HW 5 Part 2

April 17, 2020

Files to include in Appendix: BackwardEuler.jl Adams*Bashforth2.jl GL*RK.jl MyImplicit-Midpoint.jl MyForwardEuler.jl

Simple Pendulum First I need to solve the ODE and put it into first order form

```
using Plots
using SymPy
@vars p q theta t

Ham = p^2/2 +cos(q)
```

$$\frac{p^2}{2} + \cos(q)$$

I need find the nondimentional momentum of the angle

```
thetadot = diff(Ham,p)
pdot = -diff(Ham,q) |> subs(q=>theta)
```

$$\sin(\theta)$$

therefore $thetaddot = sin(theta)$

we need to get it in first order form thetaddot = x1dot thetadot = x2dot

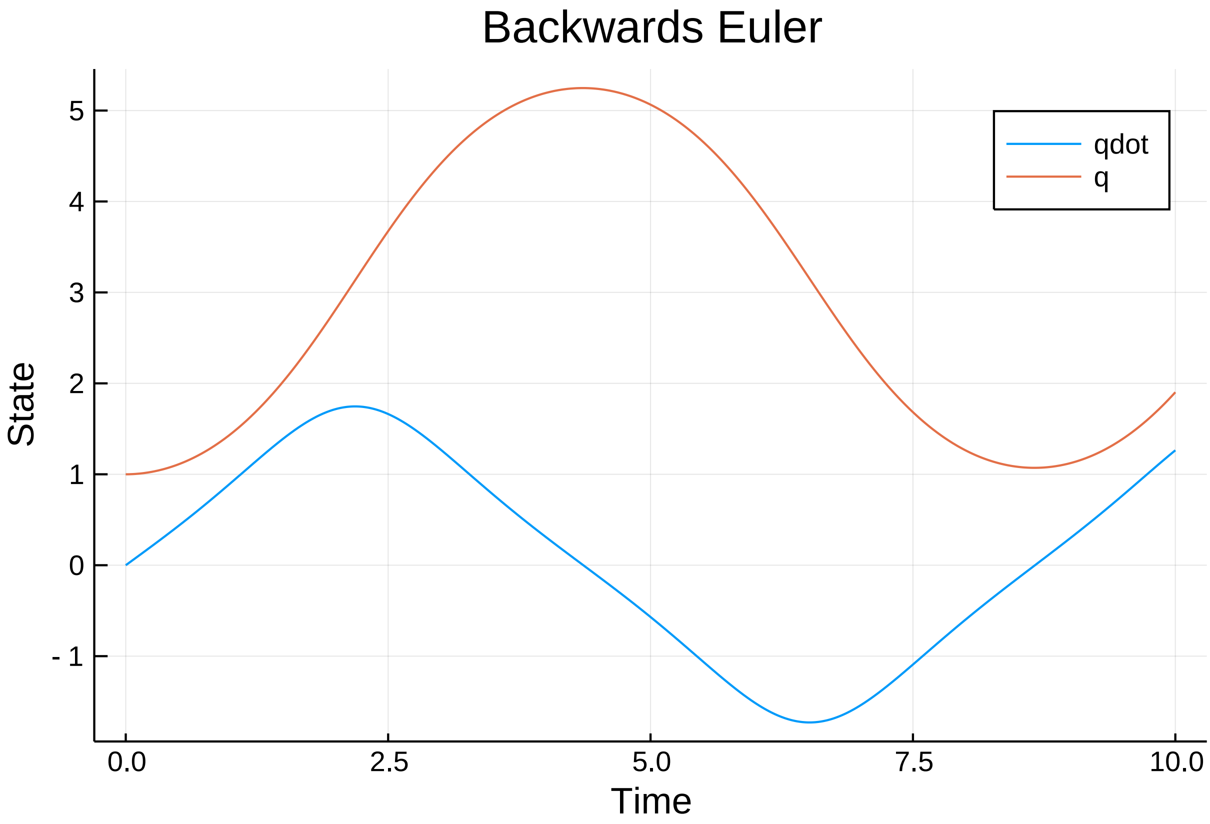
Now, there are two first order ODEs to solve.

