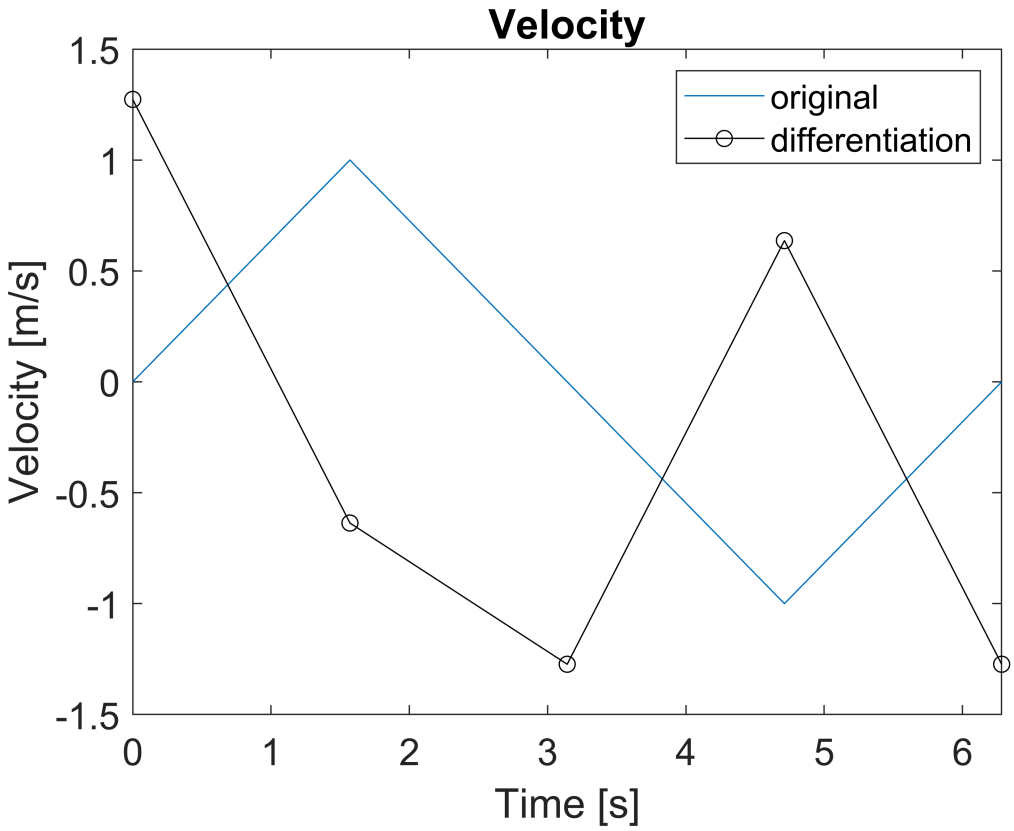
Daniel Hondal

HW 8C: Problem 2

Part A: 2nd order Accuracy - 5 samples

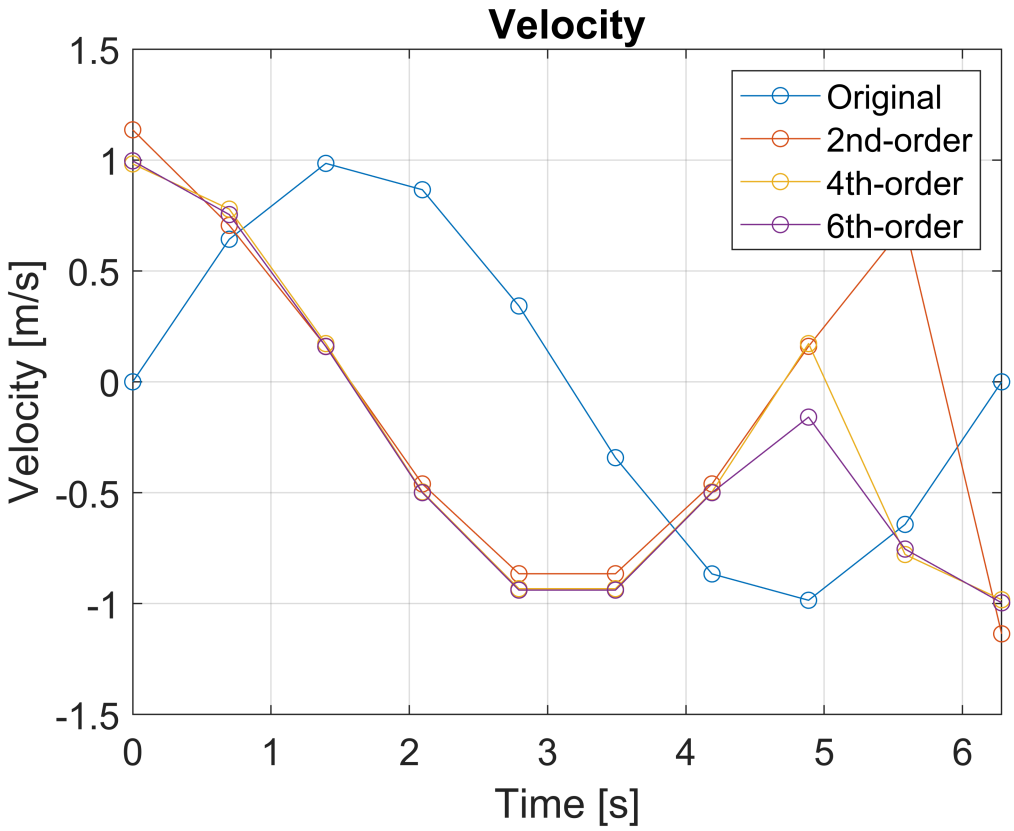
```
% Central Diff.
ta = linspace(0,2*pi,5);
dt = mean(diff(ta));
xa = sin(ta);
dxdt_a = zeros(length(xa),1); % pre-allocate acceleration values
coeff2 = [-3/2 2 -1/2];
ccd2 = [-1/2 1/2];

for k=1:length(xa)
    if k < 4 % forward when to few points to left
        dxdt_a(k) = coeff2*[xa(k)  xa(k+1)  xa(k+2)]'/dt;
    elseif k> 3 % backward when too few points to right
        dxdt_a(k) = coeff2*[xa(k)  xa(k-1)  xa(k-2)]'/dt;
    else % central
        dxdt_a(k) = ccd2*[xa(k-1)  xa(k+1)]'/dt;
    end
end
%%
plot(ta,xa,ta,dxdt_a,'k-o');
title('Velocity')
xlabel('Time [s]','fontsize',14);
ylabel('Velocity [m/s]','fontsize',14);
set(gca,'fontsize',14);
legend('original','differentiation')
```



Part B: 2nd order, 4th order, 6th order Accuracy - 10 samples

```
% Central Diff.
tb = linspace(0,2*pi,10);
dt = mean(diff(tb));
xb = sin(tb);
dxdt_b2 = zeros(length(xb),1); % pre-allocate acceleration values
dxdt_b4 = zeros(length(xb),1); % pre-allocate acceleration values
dxdt_b6 = zeros(length(xb),1); % pre-allocate acceleration values

% 2nd order
coeff2 = [-3/2 2 -1/2];
ccd2 = [-1/2 1/2];
for k=1:length(xb)
    if k < 2 % forward when to few points to left
        dxdt_b2(k) = coeff2*[xb(k)  xb(k+1)  xb(k+2)]'/dt;
    elseif k> 9 % backward when too few points to right
        dxdt_b2(k) = coeff2*[xb(k)  xb(k-1)  xb(k-2)]'/dt;
    else % central
        dxdt_b2(k) = ccd2*[xb(k-1)  xb(k+1)]'/dt;
    end
end

% 4th order
coeff4 = [-25/12 4 -3 4/3 -1/4];
ccd4 = [1/12 -2/3 2/3 -1/12];
```

