MATH1110H-B-lab-F01-2023-10-10

October 11, 2023

[1]:
```
# MATH 1110H-B Lab 2023-10-10
#
# Our objective is to learn to solve (simple!) differential equations
# using SageMath.  To do this,  we need to be able to declare a generic
# function,  be able to take derivatives of functions,  generic or
# otherwise,  and solve equations including such derivatives.
#
# We first declare y to be a generic function of x:
#
y = function('y')(x)  # y = function('y',x) should work,  too.
#
# diff(y,x) represents the derivative of y with respect to x,  and the
# desolve command is optimized to solve equations involving derivatives.
# Note that as with the basic solve command,  desolve needs to be told
# explicitly what to solve for.
#
desolve(diff(y,x) == x^2,y)
#
# Note the generic constant _C in the answer provided by desolve.
```

[1]:
```
1/3*x^3 + _C
```

[2]:
```
# In most applications of differential equations we are also given
# requirements along the lines of "when x= we should have y= ".  These
# are called "initial conditions" and can be specified in desolve by
# adding ics[<x-value>,<y-value>].  For example,  if we were to specify
# that when x=4 we should have y=4 in the example above,  we would type
# in:
#
desolve(diff(y,x) == x^2,y,ics=[4,4])
#
# Note that desolve then gives what was formerly a generic constant an
# explicit value.
```

[2]:
```
1/3*x^3 - 52/3
```

[3]:
```
diff(sin(x^2),x)  # One can also use the diff operator to just take
                  # the derivative of a function...
```

[3]:
```
2*x*cos(x^2)
```

[4]:
```
diff(sin(x^2),x,2)  # ... or its second derivatives,  i.e. the
                    # derivative of the derivative.
```

[4]:
```
-4*x^2*sin(x^2) + 2*cos(x^2)
```

[5]:
```
desolve(diff(y,x) == y^2, y)  # desolve can also cope with equations
                              # where y also appears outside the
                              # derivative,  though you may need to do
                              # a bit more work to actually finish
                              # solving for y.
```

[5]:
```
-1/y(x) == _C + x
```

[6]:
```
desolve(diff(y,x) == y^2, y, ics=[500,2])  # Again,  we can pin down the
                                           # generic constant by giving
                                           # initial conditions,  in this
                                           # case y=2 when x=500.
```

[6]:
```
-1/y(x) == x - 1001/2
```

[7]:
```
N(sin(169))  # One little bonus trick:  if you need or prefer a decimal
             # a decimal approximation to some expression that gives a
             # number,  use the N command to get 15 digits of accuracy.
```

[7]:
```
-0.601999867677605
```

[ ]:
```

```