```
#
#
#
#
#
#
#
from casadi import *
```

A simple case of Callback

Callback allows the user to create functions that can be embedded into CasADi expressions. The user creates a class that inherits from this class and implements a subset of the virtual methods. Although Callback itself is implemented is C++, the virtual methods can be implemented in Python or MATLAB thanks to cross-language polymorphism as supported by the SWIG framework.

```

class Fac(Callback):
  def __init__(self, name, opts={}):
    Callback.__init__(self)
    self.construct(name, opts)

  def get_n_in(self): return 1
  def get_n_out(self): return 1

  def eval(self, arg):
    x = arg[0]
    y = 1
    for i in range(x):
      y*=(i+1)
    return [y]

fac = Fac('fac')
```

Evaluate numerically

```
y = fac(4)

print "4! = ", y
```

```
4! =  24
```

Using the function in a graph

```

x = MX.sym("x")
y = fac(x)

f = Function('f', [x],[y])

y = f(5)

print "5! = ", y
```

```
5! =  120
```