1. $x1dot = sin(x2)$
2. $x2dot = x1$

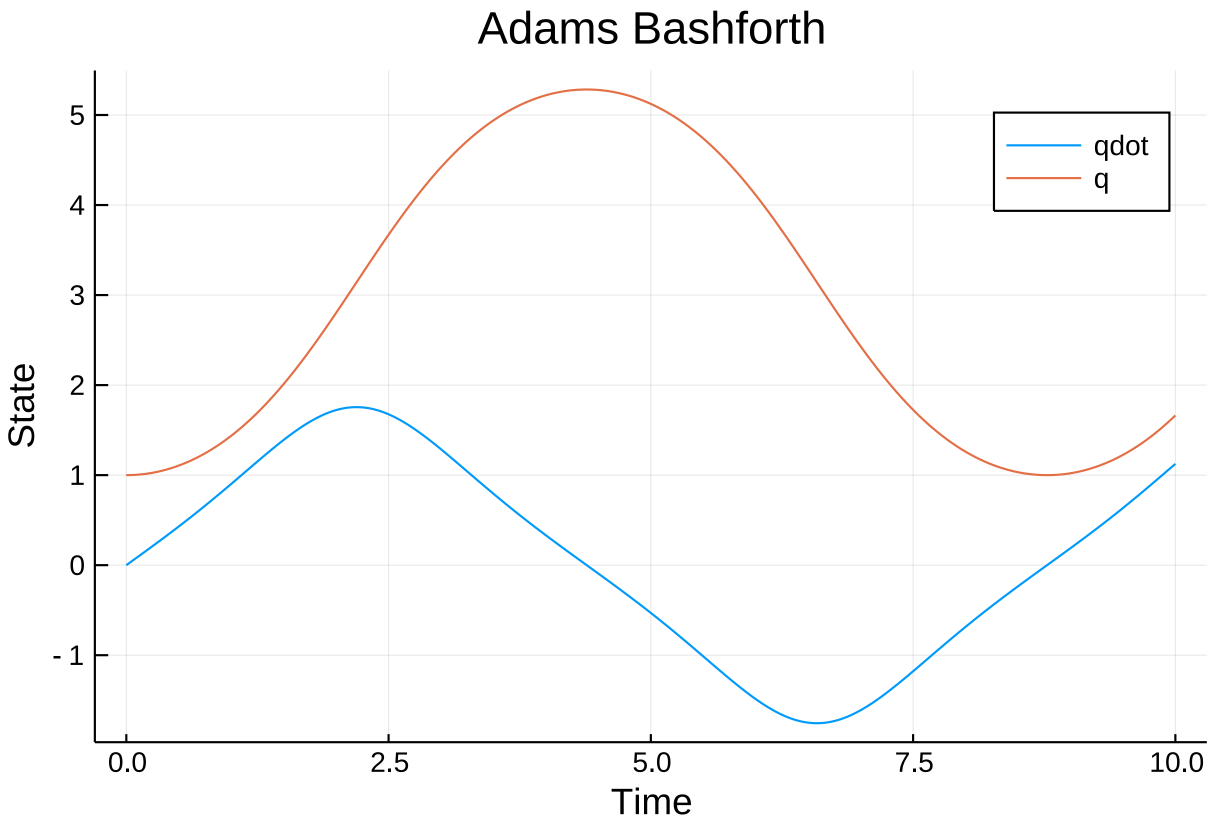
Now to solve them.

```
# Define givens
f(x) = [sin(x[2]) x[1]]
h = .01
x0 = [0 1.]
tf = 10;

include("BackwardEuler.jl")
x_BE,t = BackwardEuler.beuler(f,tf,h,x0)
plot(t,x_BE[:,1],label = "qdot")
    plot!(t,x_BE[:,2],label = "q")
```

