In [6]:
```
function euler(t,h)
    y = 3/2 * t^(1/3)
    return y + h*y
end
```

Out[6]: euler (generic function with 1 method)

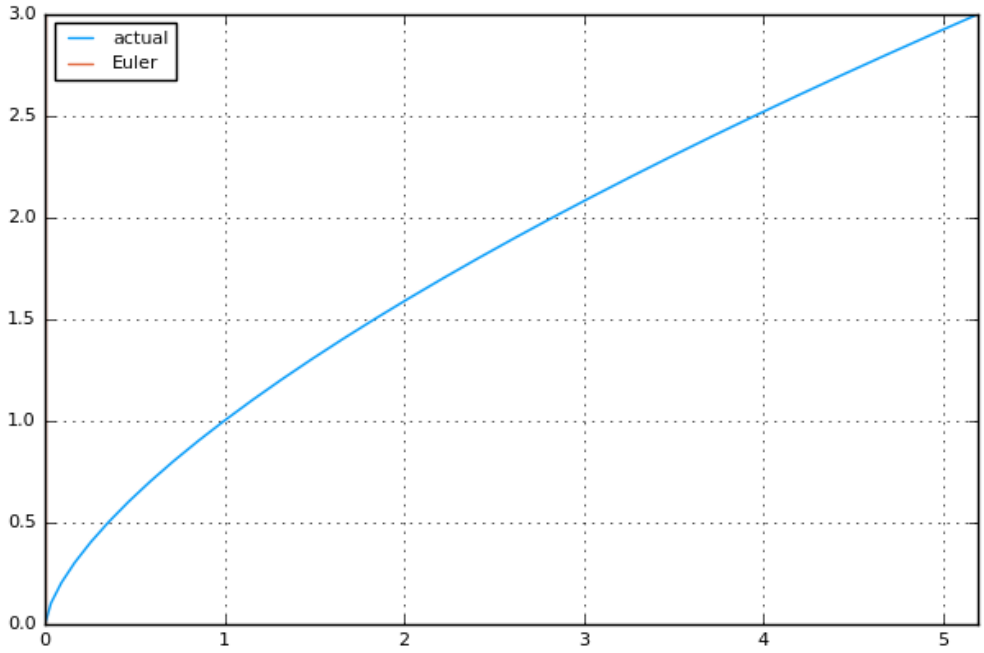
In [7]:
```
h=0.1
ts = 0:h:3
n = length(ts)
y = zeros(n,1)
actual = zeros(n,1)
@printf("Time \t Approx \t Exact \t\t Error Percent \n")
@printf("%d \t %.8f \t %.8f \t %.4f\n", ts[1], y[1], actual[1], abs(actual[
1]-y[1]))
for i=2:n
    y[i] = y[i-1] + euler(y[i-1],h)
    actual[i]=ts[i]^(3/2)
    @printf("%.2f \t %.8f \t %.8f \t %.4f\n", ts[i], y[i], actual[i], abs(a
ctual[i]-y[i])/actual[i]*100)
end
```

Time	Approx	Exact	Error Percent
0	0.00000000	0.00000000	0.0000
0.10	0.00000000	0.03162278	100.0000
0.20	0.00000000	0.08944272	100.0000
0.30	0.00000000	0.16431677	100.0000
0.40	0.00000000	0.25298221	100.0000
0.50	0.00000000	0.35355339	100.0000
0.60	0.00000000	0.46475800	100.0000
0.70	0.00000000	0.58566202	100.0000
0.80	0.00000000	0.71554175	100.0000
0.90	0.00000000	0.85381497	100.0000
1.00	0.00000000	1.00000000	100.0000
1.10	0.00000000	1.15368973	100.0000
1.20	0.00000000	1.31453414	100.0000
1.30	0.00000000	1.48222805	100.0000
1.40	0.00000000	1.65650234	100.0000
1.50	0.00000000	1.83711731	100.0000
1.60	0.00000000	2.02385770	100.0000
1.70	0.00000000	2.21652882	100.0000
1.80	0.00000000	2.41495342	100.0000
1.90	0.00000000	2.61896926	100.0000
2.00	0.00000000	2.82842712	100.0000
2.10	0.00000000	3.04318912	100.0000
2.20	0.00000000	3.26312733	100.0000
2.30	0.00000000	3.48812270	100.0000
2.40	0.00000000	3.71806401	100.0000
2.50	0.00000000	3.95284708	100.0000
2.60	0.00000000	4.19237403	100.0000
2.70	0.00000000	4.43655272	100.0000
2.80	0.00000000	4.68529615	100.0000
2.90	0.00000000	4.93852205	100.0000
3.00	0.00000000	5.19615242	100.0000

In [3]:
```
using Plots
plot(actual,ts,label="actual")
plot!(y,ts,label="Euler")
```

```
[Plots.jl] Initializing backend: pyplot
WARNING: using Plots.h in module Main conflicts with an existing identifier
.
```

Out[3]:



In [22]:
```
savefig("problem4A.png")
```

In [ ]:
```
```