
    w = w + (dwa +2*dwb+ 2*dwc +dwd)/6;
    t = t + dt;
}

void inigraf(){
    cleardevice();
    setcolor(MAGENTA);
    line(0,239,638,239);
    moveto(630,240);
    outtext("o");
    line(319,0,319,478);
    moveto(321,1);
    outtext("w");
}

void plotar(){
    putpixel(319+319*teta/(M_PI),239-239*w/(2*M_PI),WHITE);
}

void main(void)
{
int g_driver, g_mode;
```

```

int left, top, bottom, right;

detectgraph(&g_driver, &g_mode);
initgraph(&g_driver, &g_mode, "C:\\BC5\\BGI");
left = 0; right = 638; top = 0; bottom = 478;

printf("***** Diagrama de Fase
*****");
printf("\n\n\tDigite:");
printf("\n\tValor inicial do angulo de inclinacao do pendulo(radianos): ");
scanf("%Lf",&teta);
printf("\n\tValor inicial da velocidade angular do pendulo(radianos/s): ");
scanf("%Lf",&w);
printf("\n\tValor da forza de excitacao: ");
scanf("%Lf",&g);
printf("\n\tValor do fator de amortecimento: ");
scanf("%Lf",&q);
printf("\n\tValor da velocidade angular de excitacao: ");
scanf("%Lf",&wd);
printf("\n\tValor do tempo minimo para inicio do grafico: ");
scanf("%Lf",&tmin);
printf("\n\tValor do tempo maximo de interacoes para o grafico: ");
scanf("%Lf",&tmax);
t = 0;

inigraf();

for(long double i = 0;i<=tmax;i+=dt){

    dinamica();
    if(fabs(teta)>M_PI) teta = teta - 2*M_PI*fabs(teta)/teta;
    if(i>=tmin) plotar();

}

getch();
closegraph();
return;
}

```

```
/* PROGRAMA PARA SECÇÃO DE POINCARÉ*/
```

```
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
#include <math.h>
#define eps pow(10,-6)
```

```
long double teta, w, g, q, wd, t, tmin, tmax, ang, ws, dt, delta;
```

```
long double acel_pend (long double teta,long double w,long double g,long double q,long
double wd,long double t){
    return (-sinl(teta)-(w/q)+g*cosl(wd*t));
}
```

```
void dinamica (){
    long double dxa,dxb,dxc,dxd,dwa,dwb,dwc,dwd;
    dxa = dt*w;
    dwa = dt*acel_pend(teta,w,g,q,wd,t);
    dxb = dt*(w+dwa/2);
    dwb = dt*acel_pend(teta+dxa/2,w+dwa/2,g,q,wd,t+dt/2);
    dxc = dt*(w+dwb/2);
    dwc = dt*acel_pend(teta+dxb/2,w+dwb/2,g,q,wd,t+dt/2);
    dxd = dt*(w+dwc);
    dwd = dt*acel_pend(teta+dxc,w+dwc,g,q,wd,t+dt);
    teta = teta + (dxa +2*dxb +2*dxc +dxd)/6;
    w = w + (dwa +2*dwb+ 2*dwc +dwd)/6;
}
```

```
void inigraf(){
    cleardevice();
    setcolor(MAGENTA);
    line(0,239,638,239);
    moveto(630,240);
    outtext("o");
    line(319,0,319,478);
    moveto(321,1);
    outtext("w");
}
```

```
void main(void)
{
int g_driver, g_mode;
int left, top, bottom, right;
```

```

detectgraph(&g_driver, &g_mode);
initgraph(&g_driver, &g_mode, "C:\\BC5\\BGI");
left = 0; right = 638; top = 0; bottom = 478;

printf("*****Seccao de Poincaré
*****");
printf("\n\n\tDigite:");
printf("\n\tValor inicial do angulo de inclinacao do pendulo(radianos): ");
scanf("%Lf",&teta);
printf("\n\tValor inicial da velocidade angular do pendulo(radianos/s): ");
scanf("%Lf",&w);
printf("\n\tValor da forza de excitacao: ");
scanf("%Lf",&g);
printf("\n\tValor do fator de amortecimento: ");
scanf("%Lf",&q);
printf("\n\tValor da velocidade angular de excitacao: ");
scanf("%Lf",&wd);
printf("\n\tValor do tempo minimo para inicio do grafico: ");
scanf("%Lf",&tmin);
printf("\n\tValor do tempo maximo de interacoes para o grafico: ");
scanf("%Lf",&tmax);
printf("\n\tValor do angulo inicial (1/2*pi) para a analise da secao de Poincare: ");
scanf("%Lf",&ang);
printf("\n\tValor do fator estroboscopico da analise: ");
scanf("%Lf",&ws);
t = 0;
ang = 2*M_PI*ang;
ws = 2*M_PI*ws;
dt = 0.5;

inigraf());

do {
    long double w1, teta1, wt = w, tetat = teta, dtt = dt, tt = t, delta;

    dinamica();

    w1 = w; teta1 = teta;
    w = wt; teta = tetat;
    dt = dt/2;

    dinamica();
    t += dt;
    dinamica();

    if(fabssl(w-w1) > fabssl(teta-teta1)) delta = fabssl(w-w1);
    else delta = fabssl(teta-teta1);
}

```

