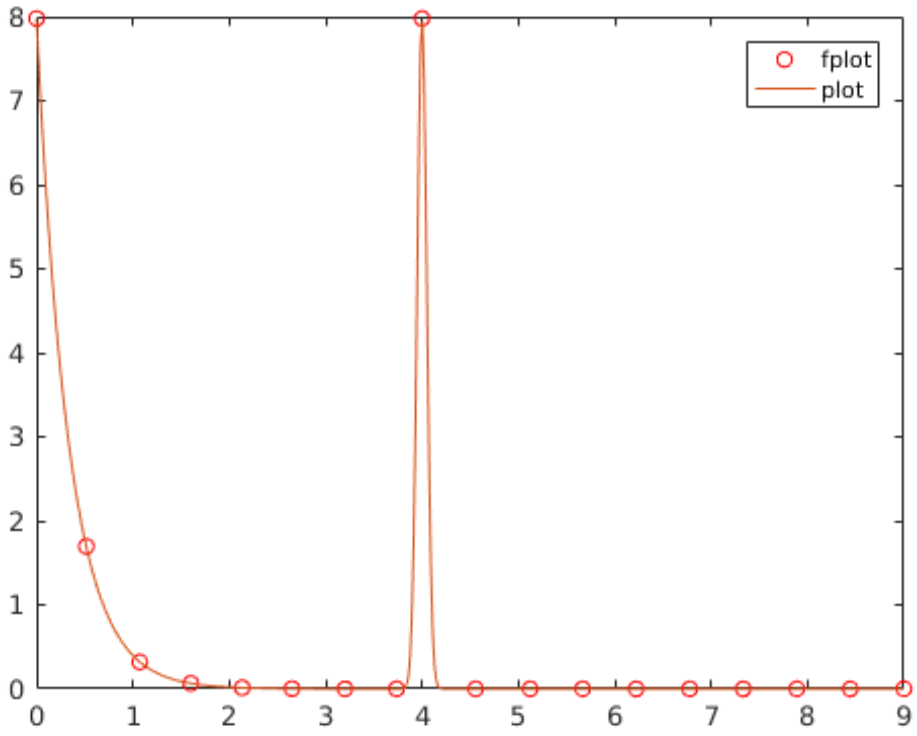
Table of Contents


```
%Cory Wolfe
sigma = .05;
a = 1/(sigma*sqrt(2*pi));
f = @(x) a*(exp(-3*x)+exp(-(x-4).^2/(2*sigma^2)));
x = linspace(0,9,500);y = f(x);
figure(1),clf
fplot(f,[0,9],'ro')
hold on
plot(x,y)
legend('fplot','plot')
```



Applying Simpson's

```
s2 = simpson13(f,0,9,2)
```

```
s98 = simpson13(f,0,9,98)
s100 = simpson13(f,0,9,100)
s1000 = simpson13(f,0,9,1000)

s40_1 = simpson13(f,0,3,40)
s42_1 = simpson13(f,0,3,42)
s40_2 = simpson13(f,3,6,40)
s42_2 = simpson13(f,3,6,42)
s40_3 = simpson13(f,6,9,40)
s42_3 = simpson13(f,6,9,42)

s120 = simpson13(f,0,9,40*3)
s126 = simpson13(f,0,9,42*3)


s2 =

   11.9683


s98 =

    3.6275


s100 =

    3.6302


s1000 =

    3.6596


s40_1 =

    2.6593


s42_1 =

    2.6593


s40_2 =

    1.0374


s42_2 =

    0.9411
```

```
s40_3 =

   4.0501e-08


s42_3 =

   4.0501e-08


s120 =

    3.6967


s126 =

    3.6004
```

Adaptive Quadrature Mehtods

```
I1 = quadadapt(f,0,9)
I1_coarse = quadadapt(f,0,9,.1)
I2 = quad(f,0,9)
I2_coarse = quad(f,0,9,.1)
```

```
I1 =

    3.6596


I1_coarse =

    3.6560


I2 =

    3.6596


I2_coarse =

    2.7235
```

Timeing Methods

```
tic,I3 = simpson13(f,0,9,1000),toc
```

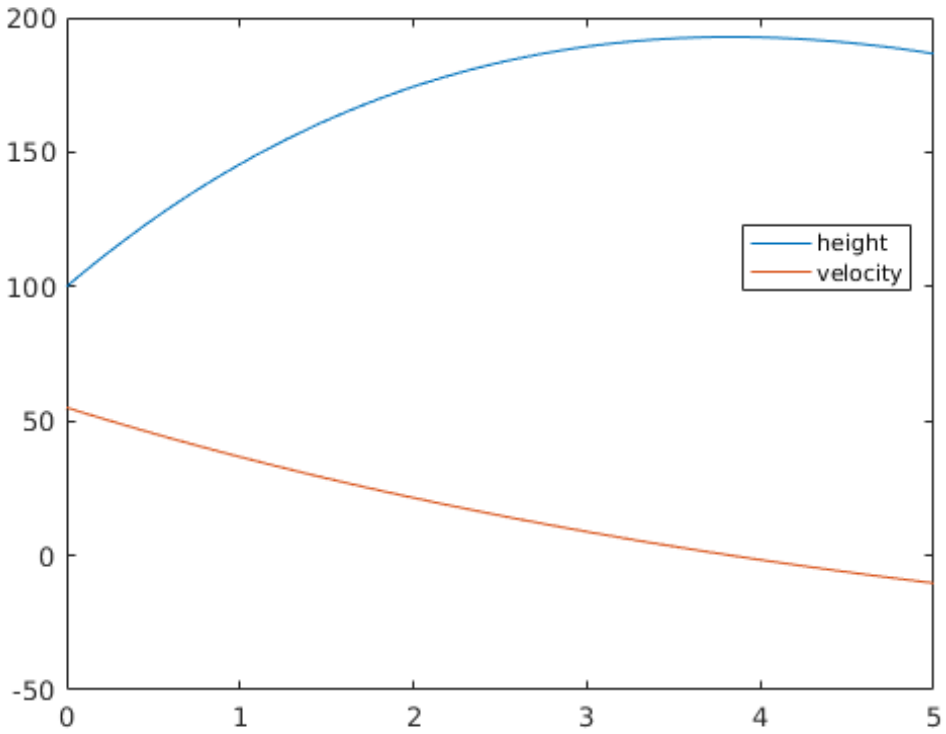
```
tic,I4 = quad(f,0,9),toc
```

```
I3 =

    3.6596

Elapsed time is 0.000615 seconds.

I4 =

    3.6596

Elapsed time is 0.000801 seconds.
```

Physical Problem

```
t = linspace(0,5);
z = myheight(t,100,55,80,15,9.81);
dz = velocity(t,100,55,80,15,9.81);
figure(2),clf
plot(t,z,t,dz)
legend('height','velocity','Location','best')
change_z = quadadapt(@velocity,0,5,[],100,55,80,15,9.81)
change_z2 = quad(@velocity,0,5,[],[],100,55,80,15,9.81)
myheight(5,100,55,80,15,9.81)-myheight(0,100,55,80,15,9.81)
```

```
change_z =

   86.6287


change_z2 =

   86.6287


ans =

   86.6287
```



Published with MATLAB® R2016b