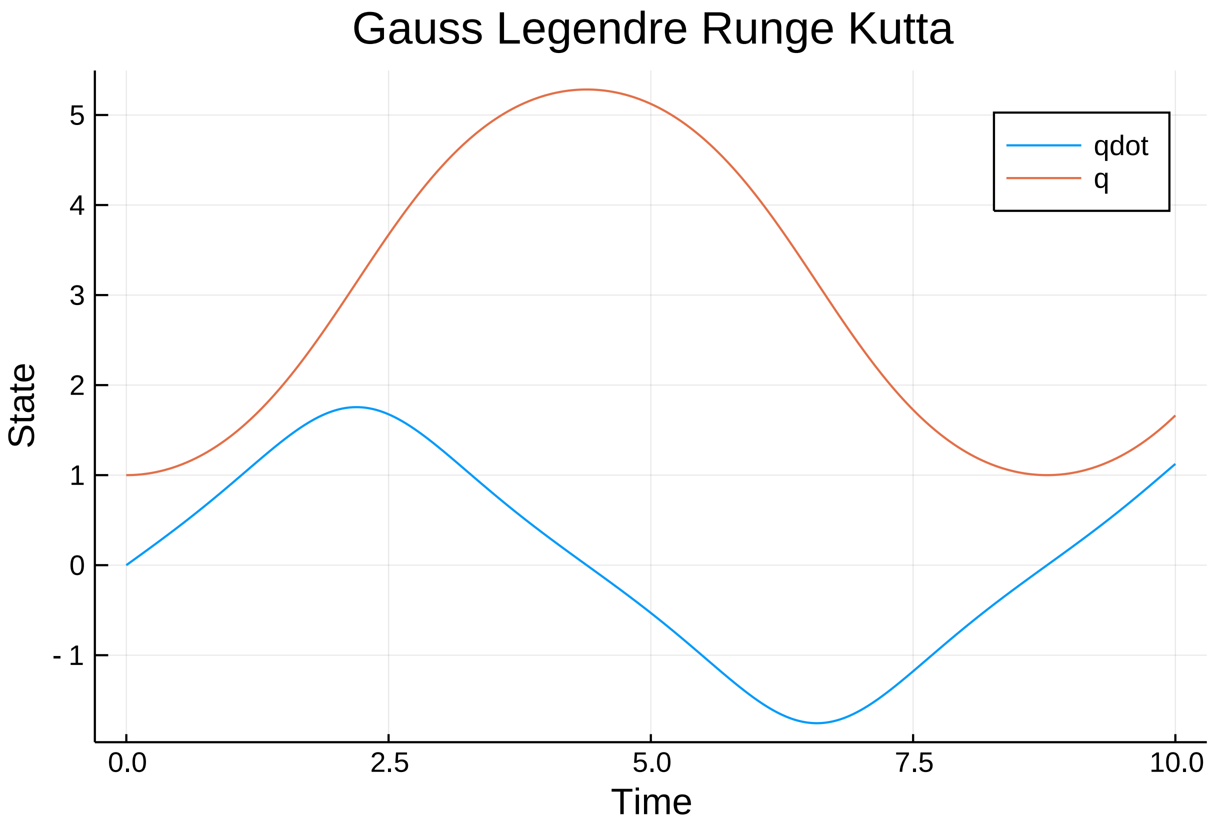
```
    xlabel!("Time")
    ylabel!("State")
    title!("Backwards Euler")
```



```
include("Adams_Bashforth2.jl")
x_AB,t = Adams_Bashforth2.ab2(f,tf,h,x0)
plot(t,x_AB[:,1],label = "qdot")
    plot!(t,x_AB[:,2],label = "q")
    xlabel!("Time")
    ylabel!("State")
    title!("Adams Bashforth")
```