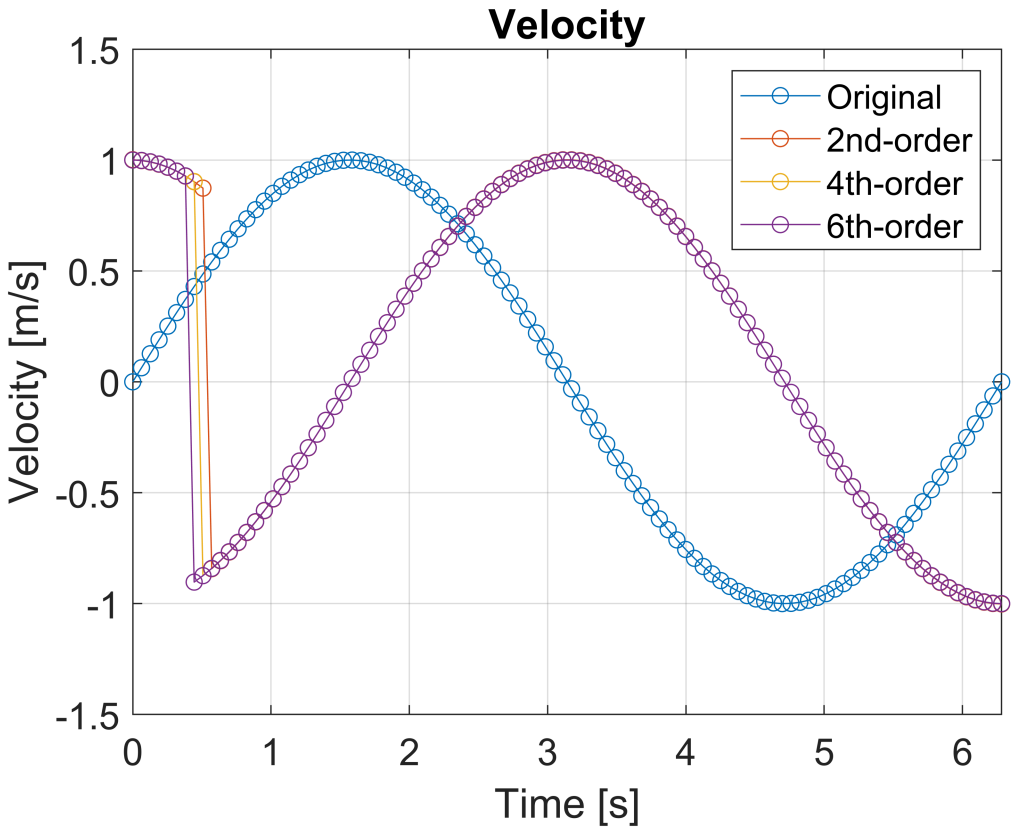
```
for k=1:length(xb)
    if k < 3 % forward when to few points to left
        dxdt_b4(k) = coeff4*[xb(k)  xb(k+1)  xb(k+2) xb(k+3) xb(k+4)]'/dt;
    elseif k> 8 % backward when too few points to right
        dxdt_b4(k) = coeff4*[xb(k)  xb(k-1)  xb(k-2) xb(k-3) xb(k-4)]'/dt;
    else % central
        dxdt_b4(k) = ccd4*[xb(k-2)  xb(k-1)  xb(k+1) xb(k+2)]'/dt;
    end
end

% 6th order
coeff6 = [-49/20 6 -15/2 20/3 -15/4 6/5 -1/6];
ccd6 = [-1/60 3/20 -3/4 3/4 -3/20 1/60];
for k=1:length(xb)
    if k < 4 % forward when to few points to left
        dxdt_b6(k) = coeff6*[xb(k)  xb(k+1)  xb(k+2) xb(k+3) xb(k+4) xb(k+5) xb(k+6)]'/dt;
    elseif k> 7 % backward when too few points to right
        dxdt_b6(k) = coeff6*[xb(k)  xb(k-1)  xb(k-2) xb(k-3) xb(k-4) xb(k-5) xb(k-6)]'/dt;
    else % central
        dxdt_b6(k) = ccd6*[xb(k-3)  xb(k-2)  xb(k-1) xb(k+1) xb(k+2) xb(k+3)]'/dt;
    end
end
%%

p = plot(tb,xb,tb,dxdt_b2,tb,dxdt_b4,tb,dxdt_b6);
p(1).Marker = 'o';
p(2).Marker = 'o';
p(3).Marker = 'o';
p(4).Marker = 'o';
grid on;
title('Velocity')
xlabel('Time [s]','fontsize',14);
ylabel('Velocity [m/s]','fontsize',14);
set(gca,'fontsize',14);
legend('Original','2nd-order','4th-order','6th-order','location','northeast');
```



