Thais Minet

2/15/18

Lab 5 Newton's again, but with PLOTS!

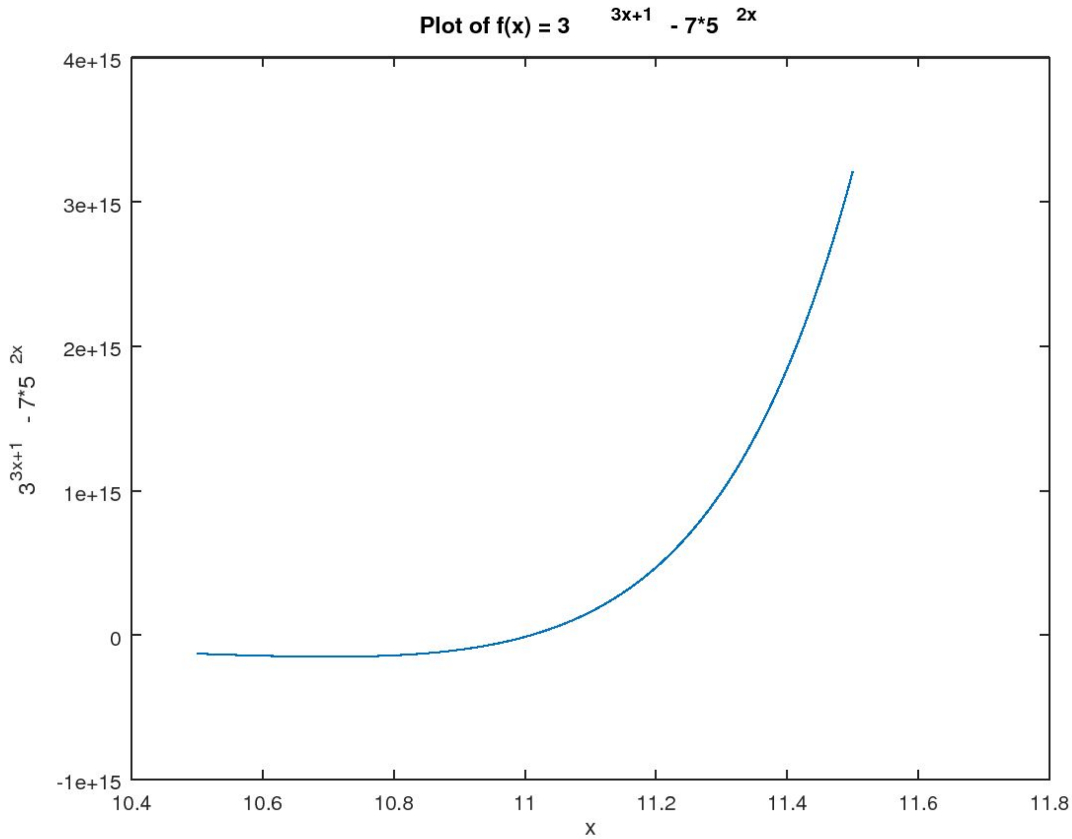
29. Let $f(x) = 3^{3x+1} - 7 * 5^{2x}$

a) Use the Maple commands solve and fsolve to try to find all roots of f.

```
>> f
f =

@(x) 3 .^ (3 * x + 1) - 7 * 5 .^ (2 * x)

>> fzero(f, 11)

ans =  11.00943864426814577939
```

In octave the function fzero(*fun*, *x0*) is used to find zeros of a function. Where *fun* is the function and *x0* is the initial guess.

b) Plot f(x) to find initial approximations to roots of f.



c) Use Newton's method to find roots of f to within 10^{-16}**.**

```
>> f
f =

@(x) 3 .^ (3 * x + 1) - 7 * 5 .^ (2 * x)

>> fprime
fprime =

@(x) log (3) * 3 .^ (3 * x + 2) - log (5) * 14 * 25 .^ x

>> newton(11, f, fprime, 10.^(-16), 100)

ITERATION 1
p0 = 11.00000000000000000000
p = 11.00973804015525203681
```

```
err = 0.00973804015525203681

ITERATION 2
p0 = 11.00973804015525203681
p = 11.00943893596633316179
err = 0.00029910418891887502

ITERATION 3
p0 = 11.00943893596633316179
p = 11.00943864426839802206
err = 0.00000029169793513972

ITERATION 4
p0 = 11.00943864426839802206
p = 11.00943864426819551738
err = 0.00000000000020250468

ITERATION 5
p0 = 11.00943864426819551738
p = 11.00943864426814577939
err = 0.00000000000004973799

ITERATION 6
p0 = 11.00943864426814577939
p = 11.00943864426814577939
err = 0.00000000000000000000

ans = 11.00943864426814577939
```

The answer is 11.00943864426814577939

Verification by direct calculation x = 11.00943864426814577939

So the result agrees with the calculation of the fzero function in octave to the specified accuracy.