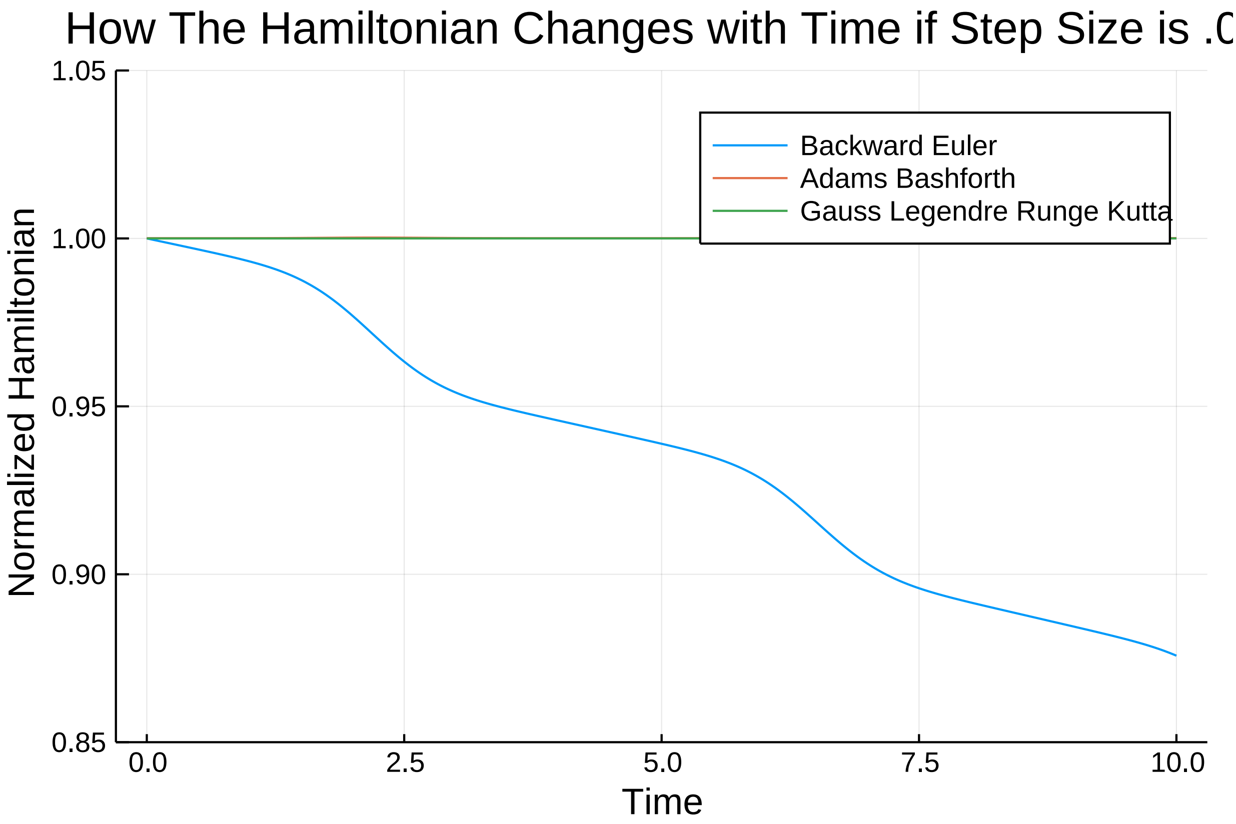



```
include("GL_RK.jl")
x_GL,t = GL_RK.gl_rk(f,tf,h,x0)
plot(t,x_GL[:,1],label = "qdot")
    plot!(t,x_GL[:,2],label = "q")
    xlabel!("Time")
    ylabel!("State")
    title!("Gauss Legendre Runge Kutta")
```



Now, I'll see what happens to the Hamiltonian as the solutions integrate in time.

```
H_BE = x_BE[:,1].^2/2 + cos.(x_BE[:,2])
H0 = x_BE[1,1].^2/2 + cos.(x_BE[1,2])

H_AB = x_AB[:,1].^2/2 +cos.(x_AB[:,2])

H_GL = x_GL[:,1].^2/2 +cos.(x_GL[:,2])

plot(t,H_BE/H0,label = "Backward Euler",yticks = .85:0.05:1.05,)
    plot!(t,H_AB/H0,label = "Adams Bashforth")
    plot!(t,H_GL/H0,label = "Gauss Legendre Runge Kutta")
    xlabel!("Time")
    ylabel!("Normalized Hamiltonian")
    title!("How The Hamiltonian Changes with Time if Step Size is .01")
    ylims!((.85,1.05))
```



AB2 was the easiest to implement for me becasue there were no implicit steps

Now, I'll compare the computation times

For Backwards Euler:

```
@elapsed BackwardEuler.beuler(f,tf,h,x0)
```

```
0.003468293
```

For Adams Bashforth 2

```
@elapsed Adams_Bashforth2.ab2(f,tf,h,x0)
```

```
0.003892726
```

For Gauss Legendre Runge Kutta

```
@elapsed GL_RK.gl_rk(f,tf,h,x0)
```

```
0.009404088
```

