# Assignment #2

Kevin Corcoran

April 15, 2021

## 1 Problem 1

Suppose

$$k = he^{1+k}.$$

Assuming

$$\begin{aligned} k &= a_1 h + a_2 h^2 \\ &= O(h) \end{aligned}$$

Then

$$\begin{aligned} k &= he^{1+O(h)} \\ &= hee^{O(h)} \end{aligned}$$

Using the first order approximation of $e^x$ and evaluating it at $x = O(h)$

$$\begin{aligned} k &= he(1 + O(h)) \\ &= eh + eO(h^2) \end{aligned}$$

Since $k = a_1 h + a_2 h^2$, then $\boxed{a_1 = e}$, and we have

$$\begin{aligned} k &= eh + eh(a_1 h + O(h^2)) \\ &= eh + e^2 h^2 + O(h^3) \end{aligned}$$

So, $\boxed{a_2 = e^2}$

# 2 Problem 2

Assuming $f(u(t),t)$ is Lipschitz

$$\begin{aligned} k_1 &= f(u_n, t_n) \\ k_2 &= f(u_n + \frac{h}{2}k_1, t_n + \frac{h}{2}) \\ \implies u_{n+1} &= u_n + \frac{h}{2}(k_1 + k_2) \end{aligned}$$

## 2.1 Part 1:

$$\begin{aligned} |u_{n+1} - v_{n+1}| &= |(u_n - v_n) + \frac{h}{2}\left(k_1^u - k_1^v + k_2^u - k_2^v\right)| \\ &= |(u_n - v_n) + \frac{h}{2}\left(\underbrace{f(u_n,t_n) - f(v_n,t_n)}_{\leq C|u_n - v_n|} + \underbrace{f(u_n + \frac{h}{2}k_1, t_n + \frac{h}{2}) - f(v_n + \frac{h}{2}k_1, t_n + \frac{h}{2})}_{\leq C|u_n - v_n|}\right)| \end{aligned}$$

## 2.2 Part 2:

## 2.3 Part 3:

# 3 Problem 3

## 3.1 Part 1 (2-step Adams-Bashforth):

## 3.2 Part 2 (1-step Adams-Moulton):

For the following problems solve $y'' - \mu(2 - \exp((y')^2))y' + y = 0$. First convert to system.

# 4 Problem 4

Use $y_0 = 3$, ... $\mu = 0.5$, ...

