9D_Lorenz_Chaotic_systemsJulia

May 22, 2019

0.1 Julia programming language

In [1]:
```
using LinearAlgebra, Plots
```

In [2]:
```
# Number of points in each subinterval
N=23
# width of the subintervals
p=0.01
h=p/(N-1);
```

In [3]:
```
function Matrices(N)

    A=zeros(N-1, N-1)

    for i=1:N-1
        for j=1:N-1
            if j==i-1 || j==i+1
                A[i,j]=1.0/3
            elseif j==i
                A[i,j]=1.0
            end
        end
    end


    B=zeros(N-1,N-1)

    B[1,1]=-17.0/12
    B[1,2]=83.0/36
    B[1,3]=-11/9
    B[1,4]=2.0/3
    B[1,5]=-37.0/180
    B[1,6]=1.0/36

    B[2,1]=-7.0/9
    B[2,3]=7.0/9
    B[2,4]=1.0/36
```

```
        B[N-2,N-7]=1.0/36
        B[N-2,N-6]=-7.0/36
        B[N-2,N-5]=7.0/12
        B[N-2,N-4]=-1.0
        B[N-2,N-3]=7.0/36
        B[N-2,N-2]=-7.0/12
        B[N-2,N-1]=35.0/36

        B[N-1,N-7]=7.0/45
        B[N-1,N-6]=-67.0/60
        B[N-1,N-5]=125.0/36
        B[N-1,N-4]=-55.0/9
        B[N-1,N-3]=20.0/3
        B[N-1,N-2]=-1003.0/180
        B[N-1,N-1]=451.0/180


        for i=3:N-3
            for j=1:N-1
                if j==i-2
                    B[i,j]=-1.0/36
                elseif j == i-1
                    B[i,j]=-7.0/9
                elseif j == i+1
                    B[i,j]=7.0/9
                elseif j==i+2
                    B[i,j]=1.0/36
                end
            end
        end
        B=(1/h)*B
        return [A,B]
    end
```

```
Out[3]: Matrices (generic function with 1 method)
```

```
In [4]: a=0.5; b1=4.0*(1.0+a^2)/(1.0+2.0*a^2); b2=(1.0+2.0*a^2)/(2.0*(1.0+a^2)); b3
        b4=(a^2)/(1.0+a^2); b5=(8.0*a^2)/(1.0+2.0*a^2); b6=4.0/(1.0+2.0*a^2); r=14.

        function f1(U)
            return U[2].*U[4]-b4*(U[4].^2)-b3*U[3].*U[5]
        end

        function f2(U)
            return -U[1].*U[4]+U[2].*U[5]-U[4].*U[5]
        end

        function f3(U)
```

```
    return -U[2].*U[4]+b4*(U[2].^2)+b3*U[1].*U[5]
end


function f4(U)
    return U[2].*U[3]+U[2].*U[5]-U[4].*U[5]
end

function f5(U)
    return -0.5*(U[2].^2)+0.5*(U[4].^2)
end

function f6(U)
    return -U[2].*U[9]+U[4].*U[9]
end

function f7(U)
    return -2.0*U[5].*U[8]+U[4].*U[9]
end

function f8(U)
    return 2.0*U[5].*U[7]-U[2].*U[9]
end

function f9(U)
    return 2.0*U[2].*U[6]-2.0*U[4].*U[6]-U[4].*U[7]+U[2].*U[8]
end

function globf(U,f,m)
    return f[m](U)
end

f=[f1,f2,f3,f4,f5,f6,f7,f8,f9]

alpha11=σ*b1; alpha12=0.0; alpha13=0.0; alpha14=0.0; alpha15=0.0; alpha16=0
alpha18=0.0;alpha19=0.0;

alpha21=0.0; alpha22=σ; alpha23=0.0; alpha24=0.0; alpha25=0.0; alpha26=0.0;
alpha28=0.0;alpha29=0.5*σ;

alpha31=0.0; alpha32=0.0; alpha33=σ*b1; alpha34=0.0; alpha35=0.0; alpha36=0
alpha38=-σ*b2;alpha39=0.0;

alpha41=0.0; alpha42=0.0; alpha43=0.0; alpha44=σ; alpha45=0.0; alpha46=0.0;
alpha48=0.0;alpha49=-0.5*σ;

alpha51=0.0; alpha52=0.0; alpha53=0.0; alpha54=0.0; alpha55=σ*b5; alpha56=0
alpha58=0.0;alpha59=0.0
```

```
alpha61=0.0; alpha62=0.0; alpha63=0.0; alpha64=0.0; alpha65=0.0; alpha66=b6
alpha68=0.0;alpha69=0.0

alpha71=r; alpha72=0.0; alpha73=0.0; alpha74=0.0; alpha75=0.0; alpha76=0.0;
alpha78=0.0;alpha79=0.0

alpha81=0.0; alpha82=0.0; alpha83=-r; alpha84=0.0; alpha85=0.0; alpha86=0.0
alpha88=b1;alpha89=0.0

alpha91=0.0; alpha92=r; alpha93=0.0; alpha94=-r; alpha95=0.0; alpha96=0.0;
alpha98=0.0;alpha99=1.0


Alphas=[[alpha11,alpha12,alpha13,alpha14,alpha15,alpha16,alpha17,alpha18,al
    [alpha21,alpha22,alpha23,alpha24,alpha25,alpha26,alpha27,alpha28,alpha2
    [alpha31,alpha32,alpha33,alpha34,alpha35,alpha36,alpha37,alpha38,alpha3
    [alpha41,alpha42,alpha43,alpha44,alpha45,alpha46,alpha47,alpha48,alpha4
    [alpha51,alpha52,alpha53,alpha54,alpha55,alpha56,alpha57,alpha58,alpha5
    [alpha61,alpha62,alpha63,alpha64,alpha65,alpha66,alpha67,alpha68,alpha6
    [alpha71,alpha72,alpha73,alpha74,alpha75,alpha76,alpha77,alpha78,alpha7
    [alpha81,alpha82,alpha83,alpha84,alpha85,alpha86,alpha87,alpha88,alpha8
    [alpha91,alpha92,alpha93,alpha94,alpha95,alpha96,alpha97,alpha98,alpha9
]

function alphaU(n, U)
    Sum=zeros(N-1)
    for i=1:length(U)
        if i!=n
            Sum += Alphas[n][i]*U[i]
        end
    end
    return Sum
end

g1=0.0
g2=0.0
g3=0.0
g4=0.0
g5=0.0
g6=0.0
g7=0.0
g8=0.0
g9=0.0

Gr=[g1,g2,g3,g4,g5,g6,g7,g8,g9]

#inverse of A
```

```
inv_A=inv(Matrices(N)[1])

# Matrix E
E=inv_A*(Matrices(N)[2])


# Matrices to be used


E_inv=[inv(E+Alphas[i][i]*Matrix{Float64}(I,N-1,N-1)) for i=1:length(Gr)];
```

In [ ]:

In [5]:
```
# Function to solve the problem on a subdomain [lower_bound, lower_bound+0.

function Solver(f, IC, Max_iter)

    #Vectors Ki
    K=[zeros(N-1) for k=1:length(f)]
    for j=1:length(f)
        K[j][1]=(-7.0/(45*h))*IC[j]
        K[j][2]=(-1.0/(36*h))*IC[j]
    end

    # Vectors Hi
    H=[inv_A*K[k] for k=1:length(f)]

    #Gr
    G=[Gr[k]*ones(N-1) for k=1:length(f)]

    #Approximating Ur
    Ur=[IC[k]*ones(N-1) for k=1:length(f)]

    # Fi
    F=[[] for k=1:length(f)]

    for j=1:Max_iter
        for k=1:length(f)
            F[k]=globf(Ur,f,k)
            Ur[k]=E_inv[k]*(G[k]-H[k]-alphaU(k, Ur)-F[k])
        end
    end
    #derivative approximation
    Ur_p=[E*(Ur[z])+H[z] for z=1:length(f)]
    return [Ur, Ur_p]
end
```

Out[5]: Solver (generic function with 1 method)

In [9]:
```
r=14.1

alpha11=σ*b1; alpha12=0.0; alpha13=0.0; alpha14=0.0; alpha15=0.0; alpha16=0
alpha18=0.0;alpha19=0.0;

alpha21=0.0; alpha22=σ; alpha23=0.0; alpha24=0.0; alpha25=0.0; alpha26=0.0;
alpha28=0.0;alpha29=0.5*σ;

alpha31=0.0; alpha32=0.0; alpha33=σ*b1; alpha34=0.0; alpha35=0.0; alpha36=0
alpha38=-σ*b2;alpha39=0.0;

alpha41=0.0; alpha42=0.0; alpha43=0.0; alpha44=σ; alpha45=0.0; alpha46=0.0;
alpha48=0.0;alpha49=-0.5*σ;

alpha51=0.0; alpha52=0.0; alpha53=0.0; alpha54=0.0; alpha55=σ*b5; alpha56=0
alpha58=0.0;alpha59=0.0

alpha61=0.0; alpha62=0.0; alpha63=0.0; alpha64=0.0; alpha65=0.0; alpha66=b6
alpha68=0.0;alpha69=0.0

alpha71=r; alpha72=0.0; alpha73=0.0; alpha74=0.0; alpha75=0.0; alpha76=0.0;
alpha78=0.0;alpha79=0.0

alpha81=0.0; alpha82=0.0; alpha83=-r; alpha84=0.0; alpha85=0.0; alpha86=0.0
alpha88=b1;alpha89=0.0

alpha91=0.0; alpha92=r; alpha93=0.0; alpha94=-r; alpha95=0.0; alpha96=0.0;
alpha98=0.0;alpha99=1.0


Alphas=[[alpha11,alpha12,alpha13,alpha14,alpha15,alpha16,alpha17,alpha18,al
    [alpha21,alpha22,alpha23,alpha24,alpha25,alpha26,alpha27,alpha28,alpha2
    [alpha31,alpha32,alpha33,alpha34,alpha35,alpha36,alpha37,alpha38,alpha3
    [alpha41,alpha42,alpha43,alpha44,alpha45,alpha46,alpha47,alpha48,alpha4
    [alpha51,alpha52,alpha53,alpha54,alpha55,alpha56,alpha57,alpha58,alpha5
    [alpha61,alpha62,alpha63,alpha64,alpha65,alpha66,alpha67,alpha68,alpha6
    [alpha71,alpha72,alpha73,alpha74,alpha75,alpha76,alpha77,alpha78,alpha7
    [alpha81,alpha82,alpha83,alpha84,alpha85,alpha86,alpha87,alpha88,alpha8
    [alpha91,alpha92,alpha93,alpha94,alpha95,alpha96,alpha97,alpha98,alpha9
];


# Time interval [I0, I1]
I0=0.0
I1=50.0

function Solution()
    #initial conditions
```

```
        IC=[0.01,0.0,0.01,0.0,0.0,0.0,0.0,0.0,0.01]

        Max_iter=8

        lower_bound=I0

        Sol=[[] for s=1:length(IC)]
        Sol_p=[[] for s=1:length(IC)]
        while lower_bound<I1
            Ur,Ur_p=Solver(f,IC, Max_iter)[1],Solver(f,IC, Max_iter)[2]
            for l=1:length(IC)
                push!(Sol[l],Ur[l]...)
                push!(Sol_p[l],Ur_p[l]...)
            end
            for r=1:length(IC)
                IC[r]=Sol[r][end]
            end
            lower_bound += p
        end
        return [Sol,Sol_p]
    end
```

```
Out[9]: Solution (generic function with 1 method)
```

```
In [10]: Soltotal=@time Solution()
         Sol, Sol_p=Soltotal[1], Soltotal[2];
```

```
 17.465644 seconds (83.80 M allocations: 5.999 GiB, 47.26% gc time)
```

```
In [11]: h1=(I1-I0)/(length(Sol[1])-1)
         tspan=I0:h1:I1
         length(tspan)
```