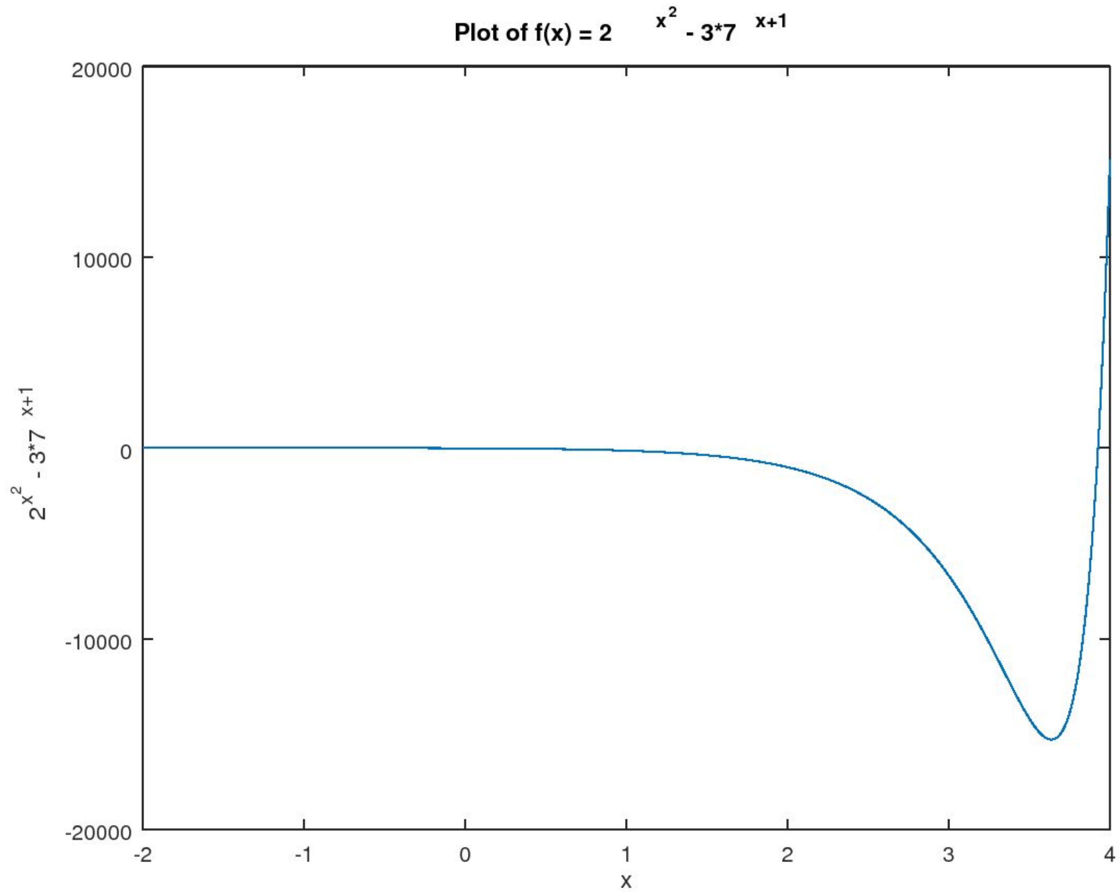
30. Repeat Exercise 29 using $f(x) = 2^{x^2} - 3 * 7^{x+1}$

a) Use the Maple commands solve and fsolve to try to find all roots of f.

```
>> f
f =

@(x) 2 .^ (x .^ 2) - 3 * 7 .^ (x + 1)

>> fzero(f, -1)

ans = -1.11874753039889740514

>> fzero(f, 4)

ans = 3.92610245245650313350
```

In octave the function fzero(*fun*, *x0*) is used to find zeros of a function. Where *fun* is the function and *x0* is the initial guess.

b) Plot f(x) to find initial approximations to roots of f.



c) Use Newton's method to find roots of f to within 10^{-16}**.**

Root 1

```
>> f
f =

@(x) 2 .^ (x .^ 2) - 3 * 7 .^ (x + 1)

>> fprime
fprime =

@(x) log (2) * x * 2 .^ (x .^ 2 + 1) - 3 * log (7) * 7 .^ (x + 1)

>> newton(-1, f, fprime, 10.^(-16), 100)

ITERATION 1
```

```
p0 = -1.00000000000000000000
p = -1.11613971332830619332
err = 0.11613971332830619332

ITERATION 2
p0 = -1.11613971332830619332
p = -1.11874750568056557931
err = 0.00260779235225938599

ITERATION 3
p0 = -1.11874750568056557931
p = -1.11874753039889629491
err = 0.00000002471833071560

ITERATION 4
p0 = -1.11874753039889629491
p = -1.11874753039889629491
err = 0.00000000000000000000

ans = -1.11874753039889629491
```

The answer is -1.11874753039889629491

Verification by direct calculation x = -1.11874753039889740514

So the result agrees with the calculation of the fzero function in octave to the specified accuracy.

Root 2

```
>> newton(4, f, fprime, 10.^(-16), 100)

ITERATION 1
p0 = 4.00000000000000000000
p = 3.94302547649809342900
err = 0.05697452350190657100

ITERATION 2
p0 = 3.94302547649809342900
p = 3.92715551425456066426
err = 0.01586996224353276475
```

```
ITERATION 3
p0 = 3.92715551425456066426
p = 3.92610675336725600815
err = 0.00104876088730465611

ITERATION 4
p0 = 3.92610675336725600815
p = 3.92610245252850287301
err = 0.00000430083875313514

ITERATION 5
p0 = 3.92610245252850287301
p = 3.92610245245650046897
err = 0.00000000007200240404

ITERATION 6
p0 = 3.92610245245650046897
p = 3.92610245245650046897
err = 0.00000000000000000000

ans = 3.92610245245650046897
```

The answer is 3.92610245245650046897

Verification by direct calculation x = 3.92610245245650313350

So the result agrees with the calculation of the fzero function in octave to the specified accuracy.