Gauss Legendre Runge Kutta took the longest

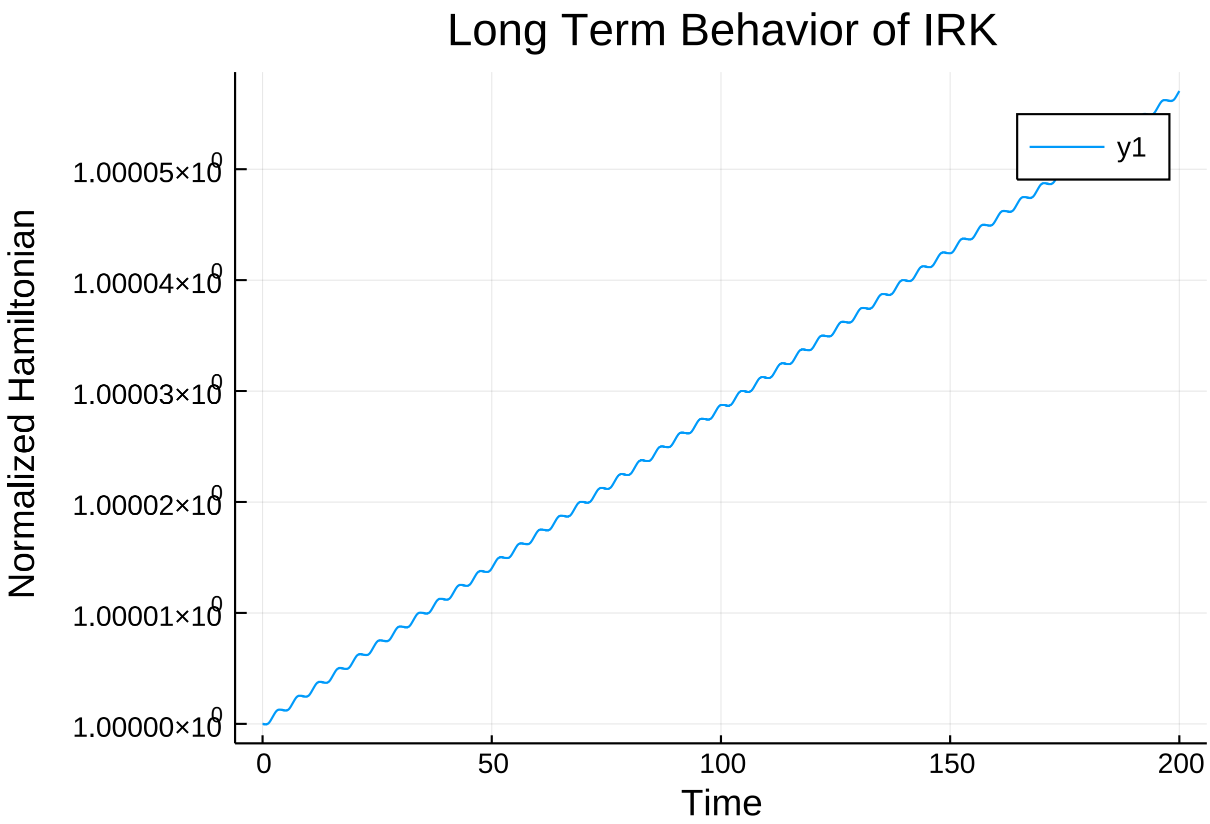
You asked us: Does the long term behavior of IRK method of (5.10) remain well behaved?

And I think the answer is that it depends. It probably depends on the step size and the eigenvalues of the problem.

So here, I integrated the above problem with IRK over a long time and with a fairly small step size.

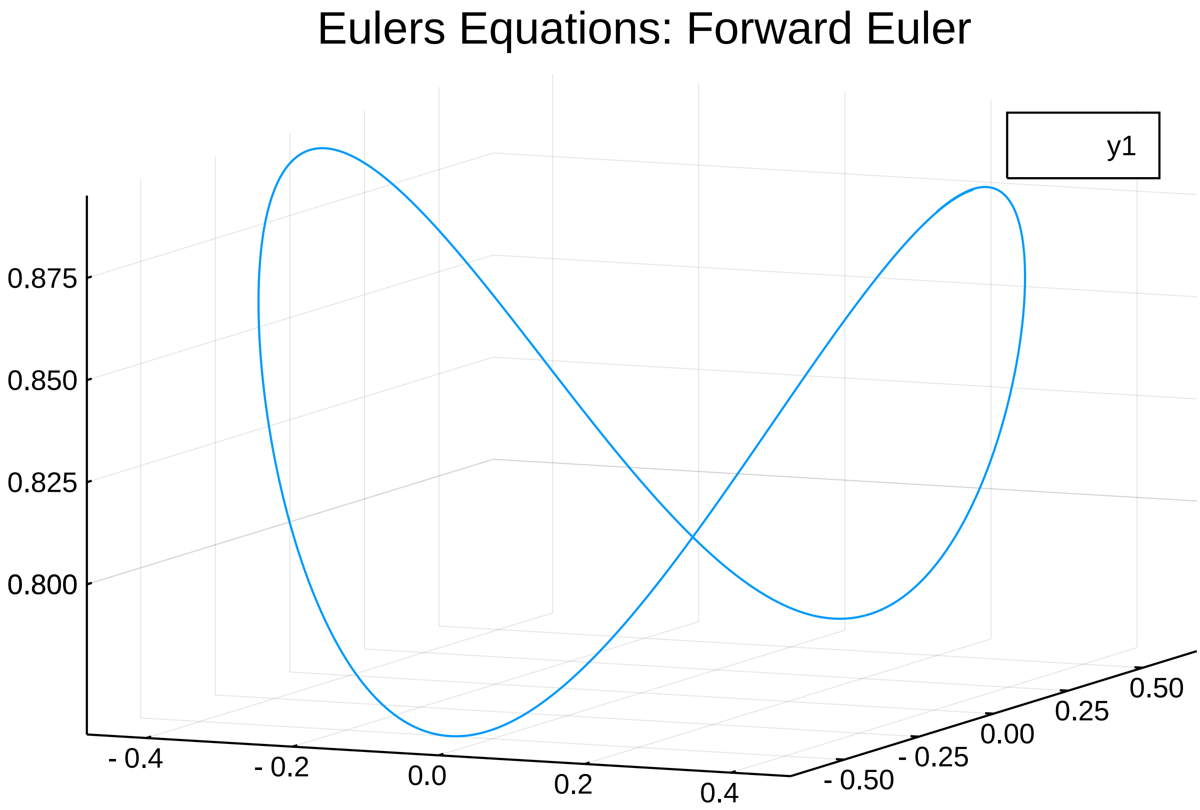
And the I'd say the long term behavior is pretty good!

```
tf_long = 200
h_long = .01
x_long,t = GL_RK.gl_rk(f,tf_long,h_long,x0)
H_GL_long = x_long[:,1].^2/2 +cos.(x_long[:,2])
plot(t,H_GL_long/H0)
    xlabel!("Time")
    ylabel!("Normalized Hamiltonian")
    title!("Long Term Behavior of IRK")
```