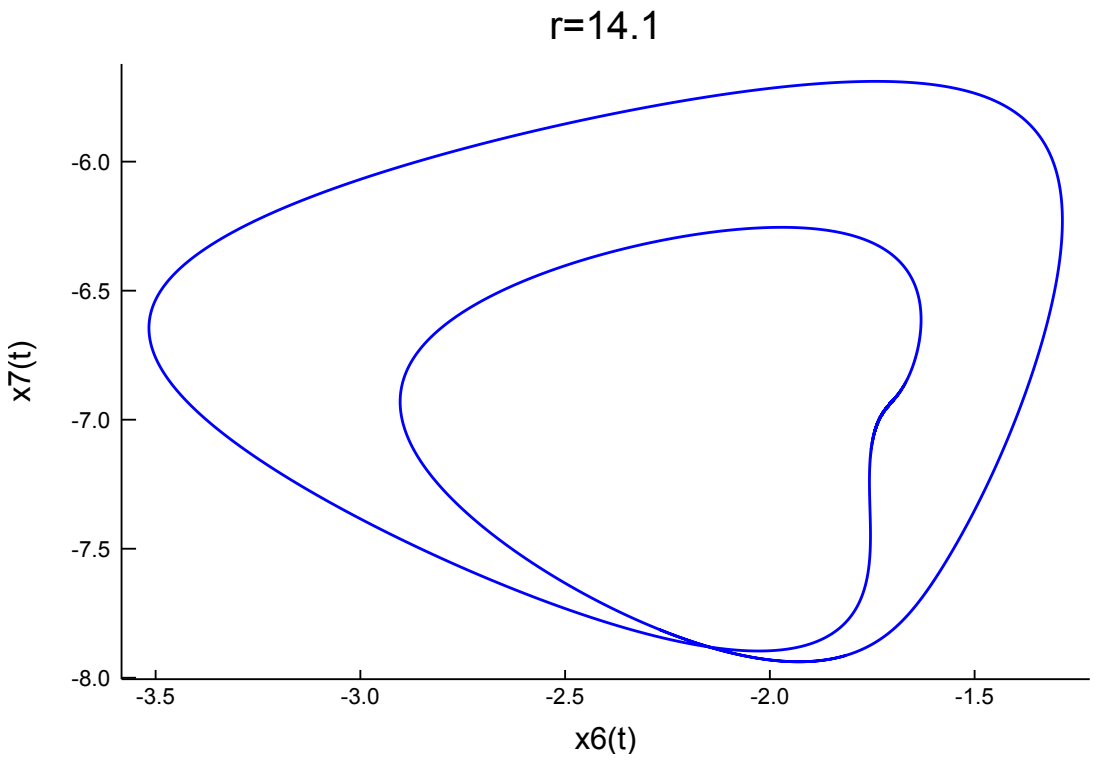
```
Out[11]: 110022
```

```
In [56]: length(Sol[1])
```

```
Out[56]: 110022
```

```
In [8]: plot(Sol[6][end-40000:end],Sol[7][end-40000:end],label="", xlabel="x6(t)",
        #savefig("9dlorpp3.png")
```

```
Out[8]:
```



In [ ]:

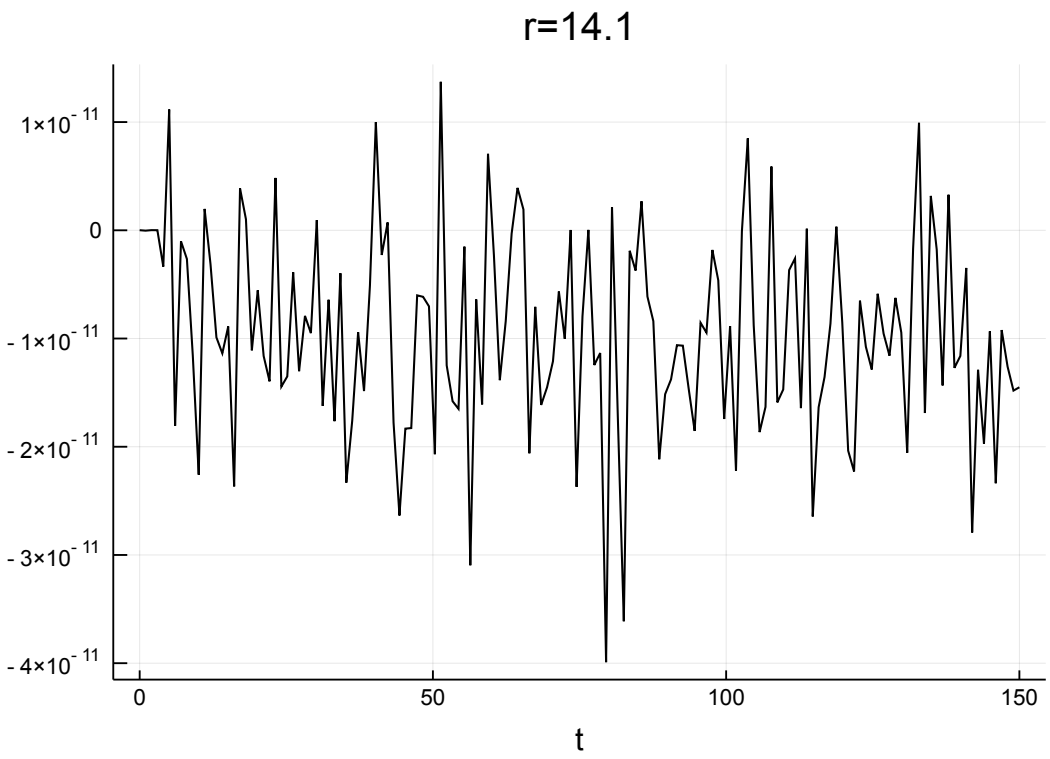
In [12]:
```
# Definition of the system. This function will be used in Runge Kutta meth
function funct(U)
    return [-σ*b1*U[1]-U[2]*U[4]+b4*U[4]^2+b3*U[3]*U[5]-σ*b2*U[7],
        -σ*U[2]+U[1]*U[4]-U[2]*U[5]+U[4]*U[5]-σ*0.5*U[9],
        -σ*b1*U[3]+U[2]*U[4]-b4*U[2]^2-b3*U[1]*U[5]+σ*b2*U[8],
        -σ*U[4]-U[2]*U[3]-U[2]*U[5]+U[4]*U[5]+σ*0.5*U[9],
        -σ*b5*U[5]+0.5*U[2]^2-0.5*U[4]^2,
        -b6*U[6]+U[2]*U[9]-U[4]*U[9],
        -b1*U[7]-r*U[1]+2.0*U[5]*U[8]-U[4]*U[9],
        -b1*U[8]+r*U[3]-2.0*U[5]*U[7]+U[2]*U[9],
        -U[9]-r*U[2]+r*U[4]-2.0*U[2]*U[6]+2.0*U[4]*U[6]+U[4]*U[7]-U[2]*U[8
    ]
    end
```

Out[12]: funct (generic function with 1 method)

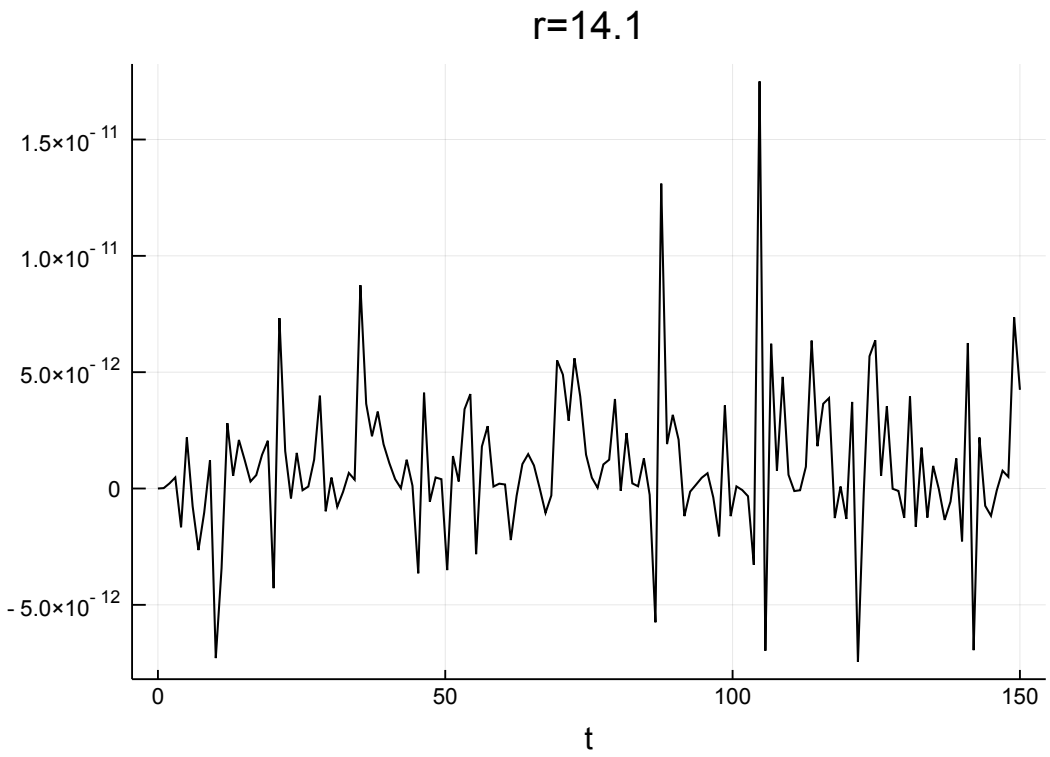
0.1.1 Accuracy

In [44]:
```
function accuracy(component,pos)
    return Sol_p[component][pos]-funct([Sol[i][pos] for i=1:length(Sol)])[
end
```

```
Out[44]: accuracy (generic function with 1 method)

In [26]: Acu=[[accuracy(j,i) for i=N-1:100*N-100:length(Sol[1])] for j=1:length(Sol

In [27]: length(Acu[1])

Out[27]: 150

In [28]: h2=I1/(length(Acu[1])-1)
         points=I0:h2:I1;

In [29]: plot(points, Acu[1], color="black", title="r=$r", label="",xlabel="t", lin
         #savefig("9dhyperres1.png")
         #savefig("9dres1.png")

Out[29]:
```

