Journal of İstanbul Kültür University
2006/4 pp. 179-190

DİNAMİK SİSTEMLER ÜZERİNE MATHEMATICA UYGULAMALARI

Ünal UFUKTEPE[1], Aslı DENİZ[2]

ÖZET

Teknolojik devrimler ve gelişen bilişim yazılımları cebirsel sistemlerin uygulama alanlarını genişletmiştir. Dinamik sistemler için 1984'ten itibaren Phaser yazılımı yaygın şekilde kullanılırken 1994'te Dynamics kullanılmaya başlanmış 2000'li yıllarda ise Mathematica, Matlab, Mathcad v.b. programlar bunların yerini almıştır. Bu yazılımlar sayesinde dinamik sistemlerin yörüngelerinin elde edilmesi, Lyapunov foksiyonlarının bulunması, periyodik çözümlerin sembolik olarak elde edilmesi, garip çekerler, doğrusal olmayan dinamik sistemlerin kaotik yapılarının daha iyi anlaşılması kolaylaşmıştır. Bu çalışmada dinamik sistemlerin temel kavramları üzerine geliştirmiş olduğumuz Mathematica ve webMathematica uygulamalarına yer verdik. Bu uygulamalar sayesinde öğrencilerinin dinamik sistemleri daha iyi kavrayabileceğini ve kendi alanlarında modellerini geliştirebileceklerini düşünüyoruz.

1. GİRİŞ

Doğrusal olmayan dinamik denklemlerle çalışan Henri Poincare, belirli sistemlerin başlangıç koşullarına aşırı duyarlı olduğunu göstermiş, ancak bu keşif, bilgisayar devri başlayana kadar insanları pek mutlu etmemiştir. Bilgisayar teknolojisinin hızlı gelişimiyle birlikte matematikçiler daha önce hiç yapamadıklarını bilgisayarlarda denemeye başladılar. Bu sayede 1963 yılında, Edward Lorenz hava durumu tahminleri üzerine geliştirdiği

$$\begin{aligned} \dot{x} &= \sigma(y-x) \\ \dot{y} &= rx - y - xz \\ \dot{z} &= xy - dz \end{aligned} \qquad (1.1)$$

diferansiyel denklem sisteminin başlangıç koşullarına göre çözümlerini bilgisayar ortamında bulmaya çalışırken, sistemin çözümlerinin tahmin edilemez ve kaotik olduğunu keşfetti [1]. Çözüm, uygun katsayılar kullanıldığında kelebek içinde bir bölgeye düştüğünden oluşturduğu şekil Lorenz kelebeği olarak bilinir.

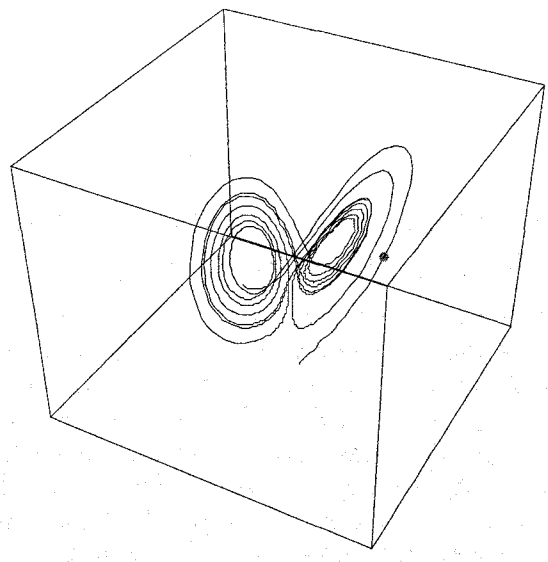
Aşağıdaki Mathematica kodlarıyla Lorenz kelebeği çiziliyor ve başlangıç koşulundaki farklılığın çözümü nasıl değiştirdiği gözlenebiliyor.

```
In[1]: = sol=NDSolve[{x'[t]□3 (y[t]-x[t]),y'[t]□26.5 x[t]-y[t]-x[t] z[t],z'[t]□x[t] y[t]-
z[t],x[0]□1,y[0]□1,z[0]□1},{x,y,z},{t,20},MaxSteps→10^4];
In[2]:= <<RealTime3D`
In[3]:= ParametricPlot3D[Evaluate[{x[t],y[t],z[t]}/.sol],{t,0,20},PlotPoints→1000]
```