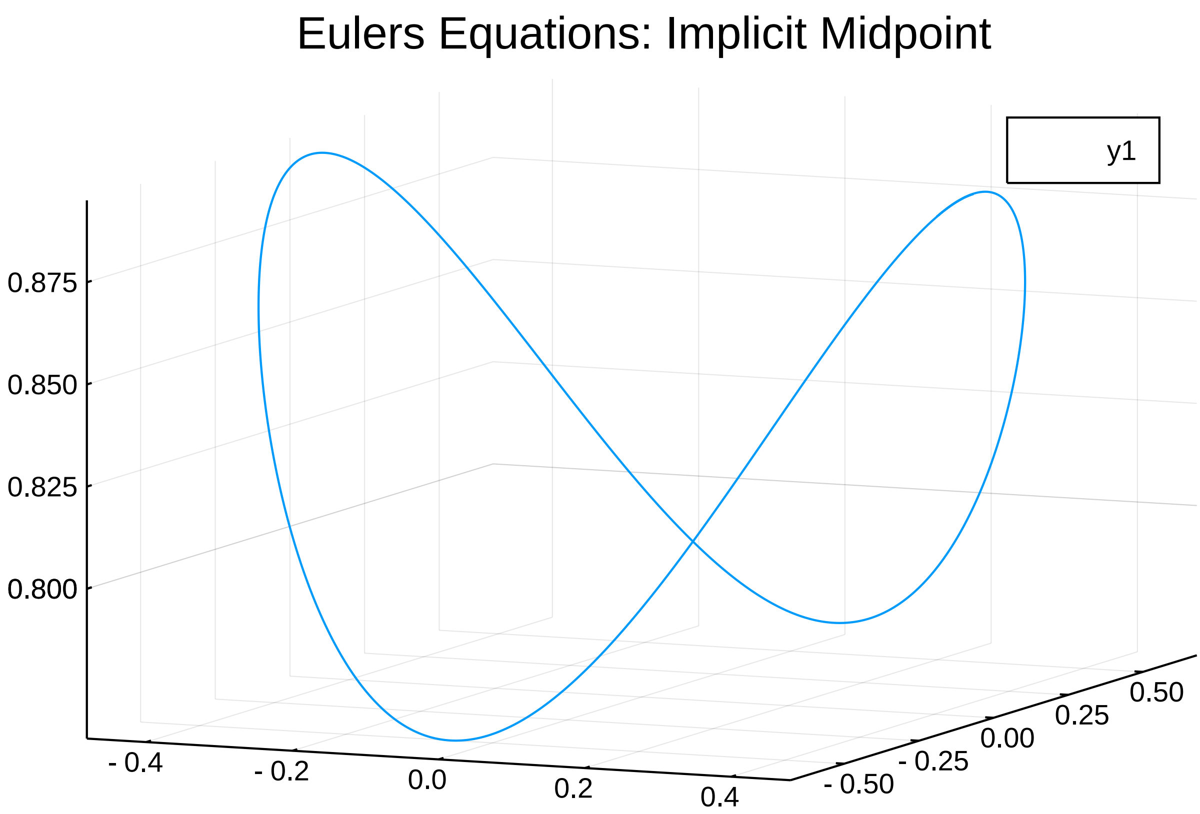


Now for The Euler Equations

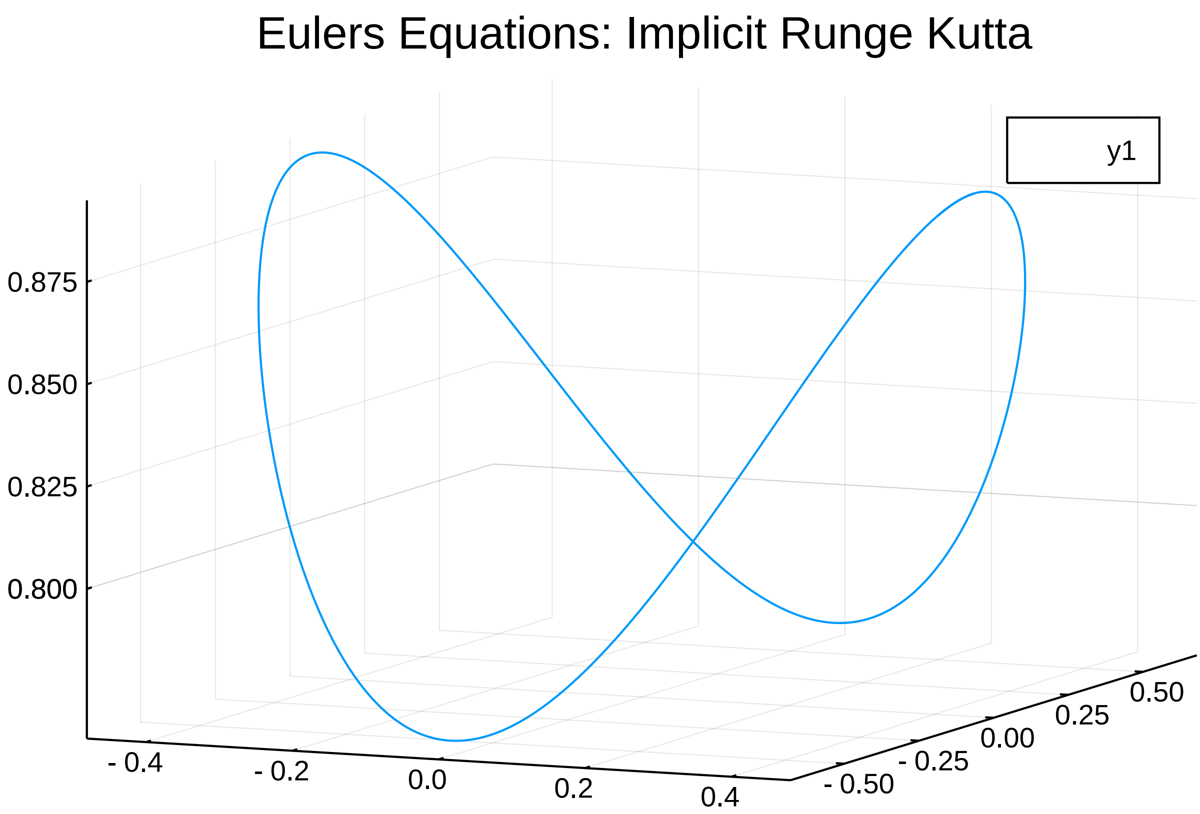
```
y0 = [cos(1.1) 0 sin(1.1)]

I1 = 2
I2 = 1
I3 = 2/3


a1 = (I2-I3)/I2/I3
a2 = (I3-I1)/I3/I1
a3 = (I1-I2)/I1/I2
f(y) = [a1*y[2]*y[3] a2*y[3]*y[1] a3*y[1]*y[2]]
h = .001
tf = 11
H0 = 1/2*(y0[1]^2/I1+y0[2]^2/I2+y0[3]^2/I3);

include("MyForwardEuler.jl")
y_FE,t = MyForwardEuler.feuler(f,tf,h,y0)
H_FE = 1/2*(y_FE[:,1].^2/I1+y_FE[:,2].^2/I2+y_FE[:,3].^2/I3)
plot3d(y_FE[:,1],y_FE[:,2],y_FE[:,3])
    title!("Eulers Equations: Forward Euler")
```




```
include("MyImplicitMidpoint.jl")
y_IM, t = MyImplicitMidpoint.my_implicit_mid(f,tf,h,y0)
H_IM = 1/2*(y_IM[:,1].^2/I1+y_IM[:,2].^2/I2+y_IM[:,3].^2/I3)
plot3d(y_IM[:,1],y_IM[:,2],y_IM[:,3])
    title!("Eulers Equations: Implicit Midpoint")
```



