projectC_1004272

Alexandra Bramer

21/02/2016

Contents

1 Exercise 5.1

```
R.<x> = RR[]
f = x^2-2
f.roots()
[(-1.41421356237310, 1), (1.41421356237310, 1)]
```

This has given us the two real roots of the equation $x^2 - 2 = 0$, namely +/-root(2).

2 Exercise 5.2

```
R.<x> = ZZ[]
f = x^3-1
f.roots()
[(1, 1)]
```

This has given us the integer root of $x^3 - 1 = 0$

```
R.<x> = RR[]
f = x^3-1
f.roots()
[(1.00000000000000, 1)]
```

This has given the real root of the equation $x^3 - 1 = 0$

```
R.<x> = CC[]
f = x^3-1
f.roots()
[(1.00000000000000, 1), (-0.500000000000000 - 0.866025403784439*I, 1), (-0.500000000000000
+ 0.866025403784439*I, 1)]
```

This has given us the complex roots of the equation $x^3 - 1 = 0$

3 Exercise 5.3

```
x = var('x') #Here I am defining a variable
f(x)=x^3+x^2+1 #Inputting the equation
df=diff(f,x) #defining a function to differentiate f(x)
NewtonMethod(x)=x-(f/df)(x) #Code to apply Newton's method
xn=-1 #The first guess
for i in range(200):
    xn=N(NewtonMethod(xn),digits=10)
    if i > 198: #this if has been used to tell the programme to \
   print only the final iteration
        print xn
-1.465571232
```

I decided to create a loop to perform a large number of iterations; large enough that is is safe to assume that the result had converged, then set it to print only the final iteration. I used this to find one root of the equation $x^3 + x^2 + 1 = 0$

4 Exercise 5.4

```
R.<x> = RR[]
x = var('x')
f = R.random_element(5)
print f
f.roots()
```

```
0.386807958585245*x^5 + 0.603809618465823*x^4 + 0.833951256598776*x^3 -
0.426728117280960*x^2 + 0.739180815579568*x + 0.292508453622625
[(-0.311660230939565, 1)]
```

Here I used the command to generate a random real polynomial, and calculated the real root(s) using the root finder.

```
f(x) = f
df=diff(f,x)
NewtonMethod(x)=x-(f/df)(x)
xn=1
for i in range(200):
    xn=N(NewtonMethod(xn),digits=10)
    if i > 198:
        print xn
```

```
-0.3116602309
```

Here I used my code for Newton's method to find the real root to 10 digits.

5 Exercise 5.5

```
R.<x> = Zmod(5)[]
f = x^5-1
f.roots()
```

```
[(1, 5)]
```

These are the roots of the equation $x^5 - 1 = 0$ modulo 5; you return a value of 1 on 5 occasions.

```
factor(x^5-1)
```

```
(x + 4)^5
```

This is how the equation $x^5 - 1 = 0$ factors over (Z/5Z)[x].

6 Exercise 5.6

```
[(6, 1), (5, 1), (4, 1), (3, 1), (2, 1), (1, 1)]
```

These are the solutions of the equation $x^6 - 1 = 0$ over (Z/7Z)

7 Exercise 5.7

There will be 2016 roots by comparison with the previous answers; each number from 1 to 2016 will appear once. One could compute all of the roots in the same way as was done in 5.6 and enumerate them for verification.

8 Exercise 5.8

```
R.<x> = Zmod(49)[]
f = x^6-1
f.roots(multiplicities=False)
[1, 18, 19, 30, 31, 48]
```

These are the roots of $x^6 - 1 = 0$ over Z/49Z. Note that it is again 6, the same number as when we computed the roots over Z/7Z.

9 Exercise 5.9

There will again be 2016 solutions, the same number of solutions as there were for the same equation over Z/2017Z.

10 Exercise 5.10

One of the solutions will be x=1, and I will compute the other using a sum.

```
n=var('n')
x = sum(4*5^n,n,0,19)
print x
95367431640624
```

11 Exercise 5.11

First I will take roots for the equation mod 7.

```
R.<x> = Zmod(7)[]
f = 3*x^5-2*x^4-x^3-4*x^2-2*x-1
f.roots()
```

```
[(6, 1), (5, 1), (3, 1), (2, 1), (1, 1)]
```

Note that all of these roots are suitable a0 values

```
R.<x> = Zp(7,5)[]
x = var('x')
f(x)=3*x^5-2*x^4-x^3-4*x^2-2*x-1
df=diff(f,x)
NewtonMethod(x)=x-(f/df)(x)
print('First use a0=1')
xn=1
for i in range(200):
    xn=N(NewtonMethod(xn),digits=10)
    if i > 198:
        print xn
R.<x> = Zp(7,5)[]
```