newton.m （2 点）

```
function [x,f] = newton( x0 )
x=x0;
delta=0.0001;
xb=x;
x = xb - func3_1p(xb)/func3_1p2(xb);
ite=0;
while abs(x-xb)>=delta
    ite=ite+1;
    xb=x;
    x = xb - func3_1p(xb)/func3_1p2(xb);
end
f=func3_1(x);
end
```

func3_1.m （1 点）

```
function f = func3_1( x )

f = x.^3 - 2*x.^2 + x + 3;
end
```

```
func3_1p.m
function fp = func3_1p( x )

fp = 3*x.^2 - 4*x + 1;
end
```

func3_1p2.m （1点）

```
function fp2 = func3_1p2( x )

fp2 = 6*x - 4;
end
```

プログラムの採点基準

計算結果が正しく出力されているのを見て、プログラムの内容をチェック。
間違っているところが明確なら、適宜減点。

＞いくつかの点からスタートして、収束するかどうか、試してみよ（この部分は、0 点～2 点）
2 つの点から試していて 1 点。
3 つ以上の点から試していて 2 点。

実行結果

```
>> [x,f]=newton(10)

x =

    1.0000


f =

     3

>> [x,f]=newton(-10)

x =

    0.3333


f =

    3.1481

>> [x,f]=newton(0)

x =

    0.3333
```

f =

3.1481

>>

考察（4 点満点）

この関数には、極値が 2 つある（x=1/3 で極大値 3.1481, 1 で極小値 3)。そのことに言及して 3 点（極大、極小の関数値に言及していないのは 2 点まで)。

おおよそ 1 より大きな値からスタートすると 1 に収束し、約 0.333…よりも小さい値からスタートすると 0.333…に収束する。その間に境界があるということに気づいて、1 点。

（参考）以下は、関数のグラフ。