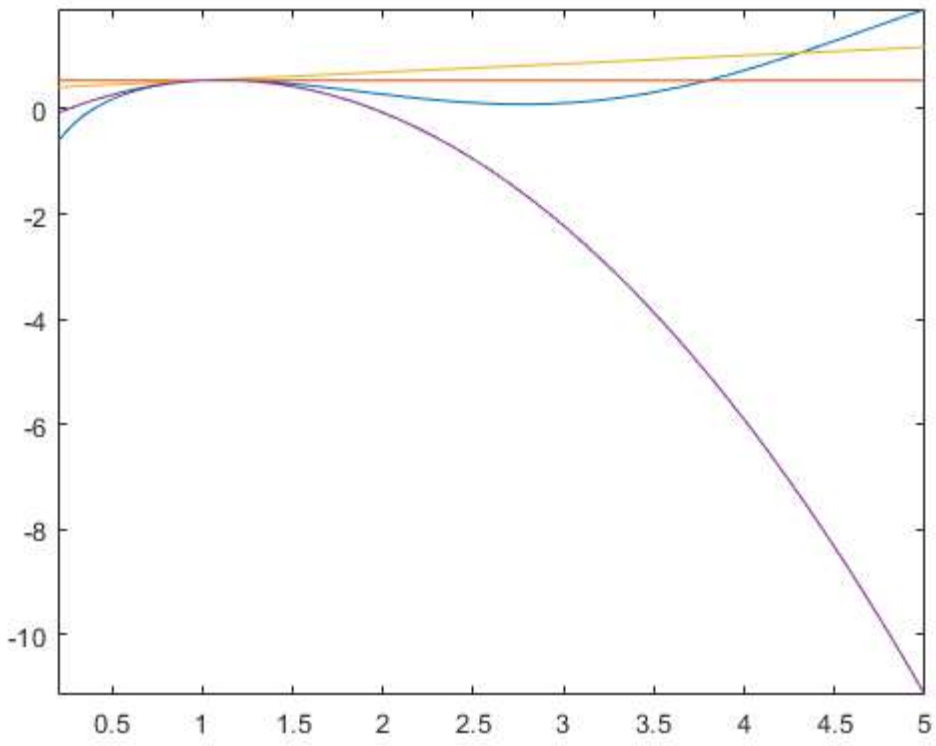
```
function taylor
f =  @(x) cos(x) + log(x);
p0 = @(x) cos(1);
p1 = @(x) cos(1) + (-sin(1)+1)*(x-1);
p2 = @(x) cos(1) + (-sin(1)+1)*(x-1) + ((-cos(1)-1)/2)*(x-1)^2;

fplot({f p0 p1 p2},[0.2 5]);
f(2)
p0(2)
p1(2)
p2(2)

% E0,1,2 --> percent relative error at f(2)

E0 = abs(p0(2)-f(2))/f(2)*100
E1 = abs(p1(2)-f(2))/f(2)*100
E2 = abs(p2(2)-f(2))/f(2)*100

% e0,1,2 --> percent relative error at f(1.5)

e0 = abs(p0(1.5)-f(1.5))/f(1.5)*100
e1 = abs(p1(1.5)-f(1.5))/f(1.5)*100
e2 = abs(p2(1.5)-f(1.5))/f(1.5)*100
end
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance,
properly vectorize your function to return an output with the same size and
shape as the input arguments.
Warning: Function behaves unexpectedly on array inputs. To improve performance,
properly vectorize your function to return an output with the same size and
shape as the input arguments.

ans =

    0.2770


ans =

    0.5403


ans =

    0.6988


ans =

   -0.0713


E0 =

   95.0547
```

```
E1 =

  152.2854

E2 =

  125.7472

e0 =

   13.4607

e1 =

   30.1058

e2 =

   10.3261
```



Published with MATLAB® R2023a