Simpson Double (Polar Coordinates)

May 16, 2018

In [1]:
```
from sympy import*
%matplotlib inline
import matplotlib.pyplot as plt
from __future__ import division
x, y, z, t = symbols('x y z t')
k, m, n = symbols('k m n', integer = True)
f, g, h = symbols('f g h', cls = Function)
import math
```

In [2]:
```
def compositeSimpson(startPoint, endPoint, numIntervals):
    a = startPoint
    b = endPoint
    n = numIntervals
    h = (b - a) / n
    leftRight = f(a) + f(b)
    oddSum = 0
    evenSum = 0
    for i in range (1, n):
        x = a + i*h
        if i % 2 == 1:
            oddSum = oddSum + f(x)
        else:
            evenSum = evenSum + f(x)
    areaEstimate = h * (leftRight + 2*evenSum + 4*oddSum) / 3
    return areaEstimate
```

In [3]:
```
def SimpsonDouble(a, b, m, n, c, d, f):
    h = (b - a)/n
    J1 = 0
    J2 = 0
    J3 = 0
    for i in range (0, n+1):
        x = a + i*h
        HX = (d(x) - c(x))/m
        K1 = f(x, c(x)) + f(x, d(x))
        K2 = 0
        K3 = 0
```

```python
            for j in range (1, m):
                y = c(x) + j*HX
                Q = f(x, y)
                if j%2 == 0:
                    K2 = K2 + Q
                else:
                    K3 = K3 + Q
            L = ((K1 + 2*K2 + 4*K3)*HX)/3
            if i ==0 or i == n:
                J1 = J1 + L
            elif i % 2 == 0:
                J2 = J2 + L
            else:
                J3 = J3 + L
        J = h*(J1 + 2*J2 + 4*J3)/3
        return J
```

In [11]:
```python
n = 12000
def f(x, y):
    #return 2
    return y**(-3)
def c(x):
    return 1 / math.cos(x)
def d(x):
    return 2*math.cos(x)
result1 = SimpsonDouble(0, math.pi/4, n, n, c, d, f)
def f1(x, y):
    return (x**2 + y**2)**(-2)
def c1(x):
    return -(1 - (x - 1)**2)**(1/2)
def d1(x):
    return 0
result2 = SimpsonDouble(1, 2, n, n, c1, d1, f1)
true = math.pi/16
print ("error using polar coordinates is " + str(abs(result1 - true)))
print ("error using Cartesian coordinates is " + str(abs(result2 - true)))
```

```
error using polar coordinates is 1.3877787807814457e-16
error using Cartesian coordinates is 5.45868461543364e-09
```