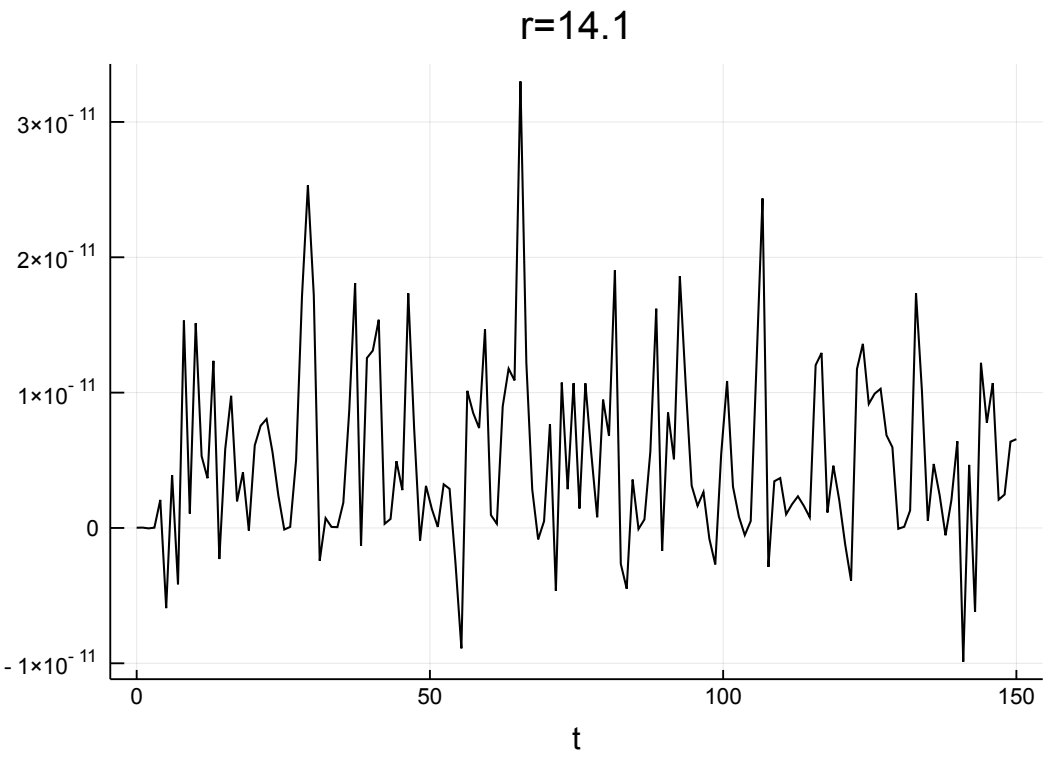



```
In [30]: plot(points, Acu[2], color="black", title="r=$r", label="",xlabel="t",line
         #savefig("9dhyperres2.png")
         #savefig("9dres2.png")

Out[30]:
```



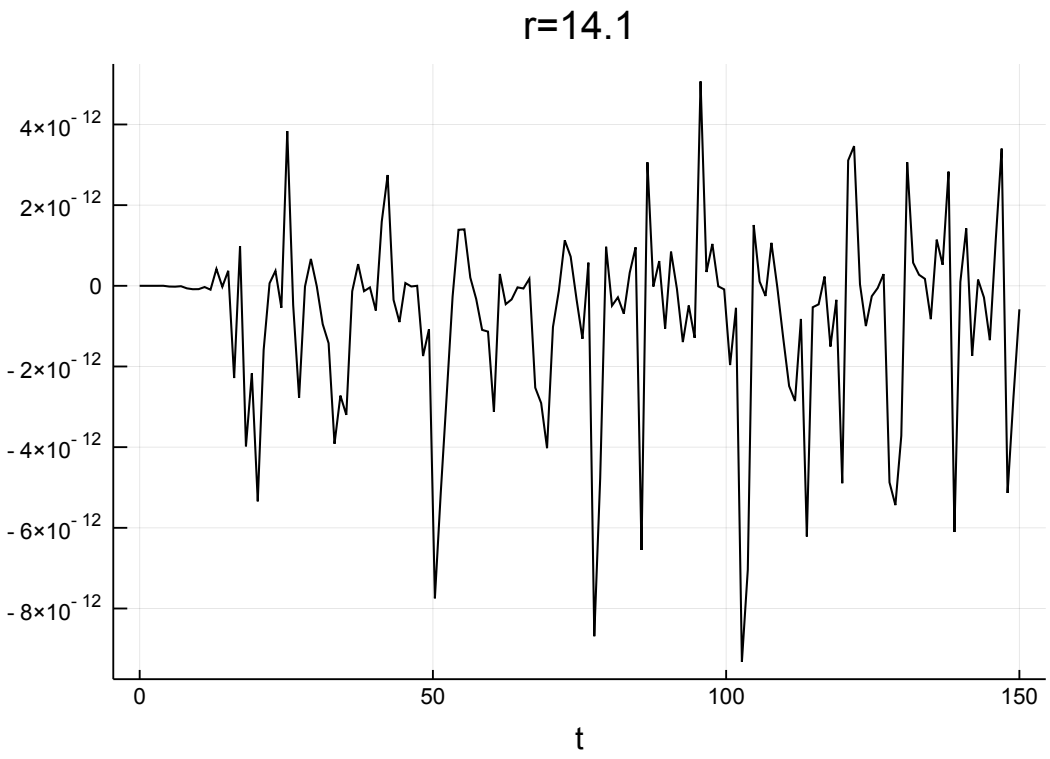
In [31]:
```
plot(points, Acu[3], color="black", title="r=$r", label="",xlabel="t",line
#savefig("9dhyperres3.png")
#savefig("9dres3.png")
```

Out[31]:



In [32]:
```
plot(points, Acu[5], color="black", title="r=$r", label="",xlabel="t",line
#savefig("9dhyperres4.png")
#savefig("9dres4.png")
```

Out[32]:




```
In [33]: [maximum([abs(minimum(Acu[l])),abs(maximum(Acu[l]))]) for l=1:length(Sol)]
```

```
Out[33]: 9-element Array{Float64,1}:
          3.99074107093611e-11
          1.749930408911915e-11
          3.299749362639659e-11
          1.7478907210488615e-11
          9.320599847484345e-12
          3.505629120326148e-11
          2.2057422555121775e-10
          1.9420387520341365e-10
          1.2046896813444619e-10
```

```
In [ ]:
```

0.2 RK4

```
In [13]: # step size
         hrk=0.0001
```

```
Out[13]: 0.0001
```

```
In [14]: # Runge kutta functions
         function K1(U)
```

```
        return hrk*funct(U)
    end

    function K2(U)
        return hrk*funct(U+(1.0/2)*K1(U))
    end

    function K3(U)
        return hrk*funct(U+(1.0/2)*K2(U))
    end

    function K4(U)
        return hrk*funct(U+K3(U))
    end

    # This function is important in the next part
    function indexY(n,m,Y)
        if m<n
            return Y[m][end-1]
        else
            return Y[m][end]
        end
    end
```

```
Out[14]: indexY (generic function with 1 method)
```

```
In [15]: function SolRK4()
             #initial conditions
             IC=[0.01,0.0,0.01,0.0,0.0,0.0,0.0,0.0,0.01]
             lower_bound=I0
             Y=[[IC[i]] for i=1:length(IC)]
             while lower_bound<I1
                 for s=1:length(IC)
                     push!(Y[s],Y[s][end]+(1.0/6)*(K1([indexY(s,j,Y) for j=1:length
                     end
                 lower_bound += hrk
             end
             return Y
         end
```

```
Out[15]: SolRK4 (generic function with 1 method)
```

```
In [16]: Y=@time SolRK4();
```

```
250.587243 seconds (5.23 G allocations: 99.693 GiB, 33.41% gc time)
```

```
In [17]: h2=(I1-I0)/(length(Y[1])-1)
         tspan1=I0:h2:I1
         length(tspan1)
```

```
Out[17]: 500001

In [50]: h2

Out[50]: 0.0001

In [ ]:

In [51]: time=[2.0,4.0,6.0,8.0,10.0, 12.0]
         print("RK4{0.0001}, r=$r:\n\n")
         pos=0
         for l=1:3
             for j=1:length(time)
                 for i=1:length(tspan1)
                     if abs(tspan1[i]-time[j])<0.0001
                         pos=i
                         break
                     end
                 end
                 print("x$l($(time[j]))=$(Y[l][pos])\n")
             end
             print("\n")
         end

RK4{0.0001}, r=14.1:

x1(2.0)=0.00407531087373653
x1(4.0)=0.2780734589674706
x1(6.0)=1.4186621429680124
x1(8.0)=1.4025246633234087
x1(10.0)=1.4634292094914914
x1(12.0)=1.4891823890827345

x2(2.0)=-0.021627282691647348
x2(4.0)=-0.8513541372384609
x2(6.0)=-0.451847346374427
x2(8.0)=-0.5855684763381117
x2(10.0)=-0.5182534349473993
x2(12.0)=-0.49025206616929484

x3(2.0)=0.0037688393200969233
x3(4.0)=-0.27325282401348694
x3(6.0)=-1.410180937224483
x3(8.0)=-1.3841042925340041
x3(10.0)=-1.415285710468242
x3(12.0)=-1.360251361359494
```