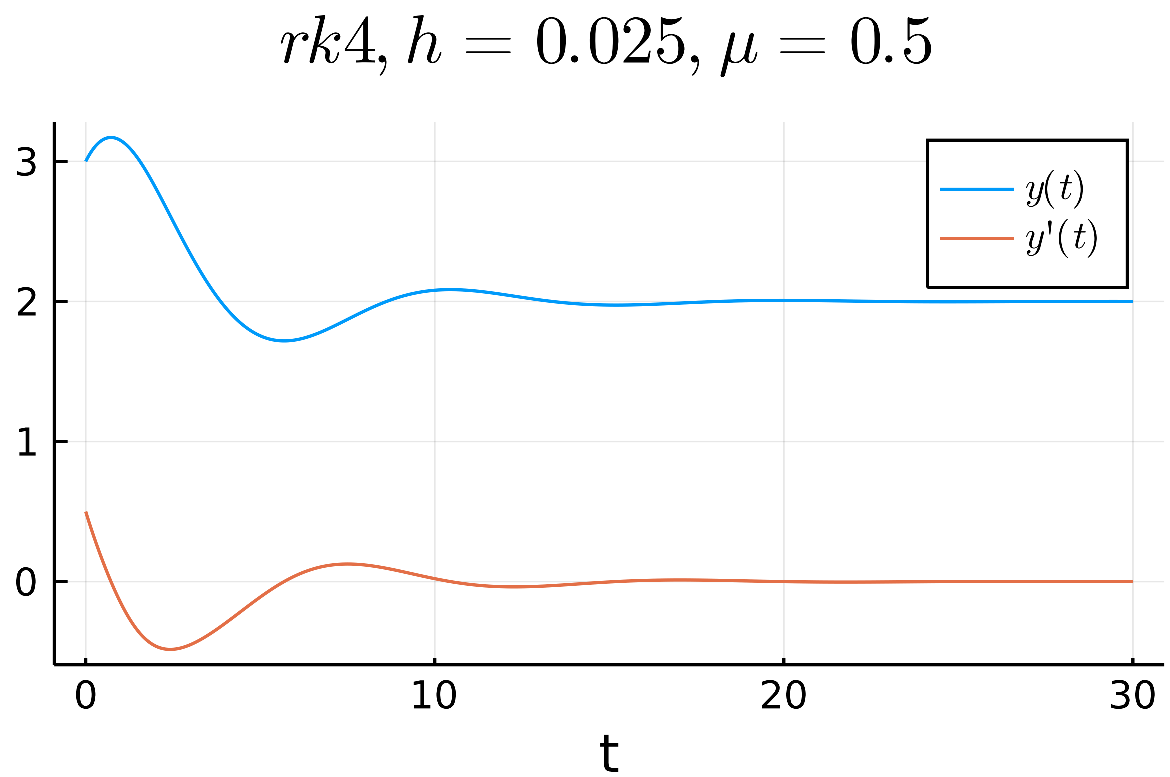
**Plot $y(t)$ vs $t$**

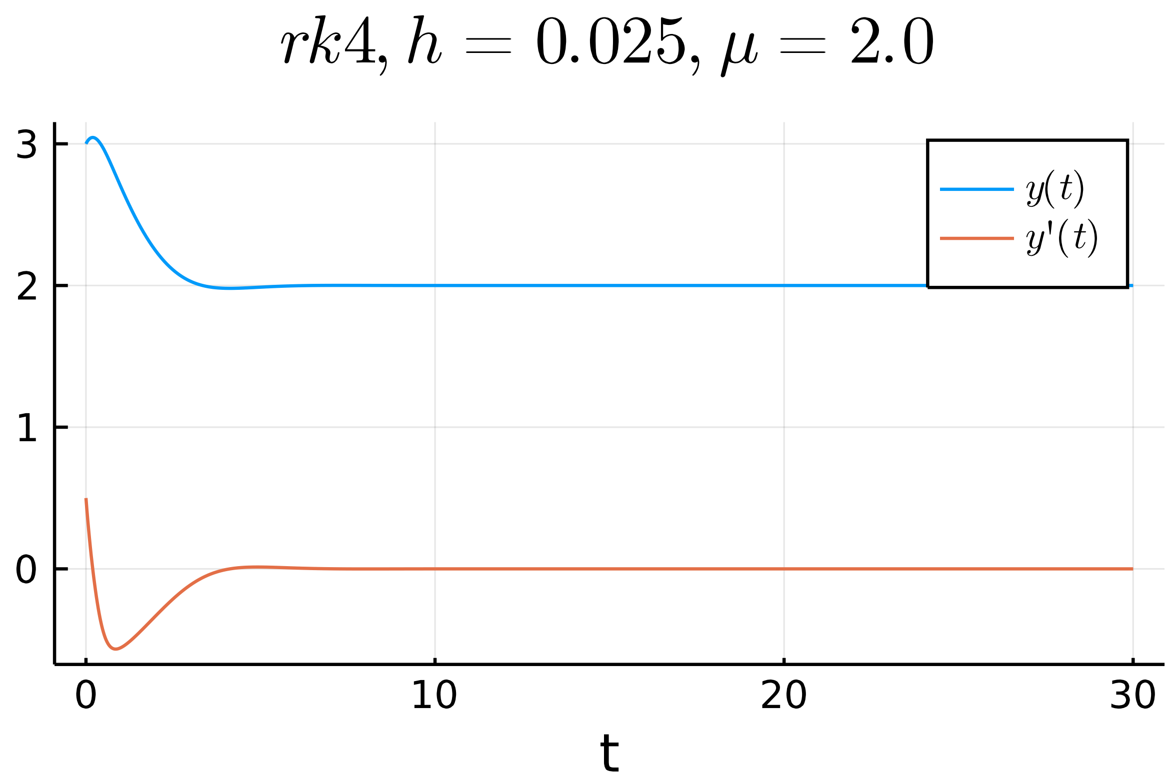
**Including packages**

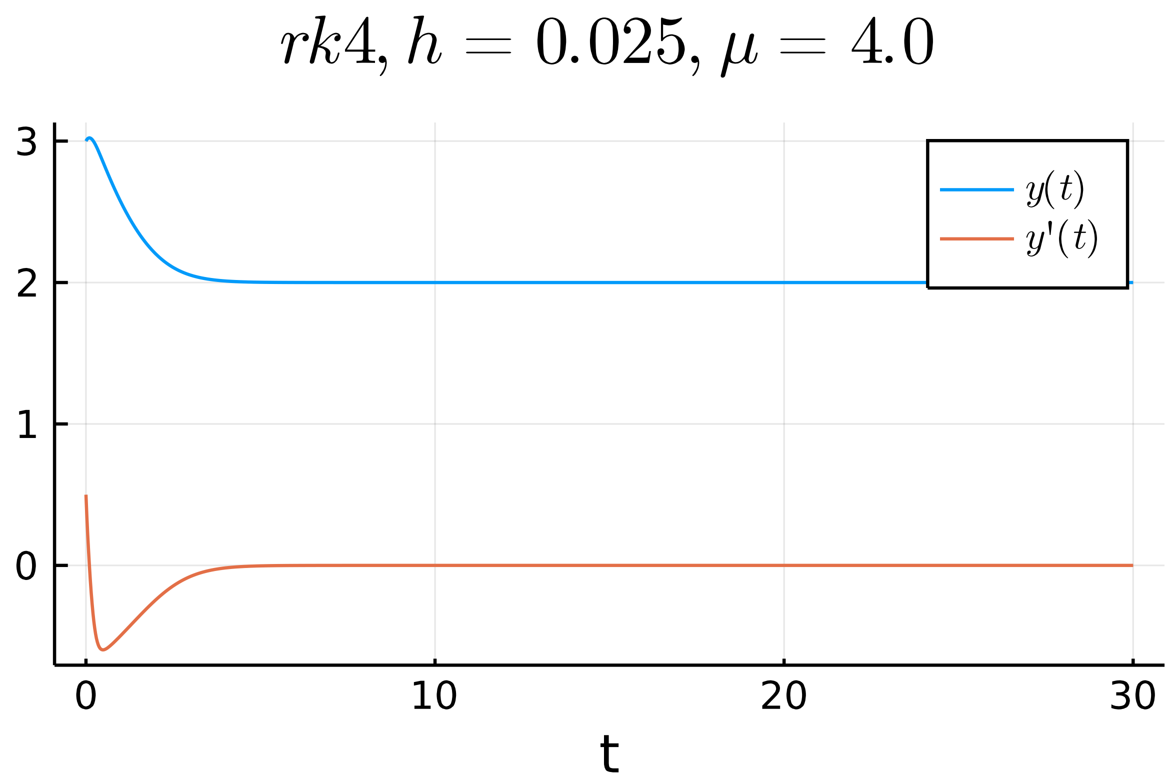
Most functions I've written in the DiffyQ.jl module. I did this in order to make the code, and this report a neater.

```
using Plots
using LaTeXStrings
# Load DiffyQ Module and required functions
using Pkg
Pkg.activate("DiffyQ")
include("code/DiffyQ.jl") # Makes sure the module is run before using it
using .DiffyQ: rk4, fehlberg
```

```julia
# differential equation
function func(u, t, μ)
    return [u[2]; μ*(2-exp(u[2]^2)*u[2] - u[1])]
end

# defining variables
u0 = [3.0; 0.5] # y(0) and y'(0)
h = 0.025; T = 30;
N = Int(T/h);
t = collect(0:N)*h

# run rk4 for various values of μ and plot the results
μs = [0.5, 2, 4]
for μ in μs
    u = rk4(func, N, T, u0, μ)
    p1 = plot(t,u[1,:], label = L"y(t)", thickness_scaling = 1.5)
    p2 = plot!(t,u[2,:], label = L"y'(t)")
    xlabel!("t")
    title!(latexstring("rk4,h=0.025,\\mu=",μ))
    display(p2)

end
```



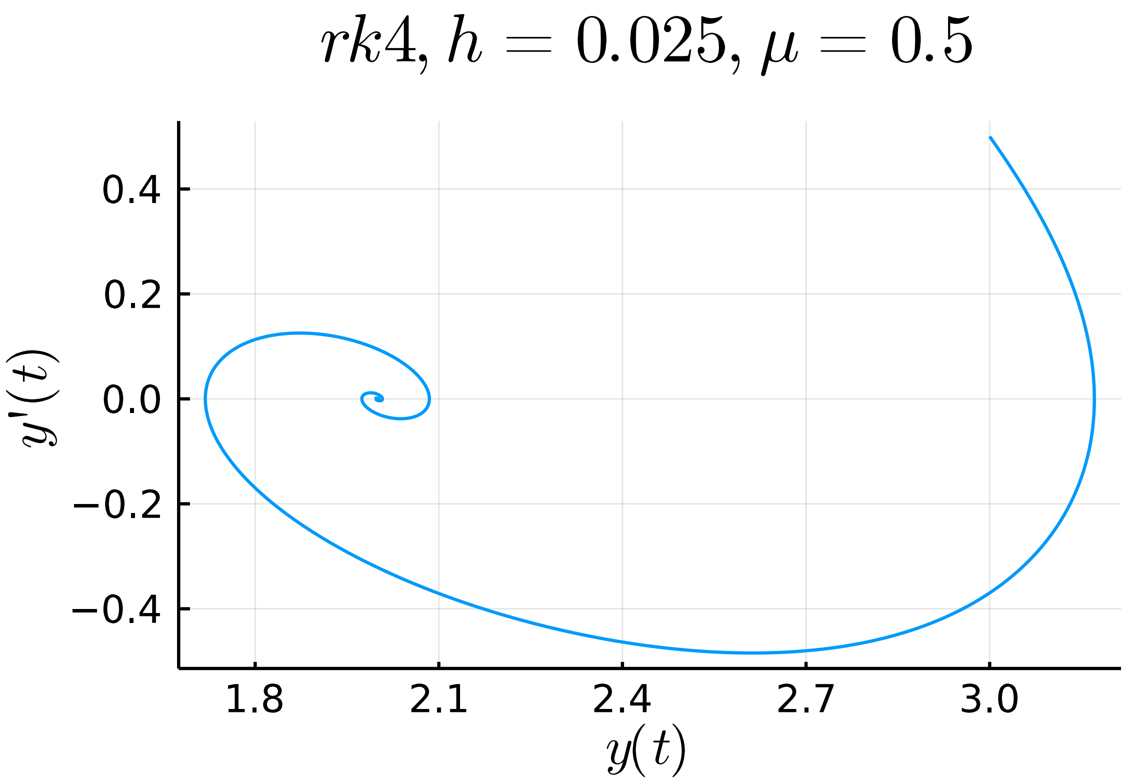


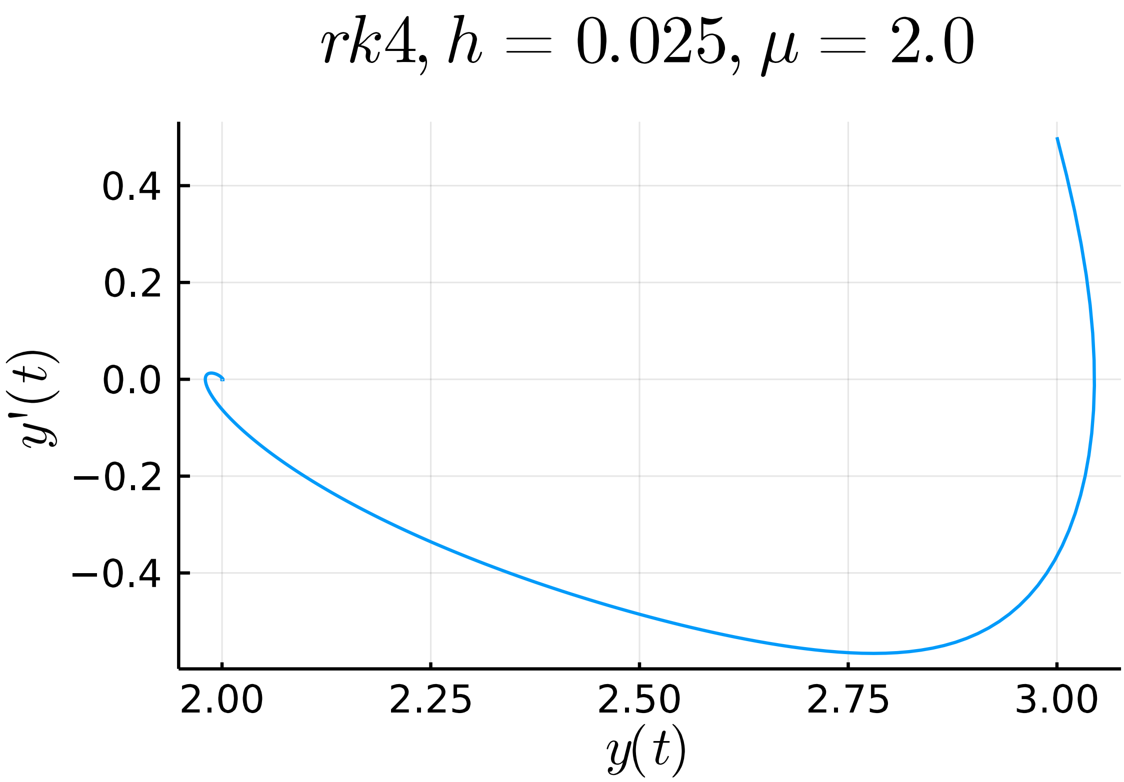



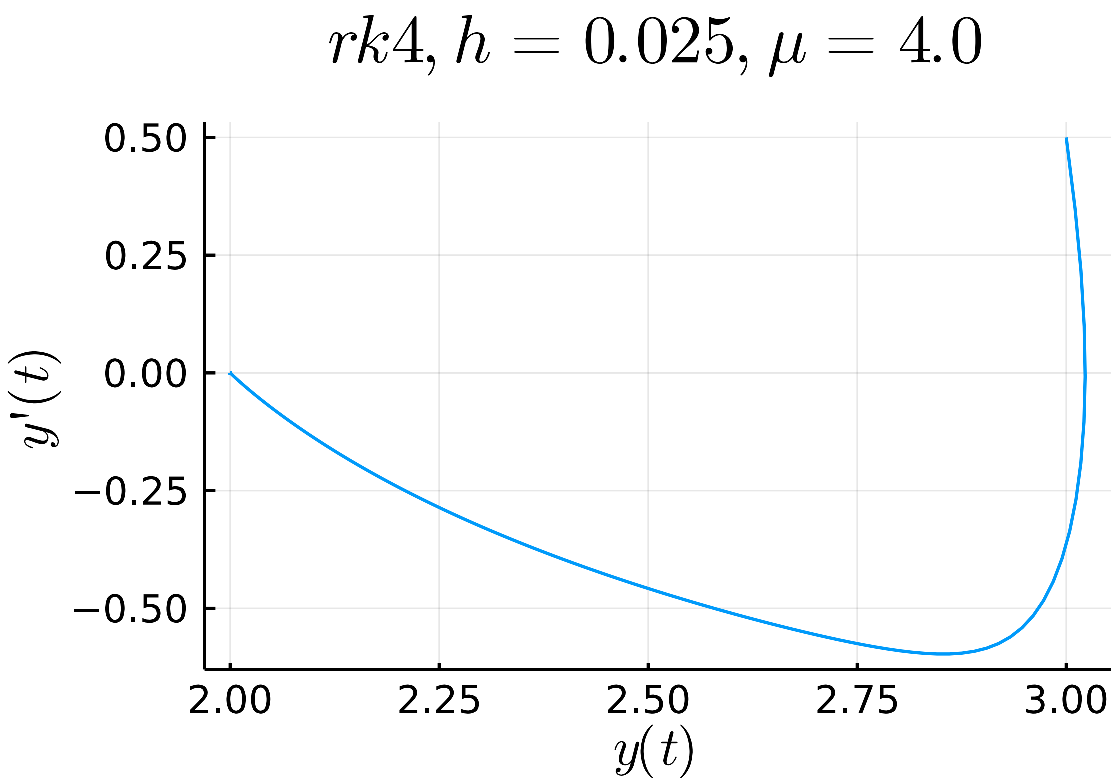

**Plot $y'(t)$ vs $y(t)$**

```
for μ in μs
    u = rk4(func, N, T, u0, μ)
    p3 = plot(u[1,:], u[2,:], thickness_scaling = 1.5, legend = false)
    xlabel!(L"y(t)")
    ylabel!(L"y'(t)")
```

```
    title!(latexstring("rk4,h=0.025,\\mu=",μ))
    display(p3)
end
```



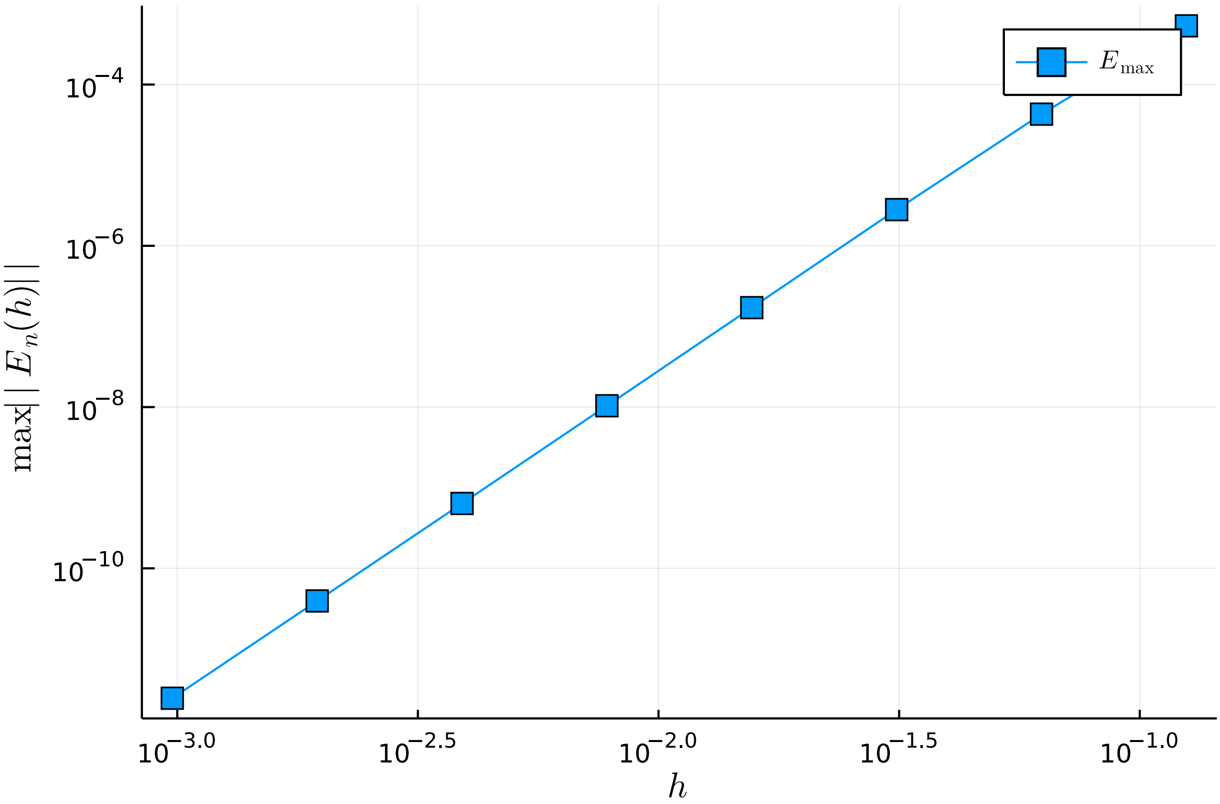



# 5 Problem 5

## 5.1 Part 1

Run simulations for decreasing step sizes and plot $E_max$ vs $h$

```
# defining variables
u0 = [3.0; 0.5] # y(0) and y'(0)
T = 30; μ = 4.0;
N = Int(T/h);
t = collect(0:N)*h

# Part 1
hs = 1 ./(2.0 .^(3:10))
E_max = zeros(size(hs))
for i = 1 : length(hs)
    N = Int(T/hs[i])
    u_nh = rk4(func, N, T, u0, μ)
    u_2nh2 = rk4(func, 2*N, T, u0, μ)

    # subtract every two from u_2nh2
    E_max[i] = maximum(abs.(u_nh[1,1:N] - u_2nh2[1,1:2:2*N])./(1-0.5^4))
end

plot(hs, E_max, label = L"E_{\max}", xaxis = :log, yaxis = :log, marker = (:square,5),
add_marker = true)
xlabel!(L"h")
ylabel!(L"\max||E_n(h)||")
```




Find the value of $h$ where $E_{\max} < 5x10^{-8}$. This occurs at the first index where $E_{\max} < 5x10^{-8}$

```
# Part 2
# Find index where E_max(h) < 5 x 10^-8
index = findfirst(x->x < 5.0*(10.0^(-8)), E_max)
hs[index]
```

```
0.0078125
```

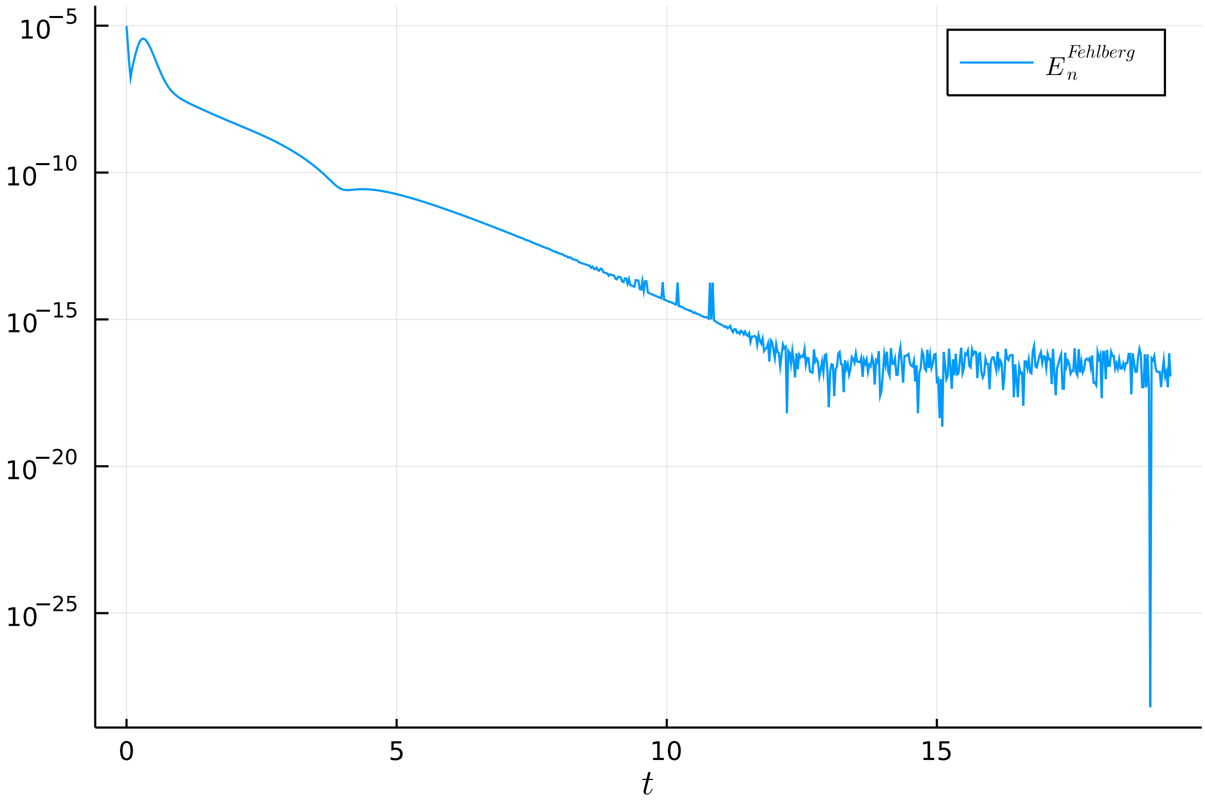
$h = \frac{1}{2^7}$

# 6 Problem 6

## 6.1 Part 1

Plot $E_n^{(Fehlberg)}(h)$ vs $t_n$

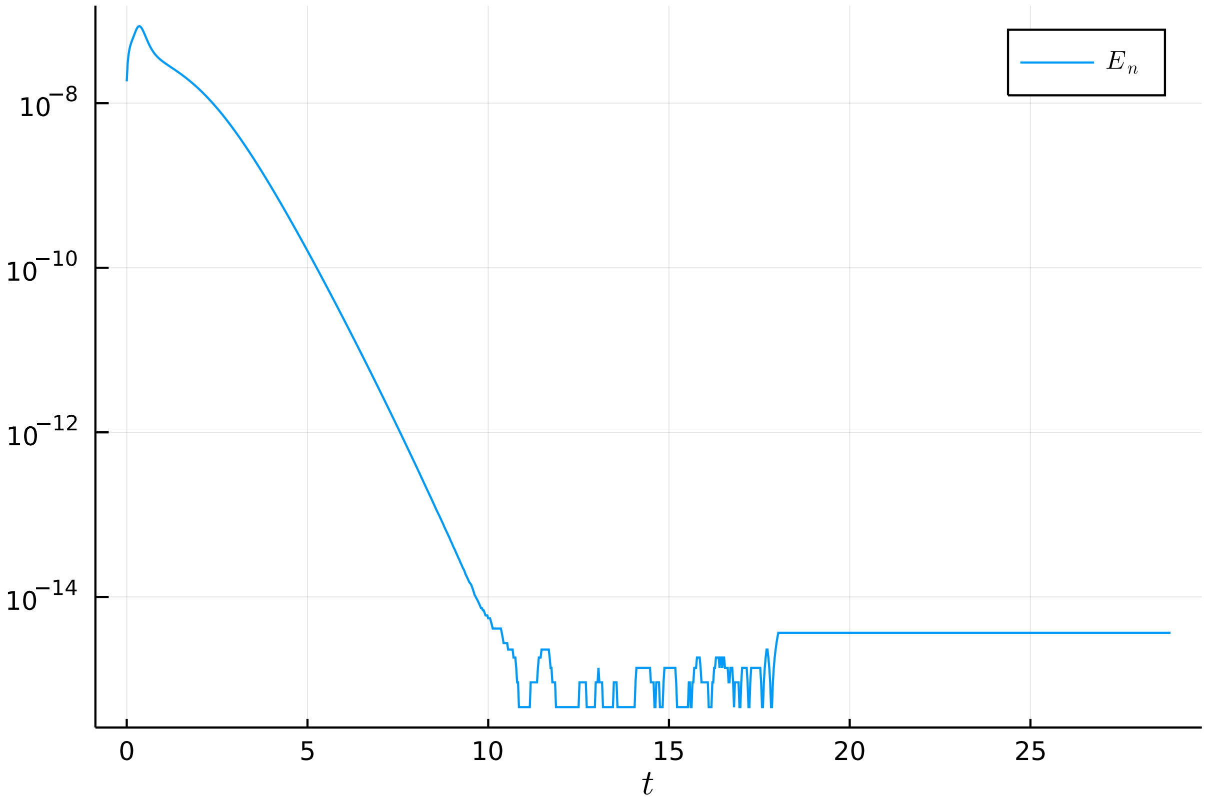
```
# defining variables
u0 = [3.0; 0.5] # y(0) and y'(0)
T = 30; μ = 4.0; h = 0.025;
N = Int(T/h);
t = collect(0:N)*h

# Part 1
(u, E_n) = fehlberg(func, N, T, u0, μ)

# E = E_n[E_n .> 10.0^(-17)]
E = E_n[E_n .!= 0.0]
plot(t[1:length(E)], E, label = L"E^{Fehlberg}_{n}", yaxis = :log)
xlabel!(L"t")
```




### 6.2 Part 2

Plot $E_n$ vs $t_n$

```
# Part 2
(u2, E_n) = fehlberg(func, 2*N, T, u0, μ)

E = abs.(u[1,1:N] - u2[1,1:2:2*N])/(1-0.5^5)
E = E[E .!= 0.0]
plot(t[1:length(E)], E, label = L"E_n", yaxis =:log)
xlabel!(L"t")
```


$10^{-8}$
$10^{-10}$
$10^{-12}$
$10^{-14}$
0
5
10
15
20
25
$t$
$E_n$