Sistemin oluşturduğu yörüngeyi ve bu yörüngede hareket eden bir cismin hareketi ve izi:

[1] İzmir Yüksek Teknoloji Enstitüsü, Matematik Bölümü, Gülbahçe Köyü, Urla, İzmir, (232)-750 76 14, unalufuktepe@iyte.edu.tr
[2] İzmir Yüksek Teknoloji Enstitüsü, Matematik Bölümü, Gülbahçe Köyü, Urla, İzmir, (232)-750 75 62 aslideniz@iyte.edu.tr

```
In[4]:= s1=Flatten[Table[Evaluate[{x[t],y[t],z[t]}/.sol],{t,0,20,0.02}],1];
In[5]:= L=Length[s1]
In[6]:=Table[Show[Graphics3D[{{Table[Line[{s1[[i+1]],s1[[i]]}],{i,1,L-1}]},
{Hue[.01],PointSize[0.02],Point[s1[[i]]]}}],PlotRange→{{-30,30},{-30,30},{-0,50}},AspectRatio→
Automatic],{i,0,L-1,1}];
```



komutlarının işletilmesiyle gözleniyor.
Bu çalışmanın birinci bölümünde doğrusal dinamik sistemlerin temel kavramları yer almaktadır. İkinci bölümde doğrusal ve doğrusal olmayan sistemlerin çözümleri, peryodik çözümler, çözümlerin kaotik yapıları Mathematica kodlarıyla birlikte verilmekte, son bölümde ise bu uygulamalar web ortamında webMathematica aracılığıyla dinamik bir metin yapısıyla verilebileceği gösterilmektetir.

2. DOĞRUSAL OTONOM SİSTEMLERİNİN FAZ YOLLARI

$$\dot{x} = ax + by, \quad \dot{y} = cx + dy \tag{2.1}$$

doğrusal sisteminin faz yörüngesinin genel karakteri zamana bağlı

$$x(t) = c_1 v_1 e^{\lambda_1 t} + c_2 v_2 e^{\lambda_2 t} \tag{2.2}$$

$$x(t) = \mathrm{Re}\left\{cve^{(\alpha+\beta i)}\right\} \tag{2.3}$$

çözümlerinden elde edilir. Daha basit bir yaklaşımla faz yörüngesinin

$$\frac{dy}{dx} = \frac{cx + dy}{ax + by} \tag{2.4}$$

diferansiyel denkleminin çözümünden elde edileceği düşünülebilir ancak bu denklemlerin çözümü için kullanılan standart yöntemler x ile y arasında geometrik olarak gösterilmesi zor kapalı bir bağıntıya neden olur.
Faz diyagramı desenleri,

$$p = a + d\ ,\ q = ad - bc \neq 0 \qquad (2.5)$$

olacak şekilde

$$\lambda^2 - p\lambda + q = 0 \qquad (2.6)$$

karakteristik denkleminin çözümü olan λ_1 ve λ_2 özdeğerlerine bağlı olarak üç sınıfa ayrılır:

a) λ_1 ve λ_2 değerlerinin reel, birbirinden farklı ve işaretlerinin aynı olması,

b) λ_1 ve λ_2 değerlerinin reel ve zıt işaretli olması,

c) λ_1 ve λ_2 değerlerinin birbirinin kompleks eşleniği olması durumu.

a) λ_1 ve λ_2 değerlerinin reel, birbirinden farklı ve işaretlerinin aynı olması durumu:

Örnek 2.1.

$$\begin{aligned} \dot{x} &= 6x - 8y \\ \dot{y} &= x \end{aligned} \qquad (2.7)$$

doğrusal dinamik sisteminin çözümünü ve vektör alanını çizelim.

Verilen değerler kullanılarak oluşturulan karakteristik denkleminin köklerinin **a)** durumunu sağladığı görülür.

Öncelikle katsayılar matrisi girilir:

In[1]:= **A={{6,-8},{1,0}}**
Out[1]:= {{6,-8},{1,0}}

Sistemin özdeğerleri ve özvektörleri bulunur.