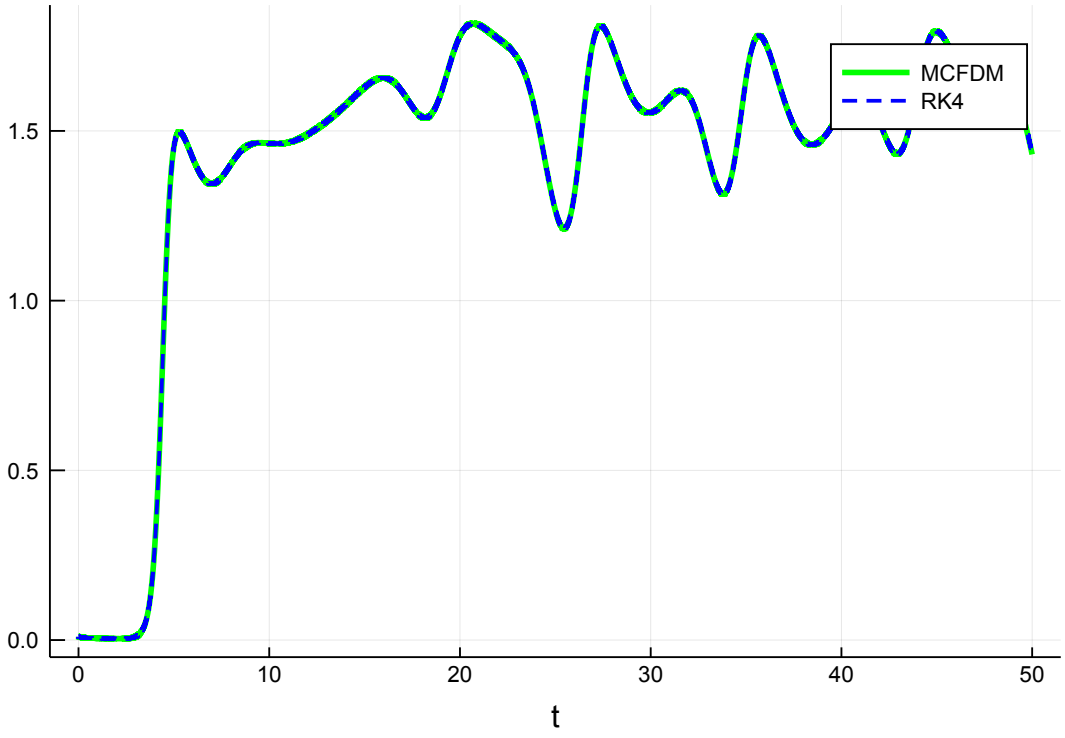
```
In [52]: #time=[2.0, 6.0, 12.0,16.0, 18.0, 25.0]
         print("CFD{p=0.01, N=$N}, r=$r:\n\n")
         pos=0
         for l=1:3
             for j=1:length(time)
                 for i=1:length(tspan)
                     if abs(tspan[i]-time[j])<h1
                         pos=i
                         break
                     end
                 end
                 print("x$l($(time[j]))=$(Sol[l][pos])\n")
             end
             print("\n")
         end
```

```
CFD{p=0.01, N=23}, r=14.1:

x1(2.0)=0.004075259485839986
x1(4.0)=0.27807345897550295
x1(6.0)=1.41864858369797
x1(8.0)=1.4025330906285909
x1(10.0)=1.4634287078632287
x1(12.0)=1.4891856588745613

x2(2.0)=-0.021631429384174436
x2(4.0)=-0.8513541372498983
x2(6.0)=-0.45184413153885084
x2(8.0)=-0.5855688312855962
x2(10.0)=-0.5182516576166014
x2(12.0)=-0.49024931139437056

x3(2.0)=0.0037686702704846126
x3(4.0)=-0.27325282402148926
x3(6.0)=-1.4101672310477285
x3(8.0)=-1.3841117496072473
x3(10.0)=-1.4152829800495117
x3(12.0)=-1.3602480637787935
```

```
In [53]: tspan[1]

Out[53]: 0.0
```

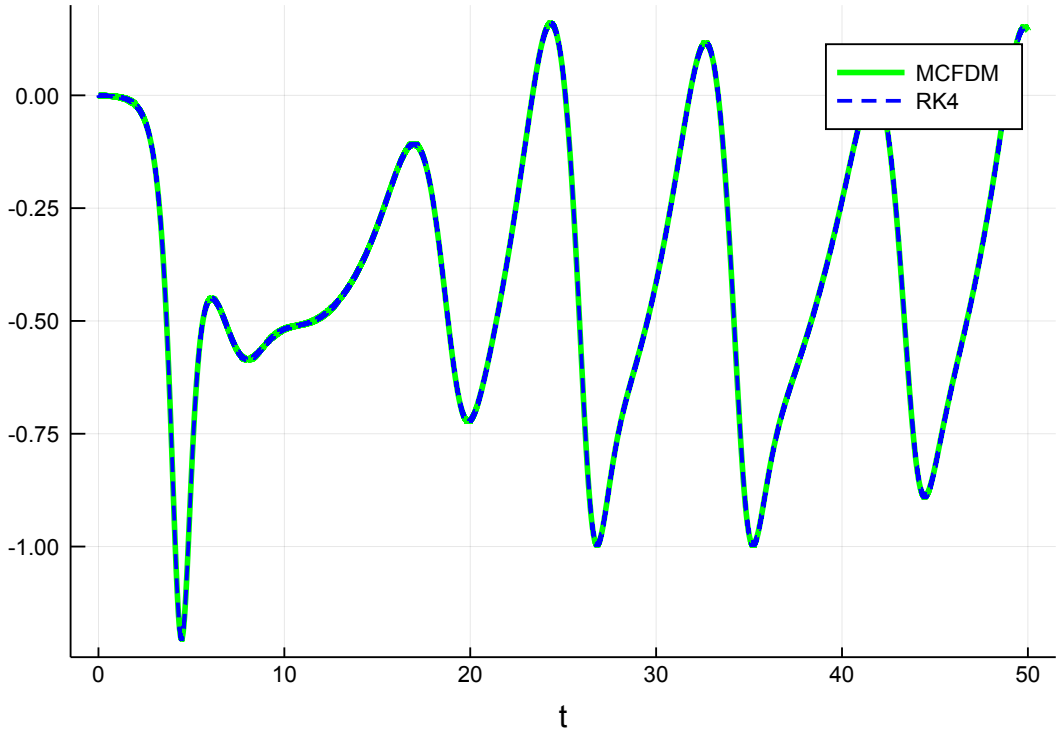
```
In [18]: plot(tspan, Sol[1], label="MCFDM", xlabel="t",color="lime", linewidth=3.0)
         plot!(tspan1, Y[1], label="RK4", xlabel="t", linestyle=:dash,color="blue",
         #savefig("9dlor1.png")
```

Out[18]:



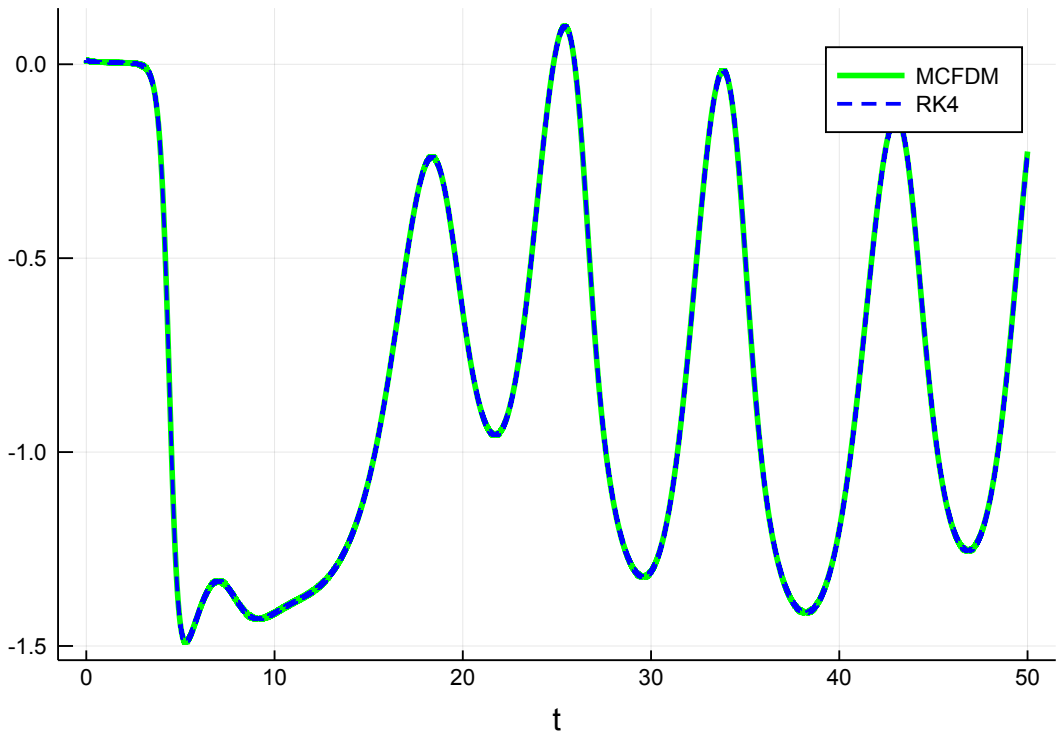
In [19]:
```
plot(tspan, Sol[2], color="lime", label="MCFDM", xlabel="t",linewidth=3.0)
plot!(tspan1, Y[2], color="blue", label="RK4", xlabel="t", linestyle=:dash
#savefig("9dlor2.png")
```

Out[19]:



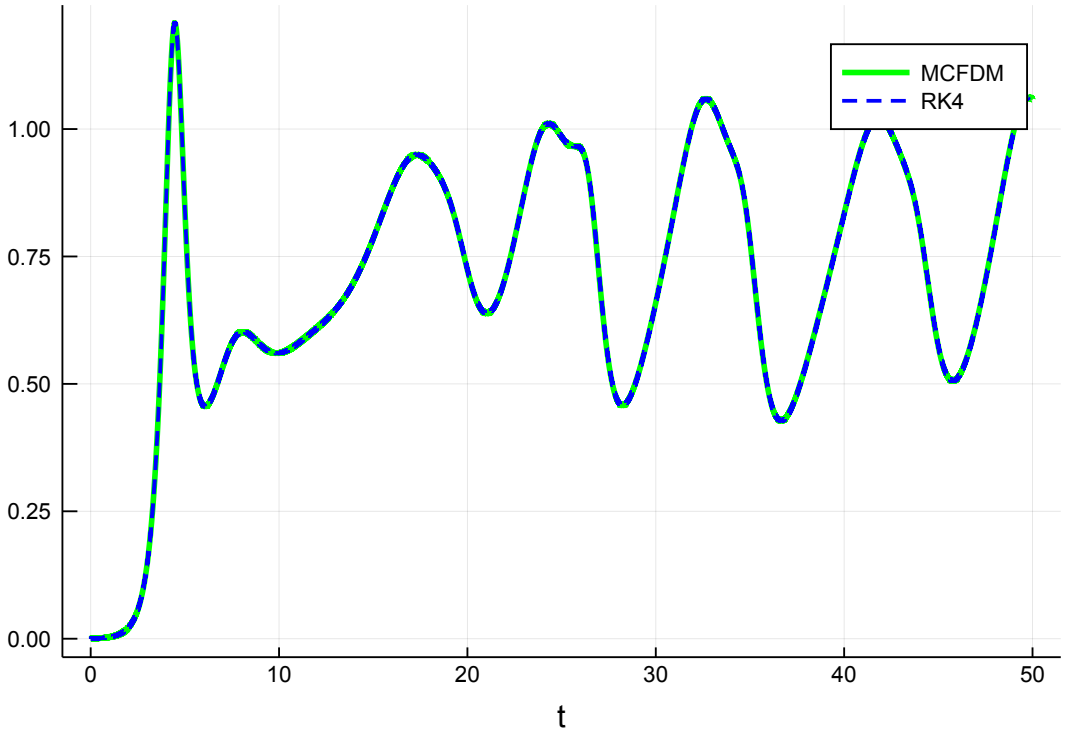
In [20]:
```
plot(tspan, Sol[3], label="MCFDM", xlabel="t",color="lime",linewidth=3.0)
plot!(tspan1, Y[3], label="RK4", xlabel="t", linestyle=:dash,color="blue",
#savefig("9dor3.png")
```