Part C: 2nd order, 4th order, 6th order Accuracy - 100 samples

```
% Central Diff.
tc = linspace(0,2*pi,100);
dt = mean(diff(tc));
xc = sin(tc);
dxdt_c2 = zeros(length(xc),1); % pre-allocate acceleration values
dxdt_c4 = zeros(length(xc),1); % pre-allocate acceleration values
dxdt_c6 = zeros(length(xc),1); % pre-allocate acceleration values

% 2nd order
coeff2 = [-3/2 2 -1/2];
ccd2 = [-1/2 1/2];
for k=1:length(xc)
    if k < 2 % forward when to few points to left
        dxdt_c2(k) = coeff2*[xc(k)  xc(k+1)  xc(k+2)]'/dt;
    elseif k> 9 % backward when too few points to right
        dxdt_c2(k) = coeff2*[xc(k)  xc(k-1)  xc(k-2)]'/dt;
    else % central
        dxdt_c2(k) = ccd2*[xc(k-1)  xc(k+1)]'/dt;
    end
end

% 4th order
coeff4 = [-25/12 4 -3 4/3 -1/4];
ccd4 = [1/12 -2/3 2/3 -1/12];
```

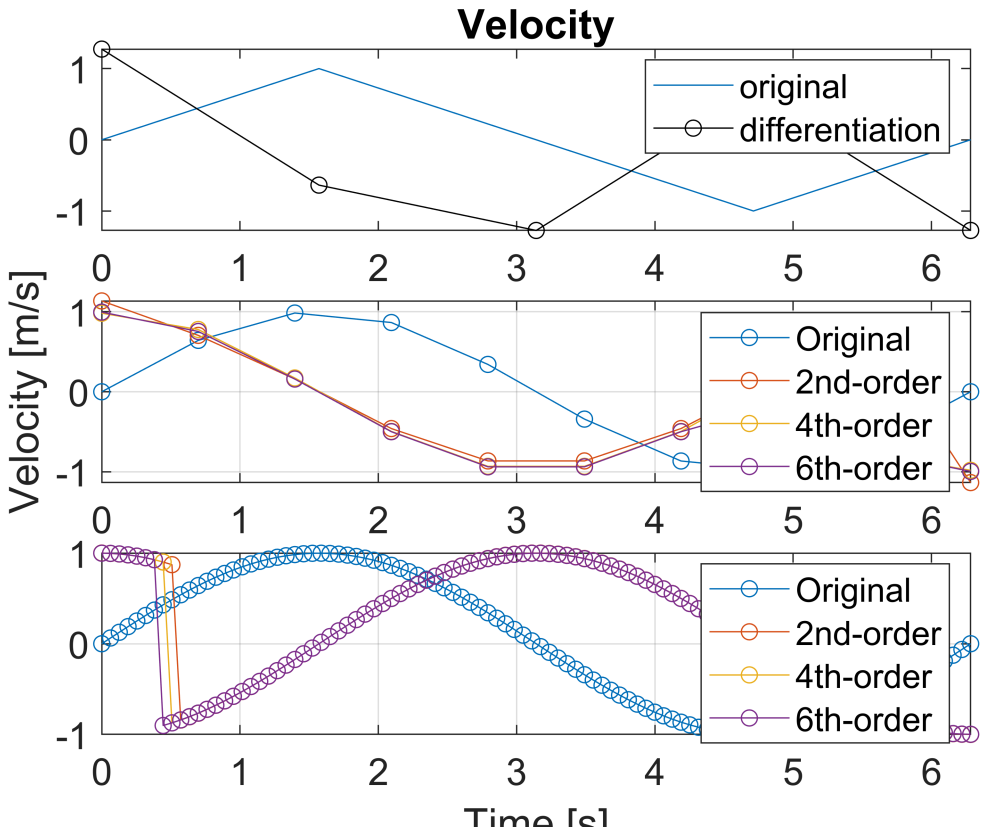
```
for k=1:length(xc)
    if k < 3 % forward when to few points to left
        dxdt_c4(k) = coeff4*[xc(k)  xc(k+1)  xc(k+2) xc(k+3) xc(k+4)]'/dt;
    elseif k> 8 % backward when too few points to right
        dxdt_c4(k) = coeff4*[xc(k)  xc(k-1)  xc(k-2) xc(k-3) xc(k-4)]'/dt;
    else % central
        dxdt_c4(k) = ccd4*[xc(k-2)  xc(k-1)  xc(k+1) xc(k+2)]'/dt;
    end
end

% 6th order
coeff6 = [-49/20 6 -15/2 20/3 -15/4 6/5 -1/6];
ccd6 = [-1/60 3/20 -3/4 3/4 -3/20 1/60];
for k=1:length(xc)
    if k < 4 % forward when to few points to left
        dxdt_c6(k) = coeff6*[xc(k)  xc(k+1)  xc(k+2) xc(k+3) xc(k+4) xc(k+5) xc(k+6)]'/dt;