```

teta = tetat; w = wt;
t = tt; dt = dtt;

    if(delta < eps){
if( t >= tmin){
    if(wd*(t+dt) >= ang){
        dt = ang/wd-t;
        dinamica();
        putpixel(319+319*(teta/M_PI),239-239*(w/M_PI),WHITE);
        dt = dtt;
        ang+=ws;
    }
}
    if( t<tmin && wd*(t+dt)>=ang) ang+=ws;
teta = tetat;
w = w1;
t+=dt;
    dt = dt*0.95*powl(eps/delta,0.25);
}
else dt = dt*0.95*powl(eps/delta,0.2);
if(fabsl(teta)> M_PI) teta = teta - 2*M_PI*fabsl(teta)/teta;
}while(t <= tmax);

getch();
closegraph();
return;
}

```



```
/* PROGRAMA PARA DIAGRAMA DE BIFURCAÇÃO*/
```

```
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
#include <math.h>
#define eps pow(10,-6)

int func;
long double teta, w, g, q, wd, t, tmin, tmax, wd0, q0, g0, teta0, w0, ang, ang0, dt, xmin,
xmax;

long double acel_pend (long double teta,long double w,long double g,long double q,long
double wd,long double t){
    return (-sinl(teta)-(w/q)+g*cosl(wd*t));
}

void dinamica (){
    long double dxa,dxb,dxc,dxd,dwa,dwb,dwc,dwd;
    dxa = dt*w;
    dwa = dt*acel_pend(teta,w,g,q,wd,t);
    dxb = dt*(w+dwa/2);
    dwb = dt*acel_pend(teta+dxa/2,w+dwa/2,g,q,wd,t+dt/2);
    dxc = dt*(w+dwb/2);
    dwc = dt*acel_pend(teta+dxb/2,w+dwb/2,g,q,wd,t+dt/2);
    dxd = dt*(w+dwc);
    dwd = dt*acel_pend(teta+dxc,w+dwc,g,q,wd,t+dt);
    teta = teta + (dxa +2*dxb +2*dxc +dxd)/6;
    w = w + (dwa +2*dwb+ 2*dwc +dwd)/6;
}

void inigraf(){
    cleardevice();
    setcolor(MAGENTA);
    line(0,239,638,239);
    moveto(630,240);
    if(func == 1)outtext("g");
    else if(func == 2) outtext("q");
    else if(func == 3) outtext("wd");
    line(319,0,319,478);
    moveto(321,1);
    outtext("w");
}

void main(void)
{
int g_driver, g_mode;
```

```

int left, top, bottom, right;

detectgraph(&g_driver, &g_mode);
initgraph(&g_driver, &g_mode, "C:\\BC5\\BGI");
left = 0; right = 638; top = 0; bottom = 478;

printf("***** Diagrama de Bifurcacao
*****");
printf("\n\n\tDigite:");
printf("\n\tValor inicial do angulo de inclinacao do pendulo(radianos): ");
scanf("%Lf",&teta0);
printf("\n\tValor inicial da velocidade angular do pendulo(radianos/s): ");
scanf("%Lf",&w0);
printf("\n\tA velocidade angular sera funcao de:\n(1) - Forca de excitacao\n(2) - Fator de
amortecimento\n(3) - Velocidade angular de excitacao:\nOpcao: ");
scanf("%d",&func);
if(func != 1){
printf("\n\tValor da forca de excitacao: ");
scanf("%Lf",&g0);
}
if(func != 2){
printf("\n\tValor do fator de amortecimento: ");
scanf("%Lf",&q0);
}
if(func != 3){
printf("\n\tValor da velocidade angular de excitacao: ");
scanf("%Lf",&wd0);
}
printf("\n\tValor do tempo minimo para inicio do grafico: ");
scanf("%Lf",&tmin);
printf("\n\tValor do tempo maximo de interacoes para o grafico: ");
scanf("%Lf",&tmax);
printf("\n\tValor de inicio do eixo x: ");
scanf("%Lf",&xmin);
printf("\n\tValor de termino do eixo x: ");
scanf("%Lf",&xmax);
printf("\n\tValor do angulo(1/2*pi) para o qual a velocidade angular sera avaliado: ");
scanf("%Lf",&ang0);

inigraf();

for(long double j = xmin; j<=xmax; j += (xmax-xmin)/638){
    if(func == 1) g0 = j;
    else if(func == 2) q0 = j;
    else if(func == 3) wd0 = j;
}

```

```

t = 0;
w = w0;
teta = teta0;
wd = wd0;
q = q0;
g = g0;
ang = 2*M_PI*ang0;
dt = 0.5;

do{
    long double w1, teta1, wt = w, tetat = teta, dtt = dt, tt = t, delta;

    dinamica();

    w1 = w; teta1 = teta;
    w = wt; teta = tetat;
    dt = dt/2;

    dinamica();
    t += dt;
    dinamica();

    if(fabs(w-w1) > fabs(teta-teta1)) delta = fabs(w-w1);
    else delta = fabs(teta-teta1);

    teta = tetat; w = wt;
    t = tt; dt = dtt;

    if(delta < eps){
    if( t >= tmin){
        if(wd*(t+dt) >= ang){
            dt = ang/wd-t;
            dinamica();
            putpixel(638*(j-xmin)/(xmax-xmin),239-239*(w/M_PI),WHITE);
            dt = dtt;
            ang+=2*M_PI;
        }
    }

    if( t<tmin && wd*(t+dt)>=ang) ang+=2*M_PI;
    teta = teta1;
    w = w1;
    t+=dt;
    dt = dt*0.95*powl(eps/delta,0.25);
    }
    else dt = dt*0.95*powl(eps/delta,0.2);
}

```

```
if(fabsl(teta)> M_PI) teta = teta - 2*M_PI*fabsl(teta)/teta;
}while(t <= tmax);
}
getch();
closegraph();
return ;
}
```