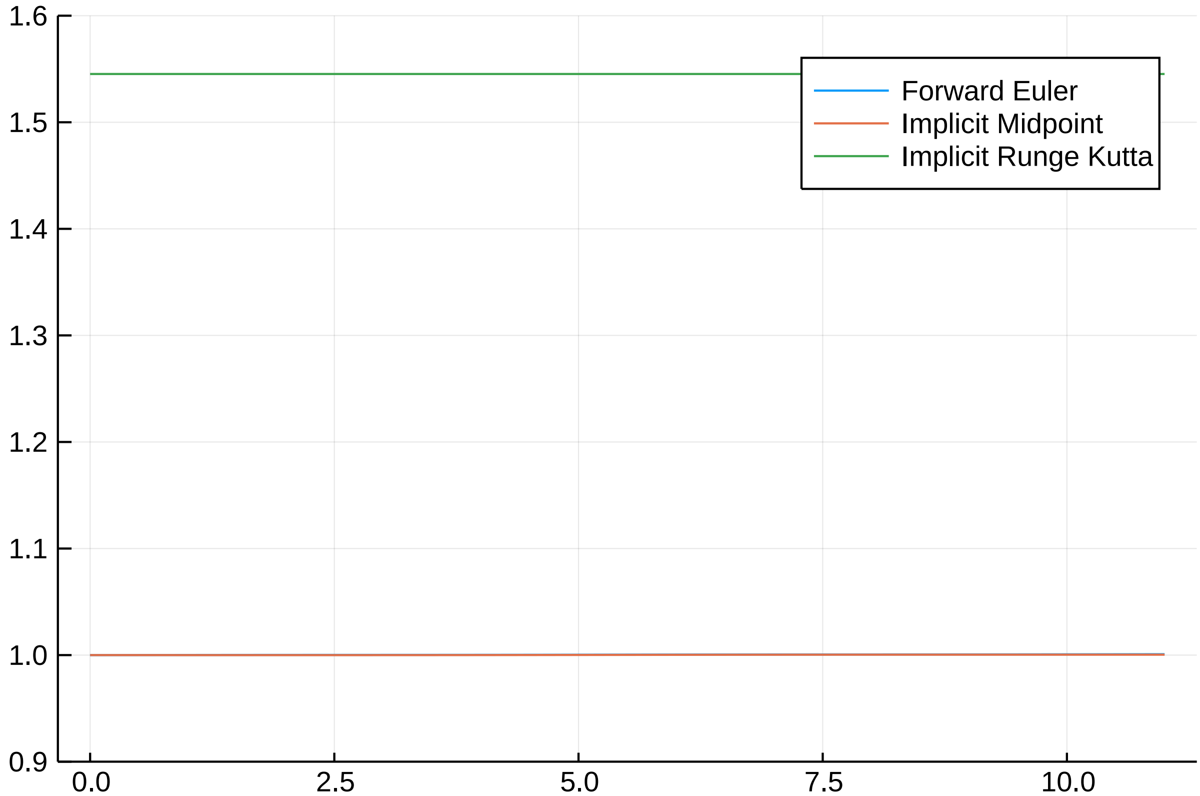

```
include("GL_RK.jl")
y_GL,ti = GL_RK.gl_rk(f,tf,h,y0)
H_GL = 1/2*(y_GL[:,1].^2/I1+y_GL[:,2].^2/I2+y_GL[:,3].^2/I3)/H0
plot3d(y_GL[:,1],y_GL[:,2],y_GL[:,3])
    title!("Eulers Equations: Implicit Runge Kutta")
```



The next plot, shows whether or not the Hamiltonian changes as the integrator moves forward. Only the IRK does not conserve the Hamiltonian.

Which, admittedly, is strange, because for the simple pendulum I said that IRK conserved the hamiltonian in its long term behavior....

```
plot(t,H_FE/H0,label= "Forward Euler")
    plot!(t,H_IM/H0,label = "Implicit Midpoint")
    plot!(t,H_GL/H0,label = "Implicit Runge Kutta")
    ylims!((.9,1.6))
```


Forward Euler
Implicit Midpoint
Implicit Runge Kutta
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.0
2.5
5.0
7.5
10.0