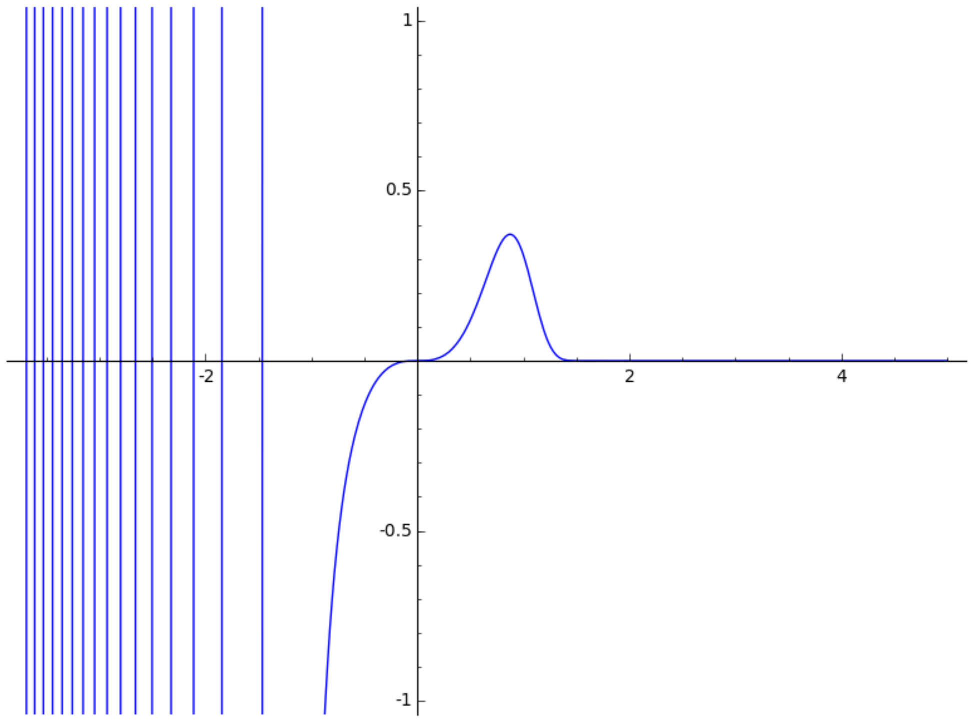
ID : 102062111, Name : Chih-Min Lin

1. (a)



(b)

Because there are two functions, sin() and exp(), are too difficult to get the numerical result. If this equation has only 'exp()' or 'sin()', we could get the numerical result more easily.

(c)

We can use 'numerical_integral()' function to get approximated value of this integral.
The result will return two values, left one is approximated value, and the other is an error estimate.

2. (a) $\because$ p is prime, and p is coprime between all integers from 1 to p except number p
$\therefore\ phi(p)\ is\ p-1$

(b) Obervation:
$\phi(11) \times \phi(13) = \phi(11 \times 13)$
In my opinion, all distinct prime numbers p and q can be done by this observation, it looks really make sence. We should prove it in the next question.

(c)

(i) Suppose we have distinct prime number p and q.

(ii) $\because$ p is prime number, and q is also prime number
$\therefore\ \phi(p) = p-1,\ \phi(q) = q-1,\ \phi(p)\phi(q) = (p-1)(q-1) = pq - p - q - 1$

(iii) $\because$ We observe "number set" coprime between $p \times q$ in range $[1,\ p \times q]$:
$\{1,\ 2,\ 3,\ ,\cdots\cdots,\ p \times q\} - \{p \times 1,\ p \times 2,\ \cdots\cdots p \times q\} - \{q \times 1,\ q \times 2,\ \cdots\cdots q \times p\} + \{p \times q\}$

$\therefore$ The number of $PRIME\ SET\ = \phi(p \times q) = p \times q - p - q + 1$

(iv) $\because (ii)\ is\ equal\ to\ (iii)$
$\therefore\ the\ equation\ holds.$

3. (a) Use is_prime() function to check whether a number is prime or not.
Sample code : is_prime(2**1279-1)
Result : 2^{1279-1} is prime number

(b) If values of a and p are really large, operations of calculating the result of $a^{p-1}\ \%\ p$ will cost a lot.

(c)

Suppose we can find that $a^n\ mod\ p = 1,\ nispositiveinteger$
If $a^{p-1}\ mod\ p = 1$, then
$a^{p-1} = a \times a \times a \times .... = a^n \times a^n \times .....,\ \frac{p-1}{n}$ will be positive integer
Otherwise, the theory will not hold.

Below is code implemented in SageMath environment :

```

def fima(a ,p):
    count = 0
    cur = 1
    n = -1
    for i in range(1, p):
        count = count + 1
        cur = (cur * a) % p
        if (cur == 1):
            n = count
            print n
            break
    return n

def check_correct(p, n):
    if (p - 1) % n != 0:
        print "Theory is not correct"
    else:
        print "Theroy is correct"

p = 1279

a = 123 # First a
b = 456 # Second a
c = 789 # Third a

check_correct(p, fima(a,p))
check_correct(p, fima(b,p))
check_correct(p, fima(c,p))
```

Below is the result of program running:

```
➜  midterm2 git:(master) ✗ sage fima.py
n = 142, p-1 = 1278
Theroy is correct
n = 639, p-1 = 1278
Theroy is correct
n = 1278, p-1 = 1278
Theroy is correct
```