Homework # 4: Computational Physics

Name: Pablo Lopez

October 7th, 2020

Problem 1

b) Using the adaptive Runge-Kutta technique, we can easily solve the resulting equation:

$$\begin{cases} \frac{\mathrm{dr}}{\mathrm{dt}} = r(a - r^2) \\ \frac{d\theta}{\mathrm{dt}} = -1 \end{cases}$$

in which we include the system of uncoupled differential equations.

In order to solve it, we must then define the parameters and initial conditions for our system:

```
A = 2;
ms=3;
tspan = [0 3.5];
yinit=[0.0001,0.5*pi;0.1,0.5*pi;0.5,0.5*pi;1.5,0.5*pi;4.0,0.5*pi];
```

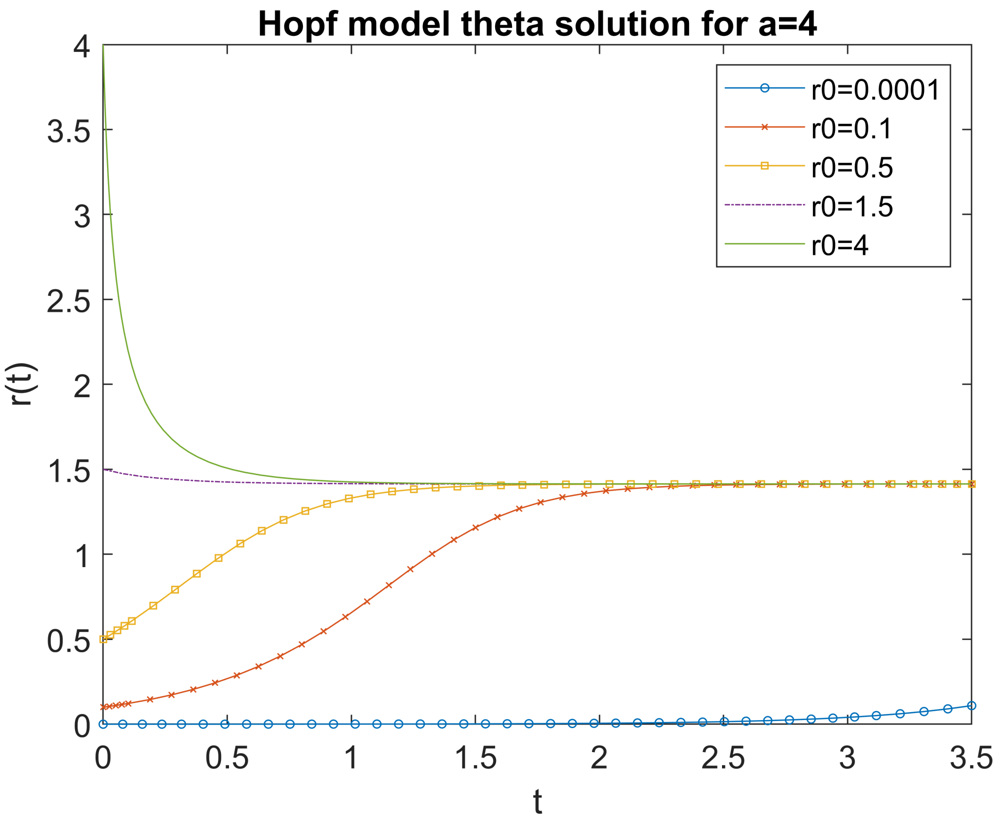
where we define a 5x2 matrix to automatize the solutions for several initial conditions. In the same manner we automatize the marker selection:

```
chars=["-o","-x","-s","-.","-"];
```

Then, we can use ode45 to solve the system, plot the solutions in the same figure and save it :

```
for ii=1:size(yinit,1)
    [t,y] = ode45(@(t,y) odefcn(t,y,A), tspan, yinit(ii,:));
    plot(t,y(:,1),chars(ii),'MarkerSize',ms)
    hold all
end
title('Hopf model theta solution for a=4');

legend('r0=0.0001','r0=0.1','r0=0.5','r0=1.5','r0=4');
box on

ax=gca;
ax.FontSize=12;

xlabel('t');
ylabel('r(t)');
```

```
%saveas(gcf,'HW4_1b_loop_a4','epsc');

hold off
```



The simpler to analyze is the θ equation as it clearly represents a constant decrease over time.

In order to implement the code above, we will define the following function:

```
function dydt = odefcn(t,y,A)
dydt = zeros(2,1);
dydt(1) = y(1)*(A-y(1)^2);
dydt(2) = -1;
end
```