In[2]:= **Eigensystem[A]**
Out[2]:= {{4,2},{{4,1},{2,1}}}

Çıktının ilk terimi {4,2}, sistemin (aynı işaretli ve birbirinden farklı) özdeğerlerini, {{4,1},{2,1}} ise bunlara karşılık gelen özvektörlerini verir. Çözüm ise

$$\begin{pmatrix} x \\ y \end{pmatrix} = c_1 \begin{pmatrix} 4 \\ 1 \end{pmatrix} e^{4t} + c_2 \begin{pmatrix} 2 \\ 1 \end{pmatrix} e^{2t} \qquad (2.8)$$

biçimindedir. Burada

$$\begin{aligned} p &= 6 + 0 = 6 \\ q &= 6.0 - (-8).1 = 8 > 0 \end{aligned} \qquad (2.9)$$

olduğundan sistem kararlı olmayan bir node'a sahiptir. Bu işlemleri bir paket içinde derleyen Gianluca Gorni'nin geliştirmiş olduğu CurvesGraphics (http://www.dimi.uniud.it/~gorni) paket programı ile yukardaki sistemin özdeğerleri ve faz yollarının sınıflandırılması elde edilir; öncelikle paket aktif hale getirilir;

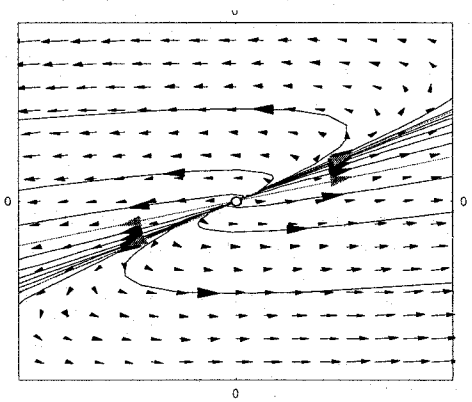
In[3]:= **Needs["CurvesGraphics`"]**

Daha sonra doğrusal sistemin katsayılar matrisi aşağıdaki formda yazılır:

In[4]:= **LinearPhasePlot2D[** $\begin{pmatrix} 6 & -8 \\ 1 & 0 \end{pmatrix}$**, ShowField->True]**

Out[4]:=

Repulsive node



b) Özdeğerlerin reel ve zıt işaretli olma durumu:
Oluşan desen, asimptotlarıyla beraber hiperboller ailesine benzer. Bu durumda orijindeki denge noktası, bir eyer noktasıdır. Eyer noktası için koşul:

$$\Delta = p^2 - 4q > 0,\ q < 0 \qquad (2.10)$$

şeklindedir ve eyer noktası her zaman kararsızdır.

Örnek 2.2

$$\dot{x} = 3x - 2y$$
$$\dot{y} = 5x - 4y \qquad (2.11)$$

doğrusal sistemin zamana bağlı çözümlerini bulup faz diyagramını çizdirelim.

Sistem,

$$\Delta = p^2 - 4q = 9 > 0 \text{ ve } q = -2 < 0 \qquad (2.12)$$

koşulunu sağladığından eyer noktasına sahip olduğunu söyleriz. Asimptotlar orijinden geçen

doğrulardır. **Örnek 2.1** deki Mathematica kodlarında (2.11)'in katsayıları kullanılarak

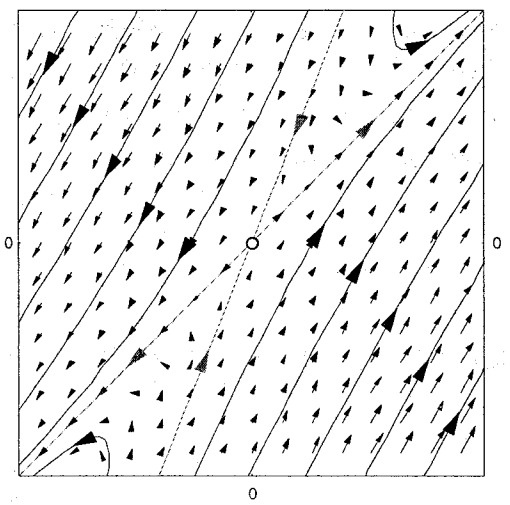
$$x(t) = c_1 e^{t} + 2c_2 e^{-2t}$$
$$y(t) = c_1 e^{t} + 5c_2 e^{-2t} \qquad (2.13)$$

çözümü elde edilir. Bu sonuçlar geometrik olarak aşağıdaki Mathematica komutunun işletilmesiyle görülebilir:

In[4]:= **LinearPhasePlot2D[** $\begin{pmatrix} 3 & -2 \\ 5 & -4 \end{pmatrix}$**, ShowField->True]**