    elseif k> 7 % backward when too few points to right
        dxdt_c6(k) = coeff6*[xc(k)  xc(k-1)  xc(k-2) xc(k-3) xc(k-4) xc(k-5) xc(k-6)]'/dt;
    else % central
        dxdt_c6(k) = ccd6*[xc(k-3)  xc(k-2)  xc(k-1) xc(k+1) xc(k+2) xc(k+3)]'/dt;
    end
end
%%

p = plot(tc,xc,tc,dxdt_c2,tc,dxdt_c4,tc,dxdt_c6);
p(1).Marker = 'o';
p(2).Marker = 'o';
p(3).Marker = 'o';
p(4).Marker = 'o';
grid on;
title('Velocity')
xlabel('Time [s]','fontsize',14);
ylabel('Velocity [m/s]','fontsize',14);
set(gca,'fontsize',14);
legend('Original','2nd-order','4th-order','6th-order','location','northeast');
```



Part D: Comparison

```
subplot(3,2,[1 2])
plot(ta,xa,ta,dxdt_a,'k-o');
title('Velocity')
set(gca,'fontsize',14);
legend('original','differentiation')
subplot(3,2,[3 4])
p = plot(tb,xb,tb,dxdt_b2,tb,dxdt_b4,tb,dxdt_b6);
p(1).Marker = 'o';
p(2).Marker = 'o';
p(3).Marker = 'o';
p(4).Marker = 'o';
grid on;
ylabel('Velocity [m/s]','fontsize',14);
set(gca,'fontsize',14);
legend('Original','2nd-order','4th-order','6th-order','location','northeast');
subplot(3,2,[5 6])
p = plot(tc,xc,tc,dxdt_c2,tc,dxdt_c4,tc,dxdt_c6);
p(1).Marker = 'o';
p(2).Marker = 'o';
p(3).Marker = 'o';
p(4).Marker = 'o';
grid on;
```

```
set(gca,'fontsize',14);
xlabel('Time [s]','fontsize',14);
legend('Original','2nd-order','4th-order','6th-order','location','northeast');
```



The increase in samples results in a smoother plot that looks more closer to a better approximation.