Work sheet 2

Sajeev G P

1/14/2019

```
# Symbolic Expressions and Simplification
x, y, z = var('x, y, z') ; q = x*y + y*z + z*x
bool(q(x=y, y=z, z=x) == q), bool(q(z=y)(y=x) == 3*x^2)
```
(True, True)

```
#To replace an expression more complex than a single variable, the \
    substitute
#method is available:
y, z = var('y, z'); f = x^3 + y^2 + z
f.substitute(x^3 == y^2, z==1)
```
2*y^2 + 1

```
#Transforming Expressions
x, y = SR.var('x,y')
f(x)=(2*x+1)^3
```

```
f.expand()
```
x |--> 8*x^3 + 12*x^2 + 6*x + 1

```
y = var('y'); u = sin(x) + x*cos(y)
v = u.function(x, y); v
```
(x, y) |--> x*cos(y) + sin(x)

```
w(x, y) = u; w
```
(x, y) |--> x*cos(y) + sin(x)

```
p = (x+y)*(x+1)^2
p2 = p.expand(); p2
```
x^3 + x^2*y + 2*x^2 + 2*x*y + x + y

```
#collect method groups terms together according to the powers of a \
    given variable:
p2.collect(x)
```
x^3 + x^2*(y + 2) + x*(2*y + 1) + y

```
((x+y+sin(x))^2).expand().collect(sin(x))
```

x^2 + 2*x*y + y^2 + 2*(x + y)*sin(x) + sin(x)^2

```
#Usual Mathematical Functions
```

```
(x^x/x).simplify()
```

x^(x - 1)

```
f = cos(x)^6 + sin(x)^6 + 3 * sin(x)^2 * cos(x)^2
f.simplify_trig()
```

1

```
#Expressions containing factorials can also be simplified:
n = var('n'); f = factorial(n+1)/factorial(n)
f.simplify_factorial()
```

n + 1

```
# Solving Equations
z, phi = var('z, phi')
eq = z**2 - 2/cos(phi)*z + 5/cos(phi)**2 - 4 == 0; eq
```

z^2 - 2*z/cos(phi) + 5/cos(phi)^2 - 4 == 0

```
#We can extract the left-hand (resp. right-hand) side with the lhs (\
    resp. rhs)
eq.lhs()
```

z^2 - 2*z/cos(phi) + 5/cos(phi)^2 - 4

```
eq.rhs()
```

0

```
#then solve it for z with solve:
solve(eq, z)
```

[z == -(2*sqrt(cos(phi)^2 - 1) - 1)/cos(phi), z == (2*sqrt(cos(phi)^2 - 1) + 1)/cos(phi)]

```
#solve the equation x^2+5x+6=0
x=var('x')
eq=x^2+5*x+6==0
solve(eq,x)
```

[x == -3, x == -2]

```
#The roots of the equation can be returned as an object of type \
    dictionary

solve(eq,x, solution_dict=True)
```

[{x: -3}, {x: -2}]

```
# Question
```

```
# Solve the generic quadratic equation ax^2 + bx + c=0 for follwoing\
    a, b and c
# i) 1, 25, 150  ii) 3, -10, -35
```

```
# Solving system of equations
x,y=var('x', 'y')
eq1= x+y==6
eq2= 2*x-4*y==0
solve([eq1,eq2],x,y)
```

[[x == 4, y == 2]]

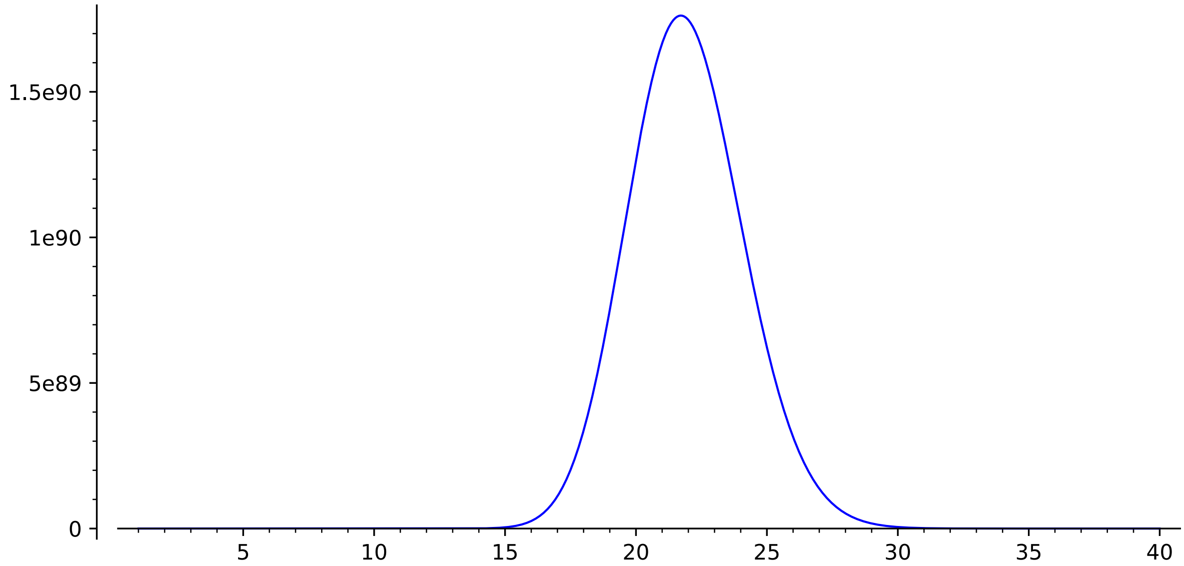
```
#Question
# Solve cos(x)sin(x)=1/2, x+y == 0
```

```

#Sequences
u(n) = n^100 / 100^n
u(1.);u(2.);u(3.);u(4.);u(5.);u(6.);u(7.);u(8.);u(9.);u(10.)
```

0.0100000000000000
1.26765060022823e26
5.15377520732011e41
1.60693804425899e52
7.88860905221012e59
6.53318623500071e65
3.23447650962476e70
2.03703597633449e74
2.65613988875875e77
1.00000000000000e80

```
#To get an idea of the variation of the sequence, we can draw the \
   graph of
#the function
plot(u(x), x, 1, 40)
```




```
#Derivative diff(f(x), x)
#n-th derivative diff(f(x), x, n)
#Antiderivative integrate(f(x), x)
#Numerical integration integral_numerical(f(x), a, b)
#Symbolic summation  sum(f(i), i, imin, imax)
#Limit limit(f(x), x=a)
#Taylor expansion taylor(f(x), x, a, n)
#Power series expansion f.series(x==a, n)
#Graph of a function plot(f(x), x, a, b)
```

```
diff(1 + x + x^2, x)
```

2*x + 1

```
numerical_integral(1 + x + x^2, 0, 3)[0]  # [1] gives error bound
```

16.500000000000004

```
diff(sin(x^2), x)
```

2*x*cos(x^2)

```
sin(x).integral(x, 0, pi/2)
```

1

```
integrate(1/(1+x^2), x, -infinity, infinity)
```

pi