Out[4]:=

Saddle



Grafik çıktısında orijinden geçen asimptotları kırmızı doğrular temsil etmektedir.

c) Özdeğerlerin kompleks olma durumu: Kompleks özdeğerler, birbirinin eşleniği olduğundan

$\lambda_1 = \alpha + i\beta$ ise $\lambda_2 = \alpha - i\beta$ için

$$x(t) = \mathrm{Re}\left\{cve^{(\alpha+i\beta)}\right\} \qquad (2.14)$$

ifadesini bileşenlerine ayırarak genel çözüm

$$x(t) = e^{\alpha t}\,\mathrm{Re}\left\{cr_1 e^{i\beta}\right\}$$
$$y(t) = e^{\alpha t}\,\mathrm{Re}\left\{cs_1 e^{(\alpha+i\beta)}\right\} \qquad (2.15)$$

olarak elde edilir. c, s_1 ve r_1 değerleri kutupsal formda yazılıp

$$c = |c|e^{i\gamma},\ r_1 = |r_1|e^{i\rho},\ s_1 = |s_1|e^{i\sigma} \qquad (2.16)$$

bu ifadelerle $\alpha = 0$ alınarak

$$x(t) = |c||r_1|\cos(\beta t + \gamma + \rho)$$
$$y(t) = |c||s_1|\cos(\beta t + \gamma + \sigma) \qquad (2.17)$$

elde edilir. Faz düzleminin değişken bir $(x(t), y(t))$ noktasının hareketi, x ve y doğrultusunda eşit dairesel frekans olan β'nın iki basit bileşeninden oluşur ancak bunlar farklı faz ve genliğe sahiptirler. Bu yüzden, faz yolları, genellikle eksenlerle eğim açısı sabit olan ve geometrik olarak elipsler ailesine benzeyen bir form oluştururlar. Dönme yönü iki şekilde de olabilir, bu yapılara "centre" denir.
Orijinde centre olma durumu $p = 0$, $q > 0$ ile sağlanır.
$\alpha \neq 0$ durumunda: Denge noktası, spiral ya da focus olarak adlandırılır, yönü, saat yönüyle aynı ya da ters yönlü olabilir.
$\alpha < 0$ için içe doğru kapanan (kararlı),
$\alpha > 0$ için dışa doğru açılan (kararsız) spiraller oluşur.

Kararlı spiral için
$$\Delta = p^2 - 4q > 0,\ q > 0,\ p < 0; \qquad (2.18)$$

Kararsız spiral için
$$\Delta = p^2 - 4q > 0,\ q > 0,\ p > 0 \qquad (2.19)$$

koşulları sağlanmalıdır [2]. Bütün bu durumları **Örnek 2.2**'deki sistemin **In[4]**'deki katsayılarını değiştirerek gözlemleyebiliriz.

3. DOĞRUSAL OLMAYAN DİNAMİK SİSTEMLER VE MATHEMATICA UYGULAMALARI

Centre, kararlı ve kararsız spiraller arasında kaldığından dejenere durum olarak görülebilir. Centre'nin varlığı, sistemdeki a ve d katsayılarının $a + d = 0$ koşulunu sağlamasına bağlı olduğundan centre, daha hassas bir yapıya sahiptir. Bu yüzden, doğrusal olmayan bir sisteme doğrusal yaklaşımla centre'nin varlığı kestirilebiliyorsa, bu orijinal sistemin kesinlikle bir centre yapısına sahip olacağı anlamına gelmez. Kararlı ya da kararsız spirallerle karşılaşılabilir. Bu, bütün dejenere durumlar için geçerlidir. Yani doğrusallaştırma yaklaşımı her zaman için çok güvenli bir yöntem değildir.

Eğer bir denge noktasının bir komşuluğunda başlayan her faz yolu sonunda denge noktasına yakınsıyorsa, bu noktaya *çeker* denir. (bu terim doğrusal ve doğrusal olmayan sistemlerin her ikisi için de kullanılır). Kararlı node ve kararlı spiral birer çekerdir. Çekicinin yönü ters çevrildiğinde buna *iter* denir. Kararsız node'lar ve kararsız spiraller iterdir ancak eyer noktası iter değildir.

Doğrusallaştırılmış denklemin özdeğerlerinin reel kısmı 0'dan farklı ise denge noktası hiperboliktir. Hiperbolik noktalarda doğrusal olmayan denklemle doğrusallaştırılmış denklemin faz diyagramları aynıdır. Spiraller, node'lar, ve eyer noktaları hiperboliktir ancak centre değildir.

Aşağıdaki modül (Mathematica içinde kendi yarattığınız komut dizgesi) faz yörüngesinin kesiksiz grafiğini verir. Köşeli parantez içindeki "**f**" vektör alanını, "**ic**" başlangıç koşullarını, "**tmax**" ise sistemin çözümü hesaplanırken belirlenen maksimum zamanı, "**opts**" ise grafik çiziminin "**Show**" komutu ile gösterilirken kullanılacak opsiyonel durumları belirtmektedir:

In[5]:=

```
PhasePlot[f_,ic_,tmax_,opts___]:=Block[{i,n=Length[ic],ff,ivp,sol,ph
aseplot},
      ff=f /. {x→x[t],y→y[t]};
      Off[ParametricPlot::"ppcom"];
      Do[

ivp={x'[t]□ff[[1]],y'[t]□ff[[2]],x[0]□ic[[i,1]],y[0]□ic[[i,2]]};
        sol=NDSolve[ivp,{x[t],y[t]},{t,-tmax,tmax}];
        phaseplot[i]=ParametricPlot[{x[t],y[t]} /. sol,{t,-
tmax,tmax},DisplayFunction→Identity]
        ,{i,1,n}];
      On[ParametricPlot::"ppcom"];

Show[Table[phaseplot[i],{i,1,n}],DisplayFunction→$DisplayFunction,o
pts]
```

Aşağıdaki modül ise sistemin vektör alanının doğrusallaştırılması sonucunda bütün tekil noktalarını bulur.

In[6]:=

Örnek 3.1

$$\begin{aligned} \dot{x} &= -y + ax\left(x^2 + y^2\right) \\ \dot{y} &= x + ay\left(x^2 + y^2\right) \end{aligned} \qquad (3.1)$$

Sisteminin faz diyagramını inceleyelim. Sistemin faz diyagramı ilk bakışta centre oluşturuyor gibi görünse de sonuç a'nın alacağı değerlere bağlıdır. Sistemin bir kritik noktası $(0,0)$'dır. Bu noktada doğrusallaştırma sonucunda elde edilen katsayılar matrisi

$$A=\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix} \tag{3.2}$$

dir. Bu matrisin çözümüne bağlı olarak faz diyagramının yapısının centre olduğu söylenebilir. Bu doğrusal olmayan sisitemin analizi için sistemi kutupsal formada yazalım:

$$x = r\cos\theta$$
$$y = r\sin\theta \tag{3.3}$$

dönüşümüyle

$$\dot{r} = ar^3$$

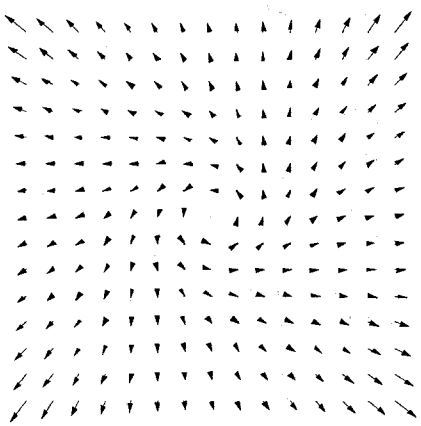
$$\dot{\theta} = \frac{x\dot{y} - y\dot{x}}{r^2} \text{ ile } \dot{\theta} = 1 \tag{3.4}$$

bulunur.

Bu da gösteriyor ki açı ve uzaklık birbirinden bağımsızdır ve bütün trajektörler orijin çevresinde sabit hızla hareket eder. $a > 0$ durumunda: $t \to \infty$ için $r(t) \to \infty$ kararsız spiral, $a < 0$ durumunda $t \to \infty$ için $r(t) \to 0$ kararlı spiral $a = 0$ için centre yapısı oluşur. Bunu sırasıyla a değişkenine 1, 0 ve -1 değerlerini vererek inceleyelim. İlk olarak $a = 1$ olsun.

In[1]:= **<<Graphics`PlotField`**

In[2]:= **PlotVectorField[{-y+x(x^2+y^2),x+y(x^2+y^2)},{x,-2,2},{y,-2,2}]**

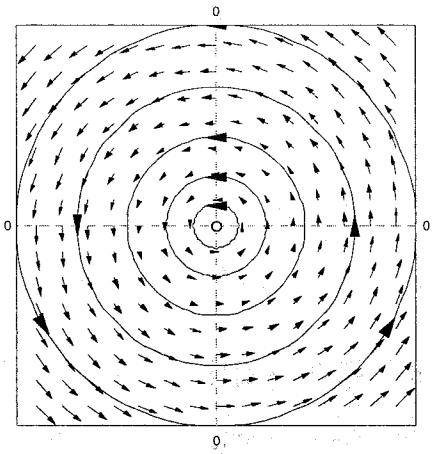


Out[2]:= -Graphics-

a=0 aldığımızda centre yapısını görürüz.

In[3]:= **LinearPhasePlot2D[** $\begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}$**, ShowField->True]**

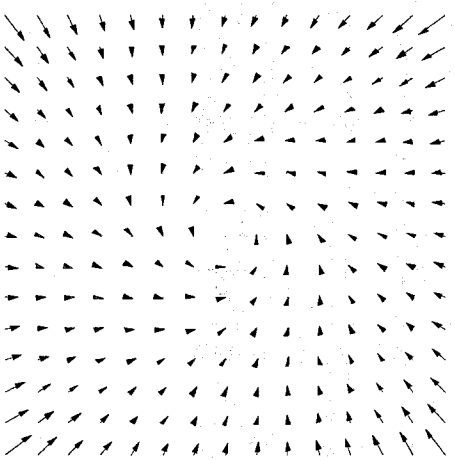
Center (periodic orbits)



Out[3]:= □Graphics□

Son olarak a = -1 içeri doğru kapanan spiral yapısını oluşturur.

In[4]:=**PlotVectorField[{-y-x(x^2+y^2),x-y(x^2+y^2)},{x,-2,2},{y,-2,2}]**



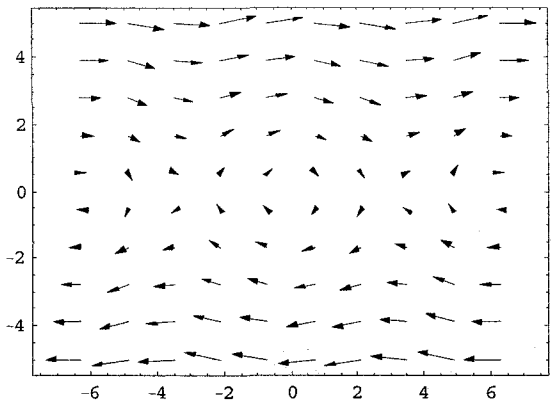
Out[4]:= □Graphics□

Örnek 3.2 **Sarkaç problemini ele alalım**

$\dot{x} = y$

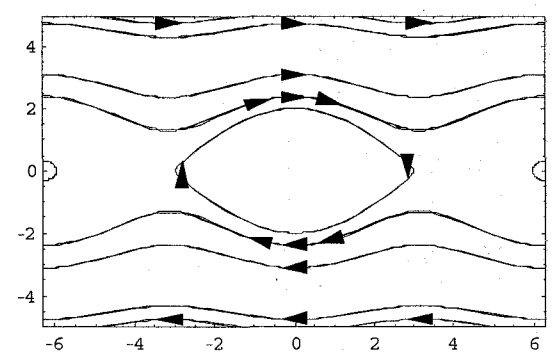
$\dot{y} = -\sin x$

In[1]:= **PlotVectorField[{y,-Sin[x]},{x,-2π,2π},{y,-5,5},Frame→True,PlotPoints→10];**



Out[1]:= Graphics

Bu peryodik yörüngeyi aşağıdaki modülü kullanırsak daha belirgin görebiliriz:
İşlem için tekrar **"CurvesGraphics`"** paketinden **"PhasePlot"** komutunu kullanıyoruz.
In[2]:= **PhasePlot[{y,-Sin[x]},{x,-2π,2π},{y,-5,5},{-6,6},Frame→True]**



Out[2]:= Graphics

4. GARİP ÇEKERLER

Çeker, bütün komşu trajektörlerin yakınsadığı kümedir. Bir A kümesinin çeker olabilmesi için aşağıdaki koşulları sağlaması gerekir:

A, invaryant bir kümedir.
A, başlangıç koşullarının açık bir kümesini çeker.
A, minimaldir.

Başlangıç koşullarına aşırı hassasiyet gösteren çekerlere "*garip çeker*" deriz. İsminin garip çeker olmasının nedeni, fraktal yapılarından kaynaklanır. Çekerin hareketi, başlangıç değerlerine aşırı hassasiyet gösterir. Bunun anlamı, birbirine çok yakın harekete başlayan iki trajektör, çok hızlı bir şekilde birbirinden uzaklaşır ve sonuçta birbirinden çok farklı davranış sergilerler. (1) nolu Lorenz denklemi buna örnektir. Bu durum, Matematica kodlarındaki başlangıç koşulları değiştirilerek gözlemlenebilir.

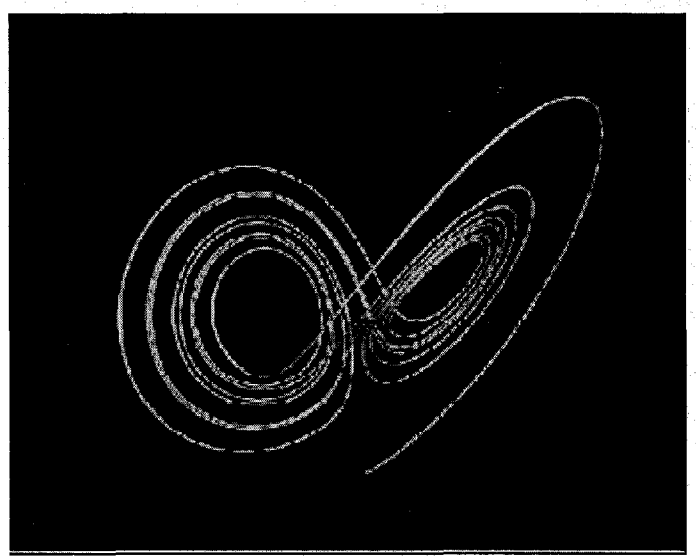
Birinci bölümdeki **<< RealTime3D`** paketi, oluşturulan üç boyutlu cismin döndürülmesini sağlar. Mouse şekil üzerindeyken sola basılı tutulup kaydırıldığında cisim, istenen yönde

hareket kazanacaktır. Böylelikle Lorenz kelebeğinin nasıl bir form oluşturduğu daha açık gözlenebilir. Ayrıca

In[3]:= **Get["MathGL3d`OpenGLViewer`"]**

paket programı yüklendiği takdirde Lorenz kelebeği, siyah zemin üzerinde hareketli olarak aşağıdaki komutla elde edilebilir:

In[4]:= **Table[MVShow3D[Graphics3D[{{Table[Line[{s1[[i]],s1[[i+1]]}],{i,1,t}]}}],PlotRange->{{-60,60},{-60,60},{-60,60}},AspectRatio-> Automatic,MVLineTubeSize->0.1,MVPolygonShading->MVSmooth,MVTubeSegments->12,MVTubeAngle->30,MVPointSphereSize->10,MVNewScene->True],{t,0.1,L-1,20}];**



