MATH 151 Lab 9 Solutions

TA: Nida Obatake

In [1]:
```
from sympy import *
from sympy.plotting import *
```

#1 Given $y = (4x + 1)^{\cot x}$:

(a) By hand, rewrite $\ln \ y$ as a fraction $\frac{f(x)}{g(x)}$. Define f and g in Python.

(b) Show $f(0)$ and $g(0)$ are either both zero or both infinite.

(c) Evaluate $f'(0)$ and $g'(0)$. Use these values to compute $\lim_{x \to 0^+} y$.

(d) Evaluate the limit directly to verify your answer to part (c).

In [2]:
```
#1
x = symbols('x',real=True)
y = (4*x+1)**(cot(x))
#1a
g = 1/cot(x)
f = ln(4*x+1)
#lny = f/g
print("1a) As a fraction, ln y = {} / {}.".format(f,g))
#1b
print("1b) f(0) = {} and g(0) = {}".format(f.subs(x,0),g.subs(x,0)))
print("Both of f(0) and g(0) are zero, hence we have an indeterminate fo
rm 0/0.")
#1c
df = simplify(diff(f,x)).subs(x,0)
dg = simplify(diff(g,x)).subs(x,0)
print("1c) f'(0) = {} and g'(0) = {}".format(df,dg))
print("The limit of y as x approaches 0 from the right-hand side is {}."
.format(exp(df/dg)))
#1d
L = limit(y,x,0,'+')
print("1d) Computed directly, the limit of y = (4x+1)^(cot(x)) as x appr
oaches 0 from the right-hand side is {}.".format(L))
```

```
1a) As a fraction, ln y = log(4*x + 1) / 1/cot(x).
1b) f(0) = 0 and g(0) = 0
Both of f(0) and g(0) are zero, hence we have an indeterminate form 0/
0.
1c) f'(0) = 4 and g'(0) = 1
The limit of y as x approaches 0 from the right-hand side is exp(4).
1d) Computed directly, the limit of y = (4x+1)^(cot(x)) as x approaches
0 from the right-hand side is exp(4).
```

#2 A piece of wire 40 cm long is divided into at most two pieces. One piece is bent into a square and the other is bent into a circle.

(a) How should the wire be divided so that the total area enclosed by the two shapes is a maximum?

(b) How should the wire be divided so that the total area enclosed by the two shapes is a minimum?

Define variables, and domain

When a wire 40 cm long is divided into at most two pieces, if w is equal to the length of one of the pieces, then the other piece has length 40-w, where 0 <= w <= 40.

The square Let's bend the piece of length w into a square. The square will have side length w/4. The area enclosed by the square is $A = (w/4)^2$.

The circle Let's bend the piece of length 40-w into a circle. The circumference of the circle (which has radius r) is $40 - w = 2\pi r$. This means that the radius of this circle is $r = (40 - w)/(2\pi)$, and the area enclosed by the circle is $A = \pi r^2$.

In [3]:
```
#2
w,r = symbols('w r',real=True)

Asq = (w/4)**2
rad = solve(w+2*pi*r-40,r)[0]
Aci = pi*rad**2

A = Asq + Aci

#2a
dA = diff(A,w)
crit = [0,40] + solve(dA,w,positive=True)
areas = [A.subs(w,c) for c in crit]
maxA = max(areas)
wmaxA = crit[areas.index(max(areas))]
print("To yield a maximum total enclosed area of approximately {}, the wire should be divided into a piece of length {} which gets bent into the square and a piece of length {} which gets bent into the circle.".format(maxA.evalf(4),wmaxA,40-wmaxA))
#2b
minA = min(areas)
wminA = crit[areas.index(min(areas))]
print("To yield a minimum total enclosed area of approximately {}, the wire should be divided into a piece of length (approximately) {} which gets bent into the square and a piece of length (approximately) {} which gets bent into the circle.".format(minA.evalf(4),(wminA).evalf(4),(40-wminA).evalf(4)))
```

```
To yield a maximum total enclosed area of approximately 127.3, the wire should be divided into a piece of length 0 which gets bent into the square and a piece of length 40 which gets bent into the circle.
To yield a minimum total enclosed area of approximately 56.01, the wire should be divided into a piece of length (approximately) 22.40 which gets bent into the square and a piece of length (approximately) 17.60 which gets bent into the circle.
```

#3 . Given $f(x) = e^{-x}$:

(a) Estimate the area under the graph between $x = 0$ and $x = 1$ with $n = 10, 100, 1000$ left endpoint rectangles.

(b) Compare your answers in part (a) to the exact area, $1 - \frac{1}{e^2}$. What do you notice?

It will be helpful for us to use the following function (which can be found in the numpy package): **arange(...)**, which returns evenly spaced values within a given interval and is used as follows:

arange([start,] stop[, step,], dtype=None)

where:

start : number, optional

Start of interval. The interval includes this value. The default start value is 0.

stop : number

End of interval. The interval does not include this value, except in some cases where step is not an integer and floating point round-off affects the length of out.

step : number, optional

Spacing between values. For any output out, this is the distance between two adjacent values, out[i+1] - out[i]. The default step size is 1. If step is specified as a position argument, start must also be given.

dtype : dtype

The type of the output array. If dtype is not given, infer the data type from the other input arguments.

Values are generated within the half-open interval `[start, stop)` (in other words, the interval including `start` but excluding `stop`).

In [4]:
```
from numpy import arange
```

In [5]:
```
#3
x, n = symbols('x n')
f = E**(-x)
#x=0, x=1
a=0
b=2
#3a
dx = (b-a)/n
print("3a)")
for nrect in [10,100,1000]:
    width = dx.subs(n,nrect)
    heights = [f.subs(x,xi) for xi in arange(a,b,width)]
    area = (sum(heights)*width).evalf(5)
    print("Using {} left endpoint rectangles, the area is approximately
{}.".format(nrect,area))
#3b
print("The exact area is 1-1/e^2, which is approximately {}.".format((1-
1/E**2).evalf(5)))
print("3b) We notice that the approximations all yielded a larger area t
han the actual area; this means that using left-endpoint rectangles, are
a is overestimated, but as the number of rectangles is increased, the es
timate gets closer to the actual area.")
```

```
3a)
Using 10 left endpoint rectangles, the area is approximately 0.95401.
Using 100 left endpoint rectangles, the area is approximately 0.87334.
Using 1000 left endpoint rectangles, the area is approximately 0.86553.
The exact area is 1-1/e^2, which is approximately 0.86467.
3b) We notice that the approximations all yielded a larger area than th
e actual area; this means that using left-endpoint rectangles, area is
overestimated, but as the number of rectangles is increased, the estima
te gets closer to the actual area.
```

All done :)