

```
>> f = @(x) cos(x) - x;
>> p0 = 0.5;
>> p1 = pi/4;
>> tol = 10^-7;
```

Reproducing Example 3 from 2.3:

```
>> c = [0 1 2 3 4 5 6];
>> for x=1:7
newtonresult(x) = newton(f, p1, tol, c(x));
secantresult(x) = secant(f, p0, p1, tol, c(x));
fpresult(x) = falsePosition(f, p0, p1, tol, c(x));
end
>> N = c'; FalsePosition = fpresult'; Secant = secantresult'; Newton =
newtonresult';
>> T = table(N, FalsePosition, Secant, Newton)
```

T =

7×4 table

N	FalsePosition	Secant	Newton
—	—————	—————	—————
0	0.5	0.5	0.785398163397448
1	0.785398163397448	0.785398163397448	0.739536133515238
2	0.736384138836582	0.736384138836582	0.73908517810601
3	0.73905813921389	0.73905813921389	0.739085133215161
4	0.73908486381471	0.739085149337276	0.739085133215161
5	0.739085130526579	0.739085133215065	0.739085133215161
6	0.739085133188329	0.739085133215065	0.739085133215161

Estimating:

```
>> f = @(x) 230*x^4 + 18*x^3 + 9*x^2 - 221*x - 9;  
>> max_iter = 100;  
>> tol = 0.000001;  
>> p0a = -1;  
>> p1a = 0;  
>> p0b = 0.5;  
>> p1b = 1;
```

With Newton:

```
>> [p, iter] = newton(f, p0a, tol, max_iter)
```

p =

-0.040659288315759

iter =

6

```
>> [p, iter] = newton(f, p0b, tol, max_iter)
```

p =

0.962398418750541

```
iter =
```

```
4
```

With secant:

```
>> [p, iter] = secant(f, p0a, p1a, tol, max_iter)
```

```
p =
```

```
-0.040659288315725
```

```
iter =
```

```
5
```

```
>> [p, iter] = secant(f, p0b, p1b, tol, max_iter)
```

```
p =
```

```
0.962398418750561
```

```
iter =
```

```
9
```

With False Position:

```
>> [p, iter] = falsePosition(f, p0a, p1a, tol, max_iter)
```

p =

-0.040658499043342

iter =

17

>> [p, iter] = falsePosition(f, p0b, p1b, tol, max_iter)

p =

0.962398408376707

iter =

8