```
%The following code used a taylor series approximation on the function
%f(x)=25x^3-6x^2+7x-88. The taylor series will approximate the value of the
%function at the point x=3 usinng the starting point of x=1. It will also
%calculate the true error after each taylor series order.

%Defining the variables used in the code.
fx=[25 -6 7 -88];   %The original function put in a matrix.
first_deriv=[75 -12 7]; %The first derivative of the function.
second_deriv=[150 -12]; %The second derivative of the function.
third_deriv=[150];  %The third derivative of the function.
dx=2;   %The step size, 3-1.

%The taylor series orders, from 0 order to the 3rd order.
zero_order=polyval(fx,1);   %The 0th order for taylor series.
first_order=zero_order+(1/factorial(1)*polyval(first_deriv,1))*dx;  %The first order for taylor series.
second_order=first_order+(1/factorial(2)*polyval(second_deriv,1))*dx^2; %2nd order taylor series.
third_order=second_order+(1/factorial(3)*(polyval(third_deriv,1)))*dx^3;    %3rd order taylor series.

%Calculating the true error after each taylor series order.
t_error=polyval(fx,3);  %The true error of the function at x=3.
zo_error=(t_error-zero_order)/t_error;  %The true error after 0 order taylor series.
fo_error=(t_error-first_order)/t_error; %The true error after 1st order taylor series.
so_error=(t_error-second_order)/t_error;    %The true error after 2nd order taylor series.
to_error=(t_error-third_order)/t_error; %The true error after 3rd order taylor series.

%Displays the true value of the function at x=3, as well as dispays the
%value calculated after each taylor series run.
fprintf('The true for fx at x=3 is %.3f\n',t_error)
fprintf('The value calculated for the 0th order of Taylor series is %3f\n',zero_order)
fprintf('The value calculated for the first order of Taylor series is %3f\n',first_order)
fprintf('The value calculated for the second order of Taylor series is %3f\n',second_order)
fprintf('The value calculated for the third order of Taylor series is %3f\n',third_order)

%Displays the true relative error of the function after each taylor series
%run.
fprintf('The true percent relative error after the 0th order is %3f\n',zo_error)
fprintf('The true percent relative error after the first order is %3f\n',fo_error)
fprintf('The true percent relative error after the second order is %3f\n',so_error)
fprintf('The true percent relative error after the third order is %3f\n',to_error)
%%
fx=[25 -6 7 -88];   %fx is the same function above
h=0.25;     %h is the step size between xi,xi+1, and xi-1

%Forward, backward, and centered approximations of the derivative of fx,
%using fx and the point x=2
forward_approx=(polyval(fx,(2+h))-polyval(fx,2))/h;
```

```
backward_approx=(polyval(fx,2)-polyval(fx,(2-h)))/h;
center_approx=(polyval(fx,(2+h))-polyval(fx,(2-h)))/(2*h);

%Display the approximations in the command window
fprintf('The forward approximation of the derivative of fx at x=2 is %3f\n',
forward_approx)
fprintf('The backward approximation of the derivative of fx at x=2 is %3f\n',
backward_approx)
fprintf('The centered approximation of the derivative of fx at x=2 is %3f\n',
center_approx)
%The centered approximation is much more accurate than the forward or
%backward, because the center approximation uses points above and below the
%desired derivative. Forward and backward approximations use only points
%above or below the desired derivative.
```