

```

b = [0.537667139546100 - 1.30768829630527i, 1.83388501459509 -
0.433592022305684i, -2.25884686100365 + 0.342624466538650i, 0.862173320368121
+ 3.57839693972576i, 0.318765239858981 + 2.76943702988488i];
x = [0.0915735525189070 + 0.140903448980048i];

```

```

q = b(:);
m = length(b);
R = real(x);
I = imag(x);
v = 2*R;
u = -1*(R*R + I*I);

```

First of all, I made a copy of given data “b” which is called “q” for computing derivation of b(x). The Horner Rule is used for derivation of a complex number.

```

q(1) = b(1);
q(2) = b(2)+v*q(1);
for k = 3:m-1,
    q(k)= b(k)+ v*q(k-1) + u*q(k-2);
end

```

```

c = q(m-1);
e = b(m) + u*q(m-2);

```

Then I computed the q(x)(which called “h” in code) that was needed to evaluate  $b^{(1)}(x)$

```

p0 = (c*R)+e;
p1 = c*I;
p = p0 + p1*1i;
w = q(1:m-2);
h = w(:);
f = length(h);
h(1) = w(1);
h(2) = w(2)+v*w(1);
for k = 3:f-1,
    h(k)= w(k)+ v*h(k-1) + u*h(k-2);
end

```

```

c1 = h(f-1);
e1 = w(f) + u*h(f-2);
q0 = c1*R+e1;
q1 = c1*I;
res = c+2*I*(q0+ q1*1i)*1i;

```

“res” is the result of  $b^{(1)}(x)$  which is the input of  $a^{(2)}(x)$

```

a = [1.34988694015652, 3.03492346633185, 0.725404224946106,
-0.0630548731896562, 0.714742903826096, -0.204966058299775, -0.124144348216312];
x = res ;
n = length(a);

```

```

factj = 1;
for j = 0:2,
    for i= 2:n-j,
        a(i)=a(i)+ x*a(i-1);
    end
    a(n-j)=factj*a(n-j);
    factj = factj*(j+1);
end
o = a(n-2);
disp(round(o,5,'significant'));

```

The Horner Rule is used for computing derivation of real numbers