Out[20]:



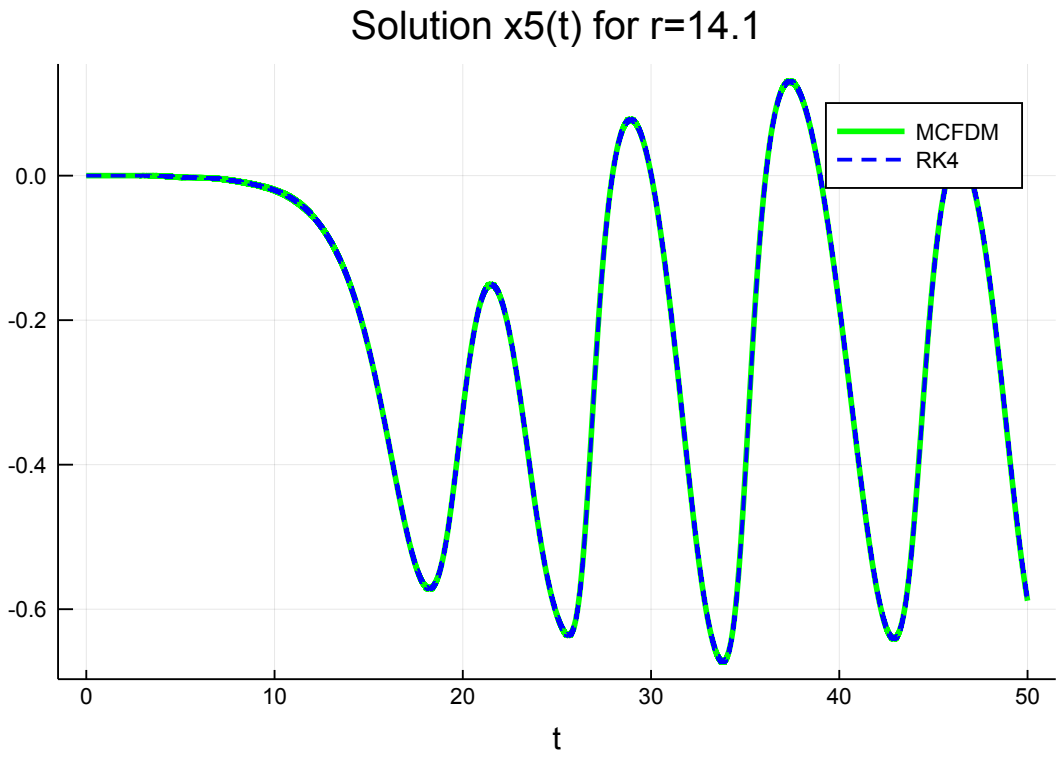
In [21]:
```
plot(tspan, Sol[4], color="lime", label="MCFDM", xlabel="t",linewidth=3.0)
plot!(tspan1, Y[4], color="blue", label="RK4", xlabel="t", linewidth=2.0,l
#savefig("9dlor4.png")
```

Out[21]:



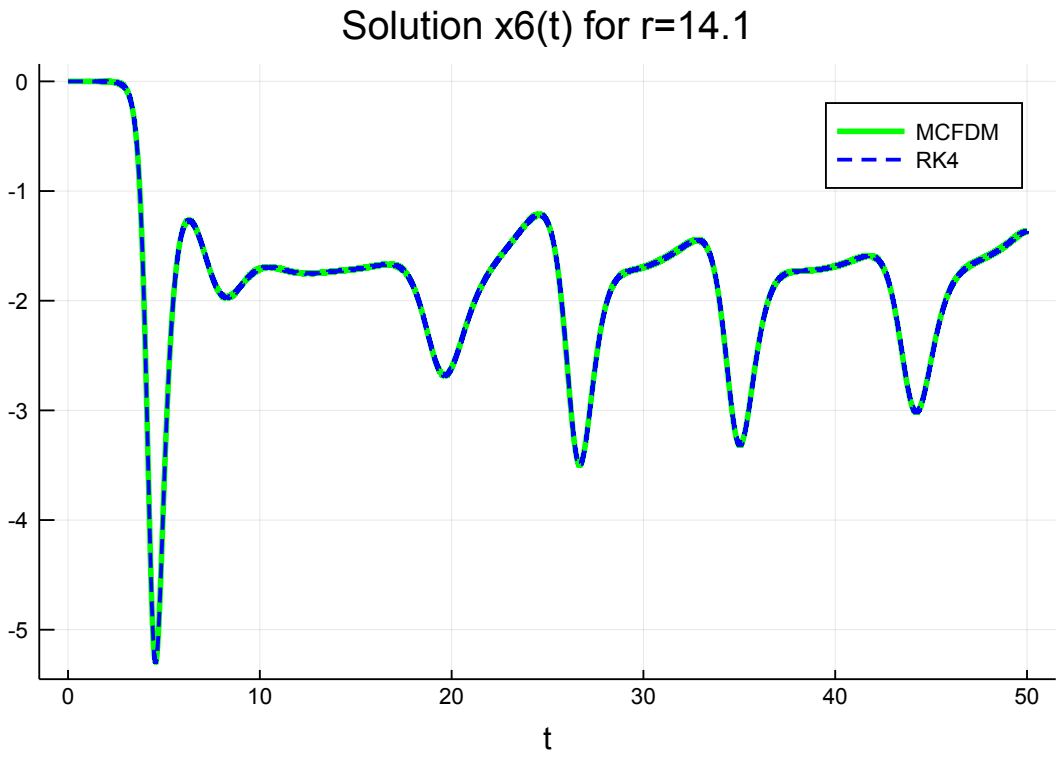
In [22]:
```
plot(tspan, Sol[5], color="lime", label="MCFDM", xlabel="t", title="Soluti
plot!(tspan1, Y[5], color="blue", label="RK4", xlabel="t", linewidth=2.0,l
```

Out[22]:



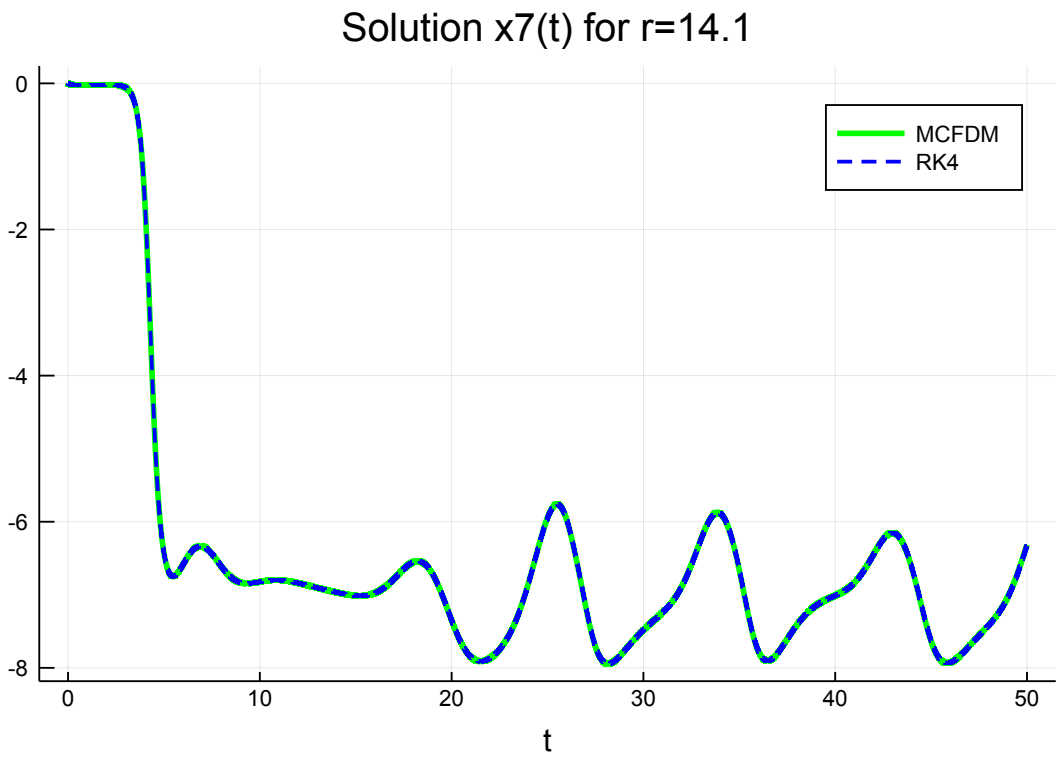

```
In [23]: plot(tspan, Sol[6], color="lime", label="MCFDM", xlabel="t", title="Soluti
         plot!(tspan1, Y[6], color="blue", label="RK4", xlabel="t", linewidth=2.0,l
```

Out[23]:



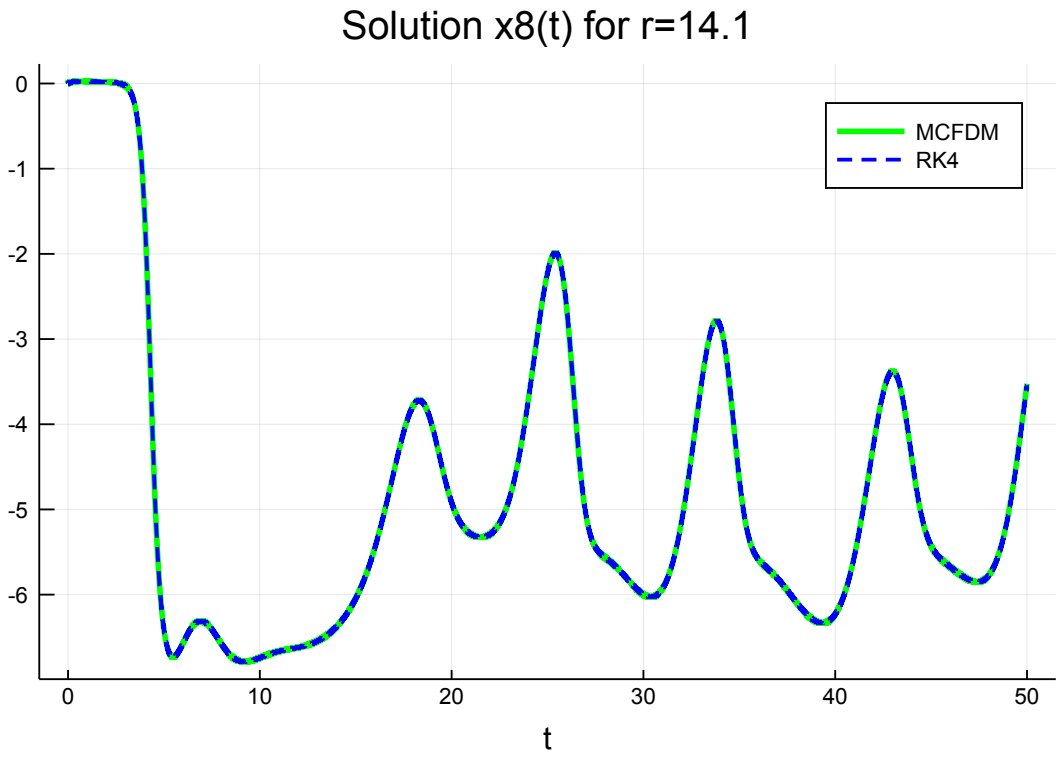

```
In [24]: plot(tspan, Sol[7], color="lime", label="MCFDM", xlabel="t", title="Soluti
         plot!(tspan1, Y[7], color="blue", label="RK4", xlabel="t", linewidth=2.0,l
```

Out[24]:



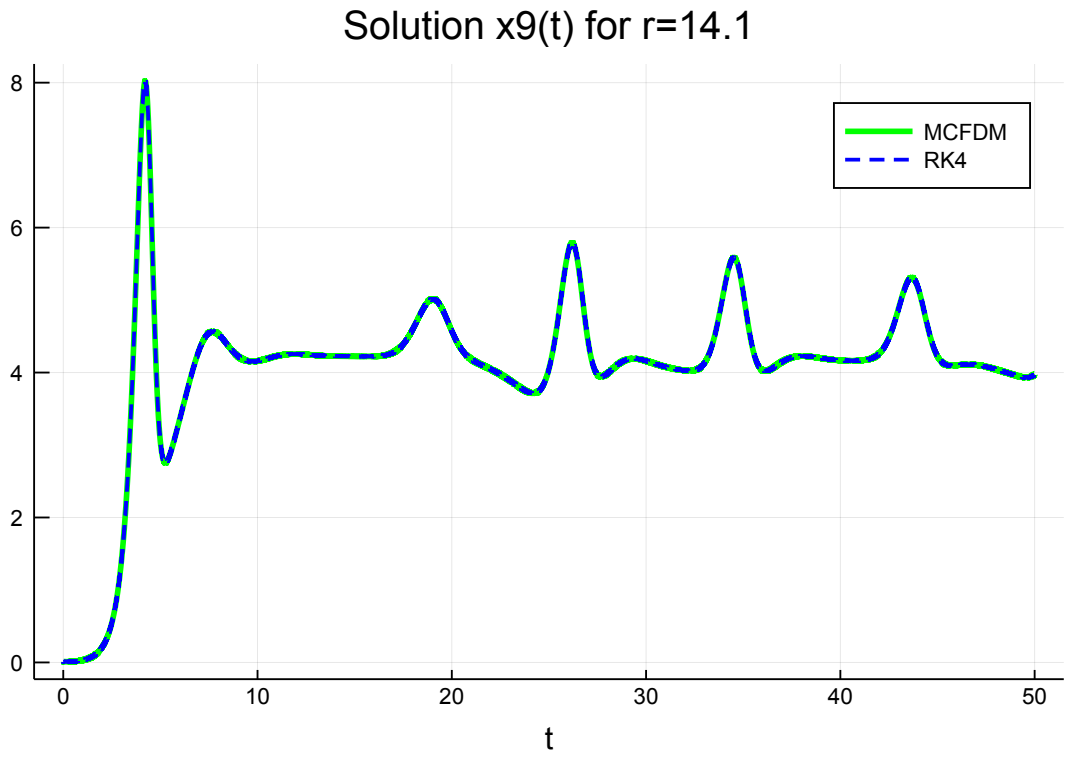

```
In [25]: plot(tspan, Sol[8], color="lime", label="MCFDM", xlabel="t", title="Soluti
         plot!(tspan1, Y[8], color="blue", label="RK4", xlabel="t", linewidth=2.0,l
```

Out[25]:



In [26]:
```
plot(tspan, Sol[9], color="lime", label="MCFDM", xlabel="t", title="Soluti
plot!(tspan1, Y[9], color="blue", label="RK4", xlabel="t",linewidth=2.0, l
```

Out[26]:



In [ ]:

1 Phase plots

1.1 r=14.1

In [12]:
```
r=15.1

        alpha11=σ*b1; alpha12=0.0; alpha13=0.0; alpha14=0.0; alpha15=0.0; alpha16=
        alpha18=0.0;alpha19=0.0;

        alpha21=0.0; alpha22=σ; alpha23=0.0; alpha24=0.0; alpha25=0.0; alpha26=0.0
        alpha28=0.0;alpha29=0.5*σ;

        alpha31=0.0; alpha32=0.0; alpha33=σ*b1; alpha34=0.0; alpha35=0.0; alpha36=
        alpha38=-σ*b2;alpha39=0.0;

        alpha41=0.0; alpha42=0.0; alpha43=0.0; alpha44=σ; alpha45=0.0; alpha46=0.0
        alpha48=0.0;alpha49=-0.5*σ;

        alpha51=0.0; alpha52=0.0; alpha53=0.0; alpha54=0.0; alpha55=σ*b5; alpha56=
        alpha58=0.0;alpha59=0.0
```

```
alpha61=0.0; alpha62=0.0; alpha63=0.0; alpha64=0.0; alpha65=0.0; alpha66=b
alpha68=0.0;alpha69=0.0

alpha71=r; alpha72=0.0; alpha73=0.0; alpha74=0.0; alpha75=0.0; alpha76=0.0
alpha78=0.0;alpha79=0.0

alpha81=0.0; alpha82=0.0; alpha83=-r; alpha84=0.0; alpha85=0.0; alpha86=0.
alpha88=b1;alpha89=0.0

alpha91=0.0; alpha92=r; alpha93=0.0; alpha94=-r; alpha95=0.0; alpha96=0.0;
alpha98=0.0;alpha99=1.0


Alphas=[[alpha11,alpha12,alpha13,alpha14,alpha15,alpha16,alpha17,alpha18,a
    [alpha21,alpha22,alpha23,alpha24,alpha25,alpha26,alpha27,alpha28,alpha
    [alpha31,alpha32,alpha33,alpha34,alpha35,alpha36,alpha37,alpha38,alpha
    [alpha41,alpha42,alpha43,alpha44,alpha45,alpha46,alpha47,alpha48,alpha
    [alpha51,alpha52,alpha53,alpha54,alpha55,alpha56,alpha57,alpha58,alpha
    [alpha61,alpha62,alpha63,alpha64,alpha65,alpha66,alpha67,alpha68,alpha
    [alpha71,alpha72,alpha73,alpha74,alpha75,alpha76,alpha77,alpha78,alpha
    [alpha81,alpha82,alpha83,alpha84,alpha85,alpha86,alpha87,alpha88,alpha
    [alpha91,alpha92,alpha93,alpha94,alpha95,alpha96,alpha97,alpha98,alpha
];

# Time interval [I0, I1]
I0=0.0
I1=200.0


IC=[0.01,0.0,0.01,0.0,0.0,0.0,0.0,0.0,0.01]

Max_iter=8

lower_bound=I0

Sol=[[] for s=1:length(IC)]

while lower_bound<I1
    Ur=Solver(f,IC, Max_iter)
    for l=1:length(IC)
            push!(Sol[l],Ur[l]...)
    end
    for r=1:length(IC)
        IC[r]=Sol[r][end]
    end
    lower_bound += p
end
```

```
In [13]: length(Sol[1])

Out[13]: 1100022

In [126]: plot(Sol[6],Sol[7],label="", linewidth=1.5, xlabel="x6(t)", ylabel="x7(t)
          #savefig("9dlorpp1.png")

In [ ]:

In [ ]:
```