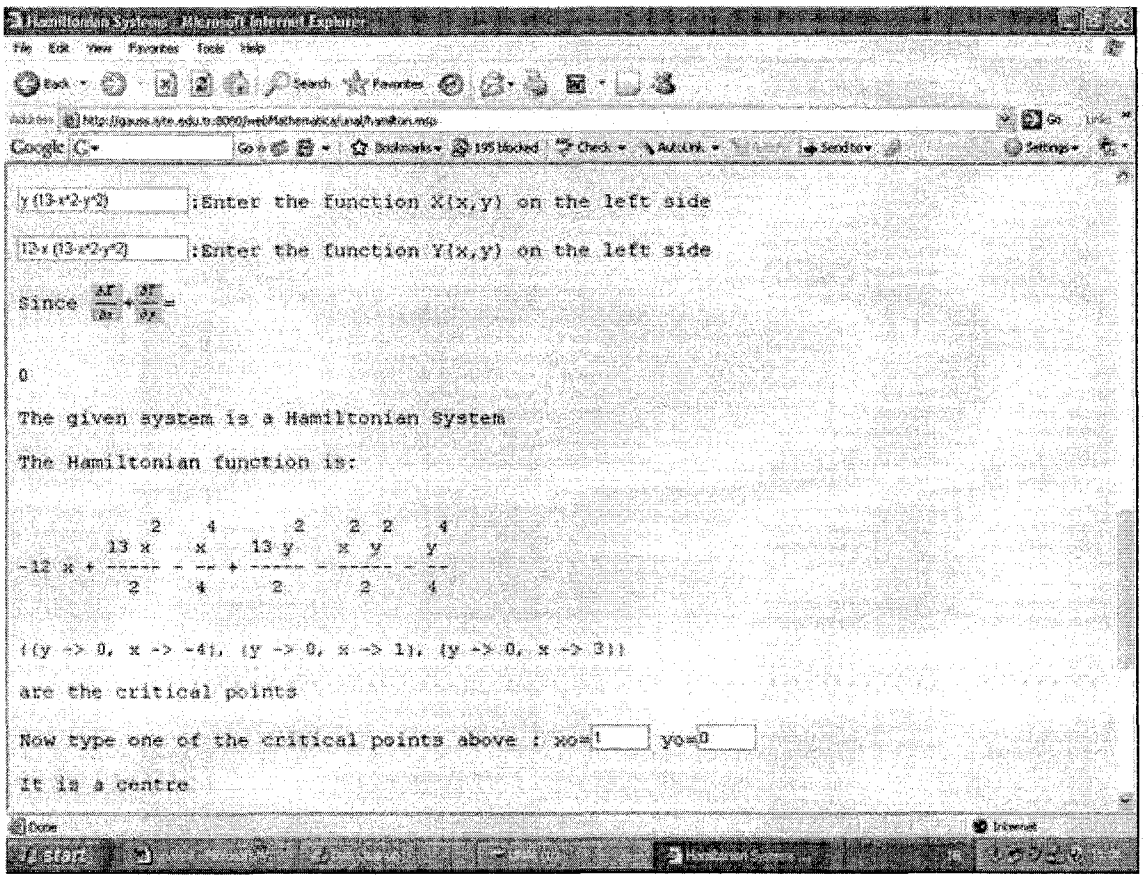
5. HAMİLTON SİSTEMLERİNİN WEB- MATHEMATICA UYGULAMALARI

Bu bölüme kadar dinamik sistemlerin genel kavramlarını Mathematica yazılımı ile vermeye çalıştık. Mathematica sonuçta bir yazılım olduğu için programı bilenler ancak bu uygulamaları yapabilir. Wolfram yazılım gurubu 2000 yılında Matematica'nın internet ortamında kullanımını sağlayan web-Mathematica programını geliştirdi. MSP teknolojisi webMathematica'nın temelidir. MSP servletler standart Java teknolojisine dayanır. Servletler bir web sunucusunda çalışan özel Java programlarıdır. Tipik olarak destek, ayrı bir program olan web sunucusuna bağlanan "servlet motoru" tarafından sağlanır. Aslında bütün gelişmiş web sunucuları servletleri doğal olarak destekler. Bu Apache, Microsoft'un IIS ve PWS, Netscape Enterprise Server, iPlanet, ve uygulama sunucularını (IBM WebSphere gibi) da kapsar. MSP teknolojisi HTML sayfalarından oluşan bir sitenin Mathematica komutlarını da içermesine izin verir. Bu sayfaların herhangi birinden bu komutların çalışmasını gerektiren bir talep geldiğinde (bunlara MSP scriptleri denir) Mathematica komutları işletilir ve hesaplanan sonuçlar sayfada gösterilir.

Kullanıcı, **http://gauss.iyte.edu.tr:8080/webMathematica/unal/Hamilton.msp** adresinden

$$\dot{x} = X(x, y)$$
$$\dot{y} = Y(x, y) \qquad (5.1)$$

formunda bir dinamik sistemini online girdiğinde, sistem server üzerinden Mathematica Kernel'ini kullanarak sistemin Hamiltonian olup olmadığını belirtip, Hamiltonian ise Hamilton fonksiyonunu bulup, sitemin kritik noklarını sınıflandırıp, çözümün faz yörüngesini vermektedir. [3]



KAYNAKLAR

[1] Jordan, D.W, (1999), Nonlinear ordinary differential equations;an introduction to dynamical Systems, Oxford University Press, Oxford

[2] Strogatz, Steven.H. (1994), Nonlinear Dynamics And Chaos (With Aplications to Physics, Biology, Chemistry and Engineering), Addison-Wesley Pub

[3] Ufuktepe,Ü., (2003) An Application with webMathematica, Lecture Notes in Computer Science, 2675, Sayfa 774-780, Springer-Verlag