

b) clear all; clc;

dx = .5;

t = 0:dx:2;

y(1) = 1;

f = @(t,y) (1+2\*t)\*sqrt(y);

for n = 1:length(t)

if n < length(t)

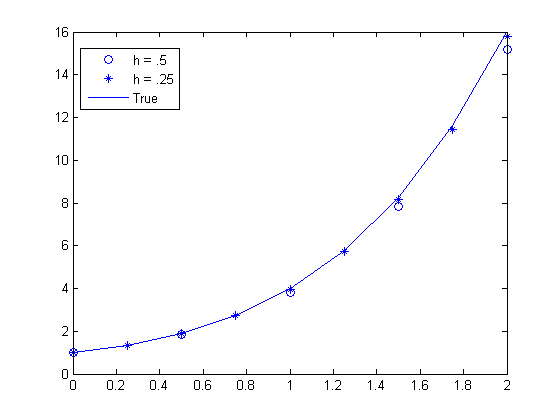
k1 = f(t(n), y(n));

k2 = f(t(n) + dx/2, y(n) + k1\*dx/2);

y(n+1) = y(n) + k2\*dx;

end

end



plot(t,y,'o');

hold on;

dx = .25;

t = 0:dx:2;

for n = 1:length(t)

if n < length(t)

k1 = f(t(n), y(n));

k2 = f(t(n) + dx/2, y(n) + k1\*dx/2);

y(n+1) = y(n) + k2\*dx;

end

end

plot(t,y,'\*');

f = @(t) (t./2 + (t.^2)./2 + 1).^2;

plot(t, f(t));

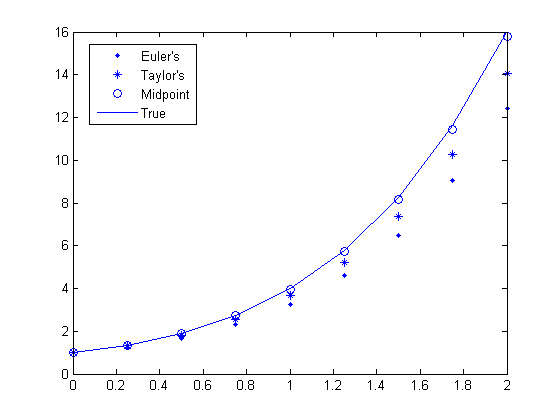
c) f = @(t,y) (1+2\*t)\*sqrt(y);

fprime = matlabFunction(diff(f,sym('t')));

dt = .25;

t=0:dt:2;

y(1)=1;



for n=1:length(t)-1

y(n+1)=y(n)+dt\*f(t(n), y(n));

end

plot(t,y,'.');

hold on;

for n=1:length(t)-1

y(n+1) = y(n) + dt\*f(t(n),y(n)) + (dt^2)/2 \* fprime(y(n));

end

plot(t,y,'\*');

for n = 1:length(t)

if n < length(t)

k1 = f(t(n), y(n));

k2 = f(t(n) + dx/2, y(n) + k1\*dx/2);

y(n+1) = y(n) + k2\*dx;

end

end

plot(t,y,'o');

f = @(t) (t./2 + (t.^2)./2 + 1).^2;

plot(t, f(t));

It seems that the Midpoint methods seems to be the most accurate out of these three with the given step size. Euler’s method looks like it was the least accurate out of the bunch. Since both Taylor’s and Euler’s method required only one line of code to be repeated, they may be more efficient than the Midpoint method.

2.

clear all; clc;

a = @(y) (-9.81\*(6.37e6)^2)/(6.37e6 + y)^2;

k = 1400;

v = matlabFunction(int(a, sym('t')) + k);

dt = 10;

t=0:dt:300;

y(1) = 0;

for n=1:length(t)-1

y(n+1)=y(n)+dt\*v(t(n),y(n));

end

plot(t,y,'.');

hold on;

dt = 1;

t=0:dt:300;

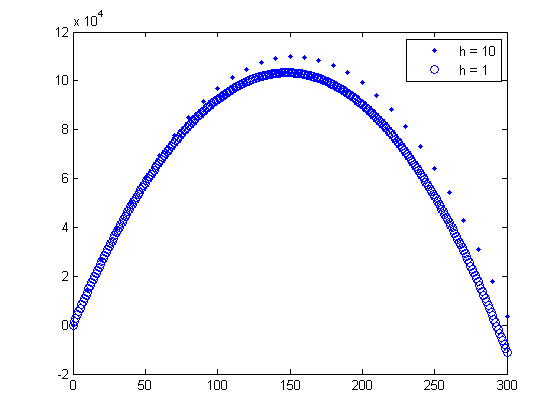
y(1) = 0;

for n=1:length(t)-1

y(n+1)=y(n)+dt\*v(t(n),y(n));

end

plot(t,y,'o');



3.

clear all; clc;

dt = .1;

t = 0:dt:20;

a1 = @(x1,x2) -3000\*x1/12000 + 2400\*(x2-x1)/12000;

a2 = @(x1,x2,x3) 2400\*(x1-x2)/10000 + 1800\*(x3-x2)/10000;

a3 = @(x2,x3) 1800\*(x2-x3)/8000;

v1 = matlabFunction(int(a1, sym('t')));

v2 = matlabFunction(int(a2, sym('t')));

v3 = matlabFunction(int(a3, sym('t')));

x1(1) = 0;

x2(1) = 0;

x3(1) = 0;

v1Vec(1) = 1;

v2Vec(1) = 0;

v3Vec(1) = 0;

for n = 1:length(t)-1

v1Vec(n+1) = v1Vec(n)+dt\*a1(x1(n),x2(n));

v2Vec(n+1) = v2Vec(n)+dt\*a2(x1(n),x2(n),x3(n));

v3Vec(n+1) = v3Vec(n)+dt\*a3(x2(n),x3(n));

x1(n+1) = x1(n) + dt\*(v1(t(n),x1(n),x2(n)) + v1Vec(n));

x2(n+1) = x2(n) + dt\*(v2(t(n),x1(n),x2(n),x3(n)) + v2Vec(n));

x3(n+1) = x3(n) + dt\*(v3(t(n),x2(n),x3(n)) + v3Vec(n));

end

figure('Name', 'v vs t');

plot(t,v1Vec, t,v2Vec, t,v3Vec);

figure('Name', 'x vs t');

plot(t,x1, t,x2, t,x3);