1.2 r=14.10

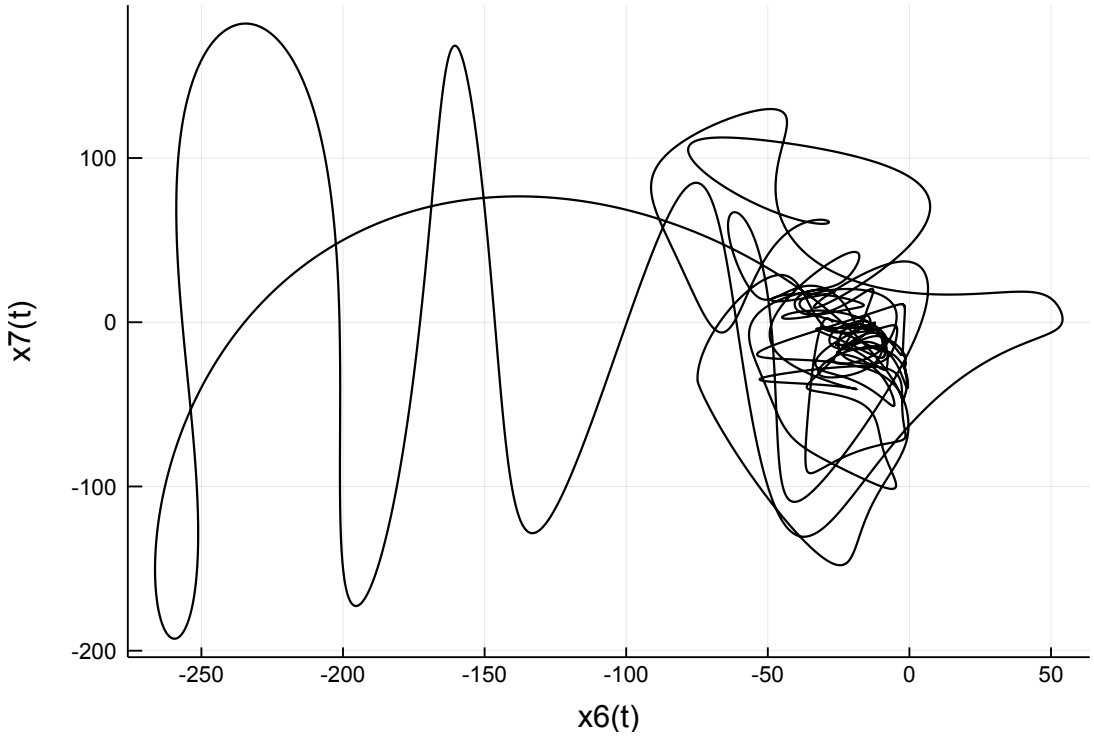
```
In [ ]:

In [19]: r=14.10

         alpha11=σ*b1; alpha12=0.0; alpha13=0.0; alpha14=0.0; alpha15=0.0; alpha16=
         alpha18=0.0;alpha19=0.0;

         alpha21=0.0; alpha22=σ; alpha23=0.0; alpha24=0.0; alpha25=0.0; alpha26=0.0
         alpha28=0.0;alpha29=0.5*σ;

         alpha31=0.0; alpha32=0.0; alpha33=σ*b1; alpha34=0.0; alpha35=0.0; alpha36=
         alpha38=-σ*b2;alpha39=0.0;

         alpha41=0.0; alpha42=0.0; alpha43=0.0; alpha44=σ; alpha45=0.0; alpha46=0.0
         alpha48=0.0;alpha49=-0.5*σ;

         alpha51=0.0; alpha52=0.0; alpha53=0.0; alpha54=0.0; alpha55=σ*b5; alpha56=
         alpha58=0.0;alpha59=0.0

         alpha61=0.0; alpha62=0.0; alpha63=0.0; alpha64=0.0; alpha65=0.0; alpha66=b
         alpha68=0.0;alpha69=0.0

         alpha71=r; alpha72=0.0; alpha73=0.0; alpha74=0.0; alpha75=0.0; alpha76=0.0
         alpha78=0.0;alpha79=0.0

         alpha81=0.0; alpha82=0.0; alpha83=-r; alpha84=0.0; alpha85=0.0; alpha86=0.
         alpha88=b1;alpha89=0.0

         alpha91=0.0; alpha92=r; alpha93=0.0; alpha94=-r; alpha95=0.0; alpha96=0.0;
         alpha98=0.0;alpha99=1.0


         Alphas=[[alpha11,alpha12,alpha13,alpha14,alpha15,alpha16,alpha17,alpha18,a
             [alpha21,alpha22,alpha23,alpha24,alpha25,alpha26,alpha27,alpha28,alpha
             [alpha31,alpha32,alpha33,alpha34,alpha35,alpha36,alpha37,alpha38,alpha
             [alpha41,alpha42,alpha43,alpha44,alpha45,alpha46,alpha47,alpha48,alpha
```

```
        [alpha51,alpha52,alpha53,alpha54,alpha55,alpha56,alpha57,alpha58,alpha
        [alpha61,alpha62,alpha63,alpha64,alpha65,alpha66,alpha67,alpha68,alpha
        [alpha71,alpha72,alpha73,alpha74,alpha75,alpha76,alpha77,alpha78,alpha
        [alpha81,alpha82,alpha83,alpha84,alpha85,alpha86,alpha87,alpha88,alpha
        [alpha91,alpha92,alpha93,alpha94,alpha95,alpha96,alpha97,alpha98,alpha
    ];
```

In [20]:
```
    # Time interval [I0, I1]
    I0=0.0
    I1=300.0

    #initial conditions

    IC=[0.01,0.0,0.01,0.0,0.0,0.0,0.0,0.0,0.01]

    Max_iter=10

    lower_bound=I0

    Sol=[[] for s=1:length(IC)]

    while lower_bound<I1
        Ur=Solver(f,IC, Max_iter)
        for l=1:length(IC)
            push!(Sol[l],Ur[l]...)
        end
        for r=1:length(IC)
            IC[r]=Sol[r][end]
        end
        lower_bound += p
    end
```

In [63]:

Out[63]:



In [ ]:

1.3 r=15.0

In [22]:
```
r=15.0

alpha11=σ*b1; alpha12=0.0; alpha13=0.0; alpha14=0.0; alpha15=0.0; alpha16=
alpha18=0.0;alpha19=0.0;

alpha21=0.0; alpha22=σ; alpha23=0.0; alpha24=0.0; alpha25=0.0; alpha26=0.0
alpha28=0.0;alpha29=0.5*σ;

alpha31=0.0; alpha32=0.0; alpha33=σ*b1; alpha34=0.0; alpha35=0.0; alpha36=
alpha38=-σ*b2;alpha39=0.0;

alpha41=0.0; alpha42=0.0; alpha43=0.0; alpha44=σ; alpha45=0.0; alpha46=0.0
alpha48=0.0;alpha49=-0.5*σ;

alpha51=0.0; alpha52=0.0; alpha53=0.0; alpha54=0.0; alpha55=σ*b5; alpha56=
alpha58=0.0;alpha59=0.0

alpha61=0.0; alpha62=0.0; alpha63=0.0; alpha64=0.0; alpha65=0.0; alpha66=b
alpha68=0.0;alpha69=0.0
```

```
alpha71=r; alpha72=0.0; alpha73=0.0; alpha74=0.0; alpha75=0.0; alpha76=0.0
alpha78=0.0;alpha79=0.0

alpha81=0.0; alpha82=0.0; alpha83=-r; alpha84=0.0; alpha85=0.0; alpha86=0.
alpha88=b1;alpha89=0.0

alpha91=0.0; alpha92=r; alpha93=0.0; alpha94=-r; alpha95=0.0; alpha96=0.0;
alpha98=0.0;alpha99=1.0


Alphas=[[alpha11,alpha12,alpha13,alpha14,alpha15,alpha16,alpha17,alpha18,a
    [alpha21,alpha22,alpha23,alpha24,alpha25,alpha26,alpha27,alpha28,alpha
    [alpha31,alpha32,alpha33,alpha34,alpha35,alpha36,alpha37,alpha38,alpha
    [alpha41,alpha42,alpha43,alpha44,alpha45,alpha46,alpha47,alpha48,alpha
    [alpha51,alpha52,alpha53,alpha54,alpha55,alpha56,alpha57,alpha58,alpha
    [alpha61,alpha62,alpha63,alpha64,alpha65,alpha66,alpha67,alpha68,alpha
    [alpha71,alpha72,alpha73,alpha74,alpha75,alpha76,alpha77,alpha78,alpha
    [alpha81,alpha82,alpha83,alpha84,alpha85,alpha86,alpha87,alpha88,alpha
    [alpha91,alpha92,alpha93,alpha94,alpha95,alpha96,alpha97,alpha98,alpha
];
```

In [23]:
```
# Time interval [I0, I1]
I0=0.0
I1=300.0

#initial conditions

IC=[0.01,0.0,0.01,0.0,0.0,0.0,0.0,0.0,0.01]

Max_iter=10

lower_bound=I0

Sol=[[] for s=1:length(IC)]

while lower_bound<I1
    Ur=Solver(f,IC, Max_iter)
    for l=1:length(IC)
        push!(Sol[l],Ur[l]...)
    end
    for r=1:length(IC)
        IC[r]=Sol[r][end]
    end
    lower_bound += p
end
```

In [22]:
```
plot(Sol[6][end-40000:end],Sol[9][end-40000:end],label="", xlabel="x6(t)",
savefig("9dlorpp2.png")
```

In [ ]:

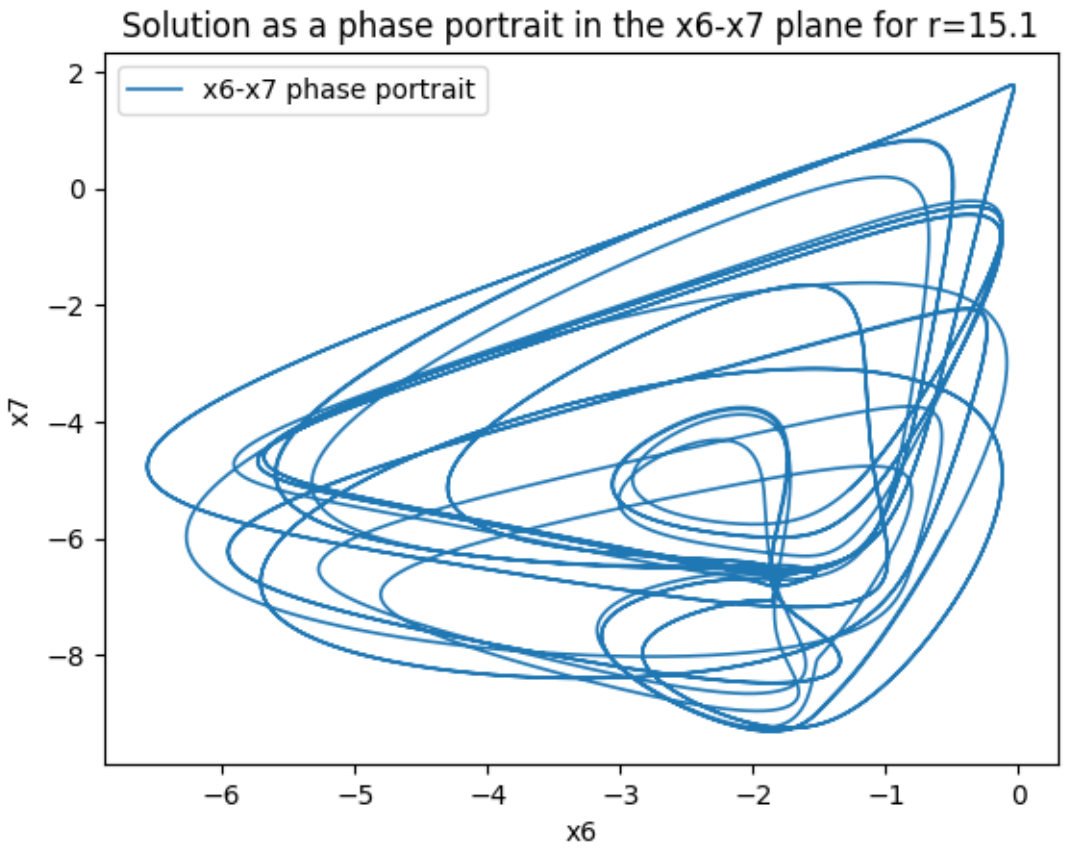
1.4 r=15.1

In [25]:
```
r=15.10

alpha11=σ*b1; alpha12=0.0; alpha13=0.0; alpha14=0.0; alpha15=0.0; alpha16=
alpha18=0.0;alpha19=0.0;

alpha21=0.0; alpha22=σ; alpha23=0.0; alpha24=0.0; alpha25=0.0; alpha26=0.0
alpha28=0.0;alpha29=0.5*σ;

alpha31=0.0; alpha32=0.0; alpha33=σ*b1; alpha34=0.0; alpha35=0.0; alpha36=
alpha38=-σ*b2;alpha39=0.0;

alpha41=0.0; alpha42=0.0; alpha43=0.0; alpha44=σ; alpha45=0.0; alpha46=0.0
alpha48=0.0;alpha49=-0.5*σ;

alpha51=0.0; alpha52=0.0; alpha53=0.0; alpha54=0.0; alpha55=σ*b5; alpha56=
alpha58=0.0;alpha59=0.0

alpha61=0.0; alpha62=0.0; alpha63=0.0; alpha64=0.0; alpha65=0.0; alpha66=b
alpha68=0.0;alpha69=0.0

alpha71=r; alpha72=0.0; alpha73=0.0; alpha74=0.0; alpha75=0.0; alpha76=0.0
alpha78=0.0;alpha79=0.0

alpha81=0.0; alpha82=0.0; alpha83=-r; alpha84=0.0; alpha85=0.0; alpha86=0.
alpha88=b1;alpha89=0.0

alpha91=0.0; alpha92=r; alpha93=0.0; alpha94=-r; alpha95=0.0; alpha96=0.0;
alpha98=0.0;alpha99=1.0


Alphas=[[alpha11,alpha12,alpha13,alpha14,alpha15,alpha16,alpha17,alpha18,a
    [alpha21,alpha22,alpha23,alpha24,alpha25,alpha26,alpha27,alpha28,alpha
    [alpha31,alpha32,alpha33,alpha34,alpha35,alpha36,alpha37,alpha38,alpha
    [alpha41,alpha42,alpha43,alpha44,alpha45,alpha46,alpha47,alpha48,alpha
    [alpha51,alpha52,alpha53,alpha54,alpha55,alpha56,alpha57,alpha58,alpha
    [alpha61,alpha62,alpha63,alpha64,alpha65,alpha66,alpha67,alpha68,alpha
    [alpha71,alpha72,alpha73,alpha74,alpha75,alpha76,alpha77,alpha78,alpha
    [alpha81,alpha82,alpha83,alpha84,alpha85,alpha86,alpha87,alpha88,alpha
    [alpha91,alpha92,alpha93,alpha94,alpha95,alpha96,alpha97,alpha98,alpha
];
```

In [26]:
```
# Time interval [I0, I1]
I0=0.0
I1=300.0
```

```
#initial conditions

IC=[0.01,0.0,0.01,0.0,0.0,0.0,0.0,0.0,0.01]

Max_iter=10

lower_bound=I0

Sol=[[] for s=1:length(IC)]

while lower_bound<I1
    Ur=Solver(f,IC, Max_iter)
    for l=1:length(IC)
        push!(Sol[l],Ur[l]...)
    end
    for r=1:length(IC)
        IC[r]=Sol[r][end]
    end
    lower_bound += p
end
```

In [27]:
```
plot(Sol[6][end-150000:end],Sol[7][end-150000:end],label="x6-x7 phase port
xlabel("x6")
ylabel("x7")
legend()
title("Solution as a phase portrait in the x6-x7 plane for r=$r")
```



Out[27]: PyObject Text(0.5, 1.0, 'Solution as a phase portrait in the x6-x7 plane f

In [ ]:

1.5 r=24.0

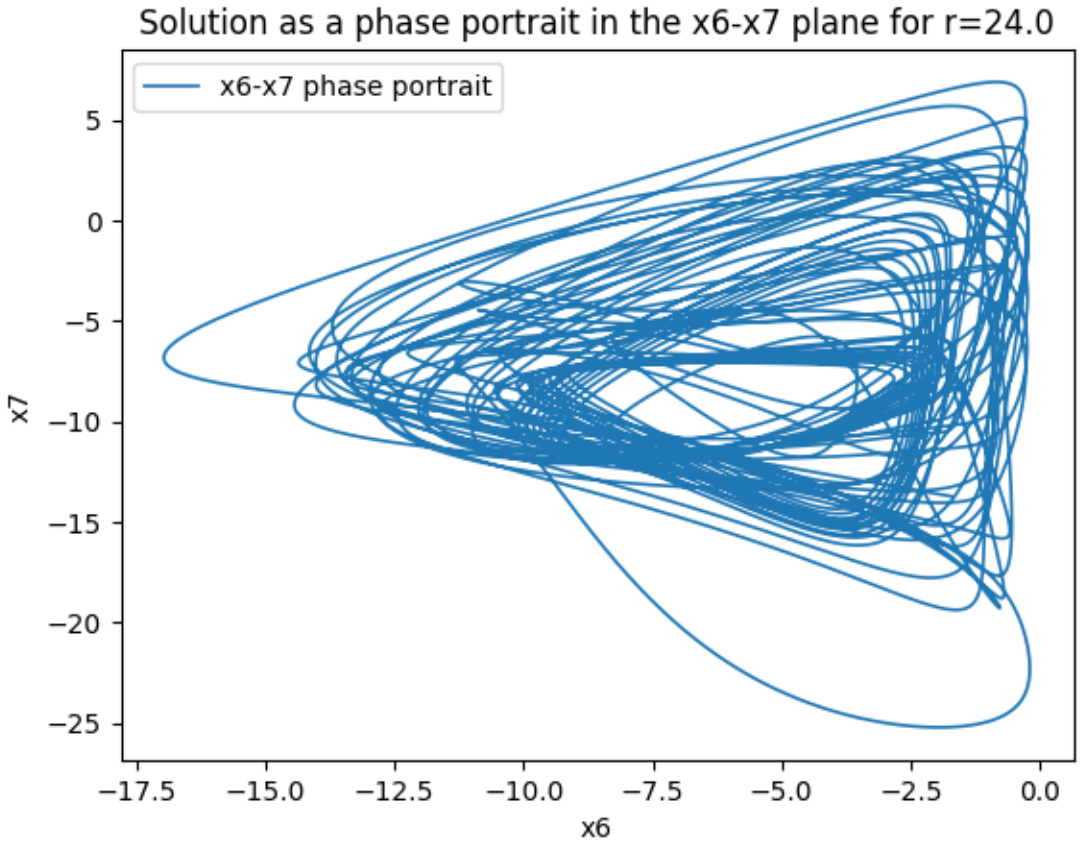
In [50]:
```
r=24.0

        alpha11=σ*b1; alpha12=0.0; alpha13=0.0; alpha14=0.0; alpha15=0.0; alpha16=
        alpha18=0.0;alpha19=0.0;

        alpha21=0.0; alpha22=σ; alpha23=0.0; alpha24=0.0; alpha25=0.0; alpha26=0.0
        alpha28=0.0;alpha29=0.5*σ;

        alpha31=0.0; alpha32=0.0; alpha33=σ*b1; alpha34=0.0; alpha35=0.0; alpha36=
        alpha38=-σ*b2;alpha39=0.0;

        alpha41=0.0; alpha42=0.0; alpha43=0.0; alpha44=σ; alpha45=0.0; alpha46=0.0
        alpha48=0.0;alpha49=-0.5*σ;
```

```
alpha51=0.0; alpha52=0.0; alpha53=0.0; alpha54=0.0; alpha55=σ*b5; alpha56=
alpha58=0.0;alpha59=0.0

alpha61=0.0; alpha62=0.0; alpha63=0.0; alpha64=0.0; alpha65=0.0; alpha66=b
alpha68=0.0;alpha69=0.0

alpha71=r; alpha72=0.0; alpha73=0.0; alpha74=0.0; alpha75=0.0; alpha76=0.0
alpha78=0.0;alpha79=0.0

alpha81=0.0; alpha82=0.0; alpha83=-r; alpha84=0.0; alpha85=0.0; alpha86=0.
alpha88=b1;alpha89=0.0

alpha91=0.0; alpha92=r; alpha93=0.0; alpha94=-r; alpha95=0.0; alpha96=0.0;
alpha98=0.0;alpha99=1.0


Alphas=[[alpha11,alpha12,alpha13,alpha14,alpha15,alpha16,alpha17,alpha18,a
    [alpha21,alpha22,alpha23,alpha24,alpha25,alpha26,alpha27,alpha28,alpha
    [alpha31,alpha32,alpha33,alpha34,alpha35,alpha36,alpha37,alpha38,alpha
    [alpha41,alpha42,alpha43,alpha44,alpha45,alpha46,alpha47,alpha48,alpha
    [alpha51,alpha52,alpha53,alpha54,alpha55,alpha56,alpha57,alpha58,alpha
    [alpha61,alpha62,alpha63,alpha64,alpha65,alpha66,alpha67,alpha68,alpha
    [alpha71,alpha72,alpha73,alpha74,alpha75,alpha76,alpha77,alpha78,alpha
    [alpha81,alpha82,alpha83,alpha84,alpha85,alpha86,alpha87,alpha88,alpha
    [alpha91,alpha92,alpha93,alpha94,alpha95,alpha96,alpha97,alpha98,alpha
];
```

In [51]:
```
# Time interval [I0, I1]
I0=0.0
I1=300.0

#initial conditions

IC=[0.01,0.0,0.01,0.0,0.0,0.0,0.0,0.0,0.01]

Max_iter=10

lower_bound=I0

Sol=[[] for s=1:length(IC)]

while lower_bound<I1
    Ur=Solver(f,IC, Max_iter)
    for l=1:length(IC)
        push!(Sol[l],Ur[l]...)
    end
    for r=1:length(IC)
        IC[r]=Sol[r][end]
```

```
        end
        lower_bound += p
    end
```

In [52]:
```
plot(Sol[6][end-120000:end],Sol[7][end-120000:end],label="x6-x7 phase port
xlabel("x6")
ylabel("x7")
legend()
title("Solution as a phase portrait in the x6-x7 plane for r=$r")
```



Out[52]: PyObject Text(0.5, 1.0, 'Solution as a phase portrait in the x6-x7 plane f

In [ ]: