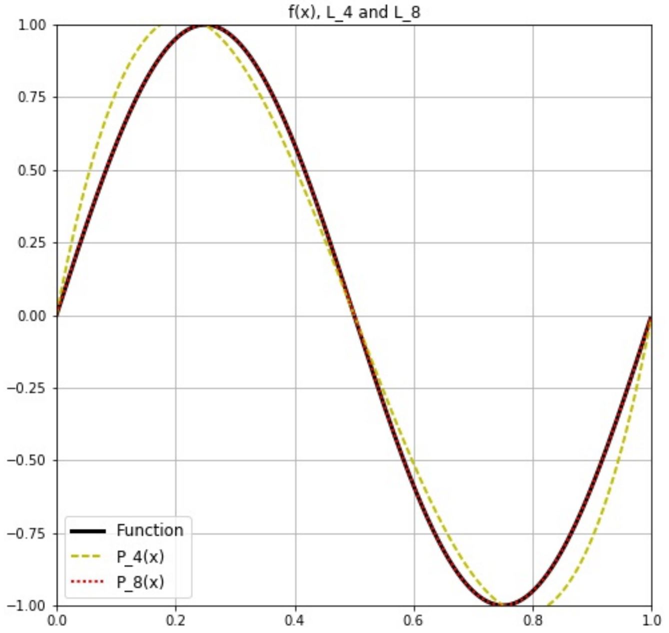
Lagrange Interpolations

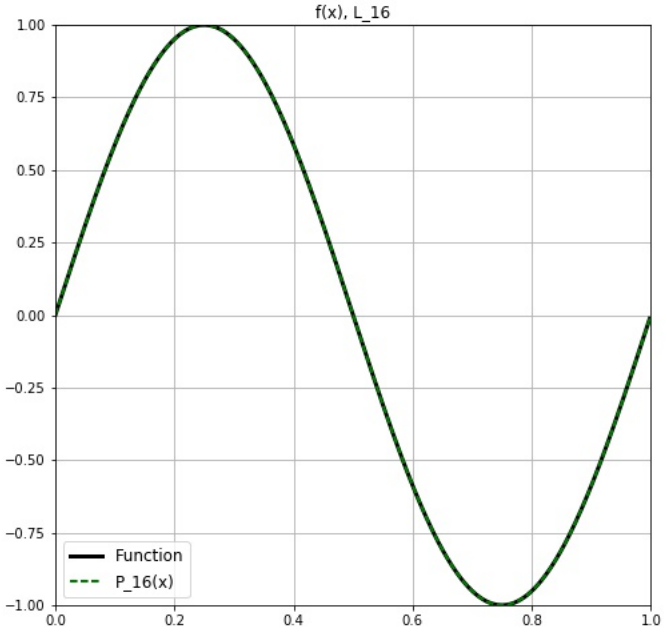
Colton Williams

September 20, 2017

Part I

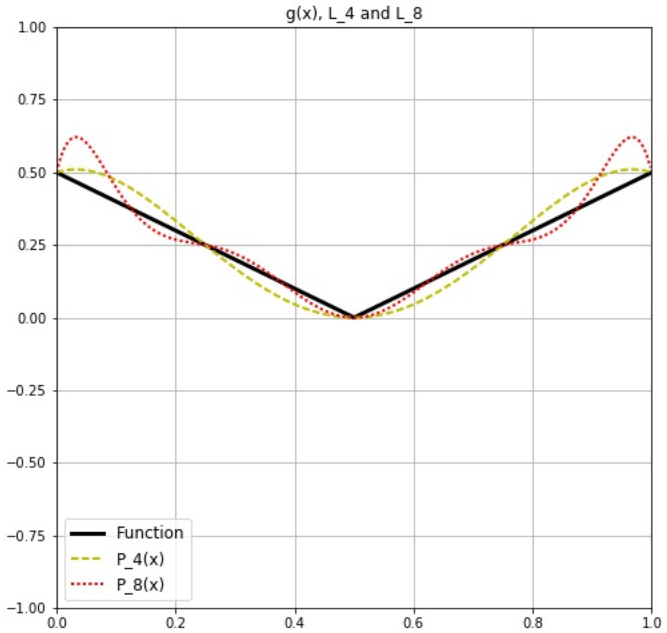
Interpolations for $f(x) = sin(2\pi x)$

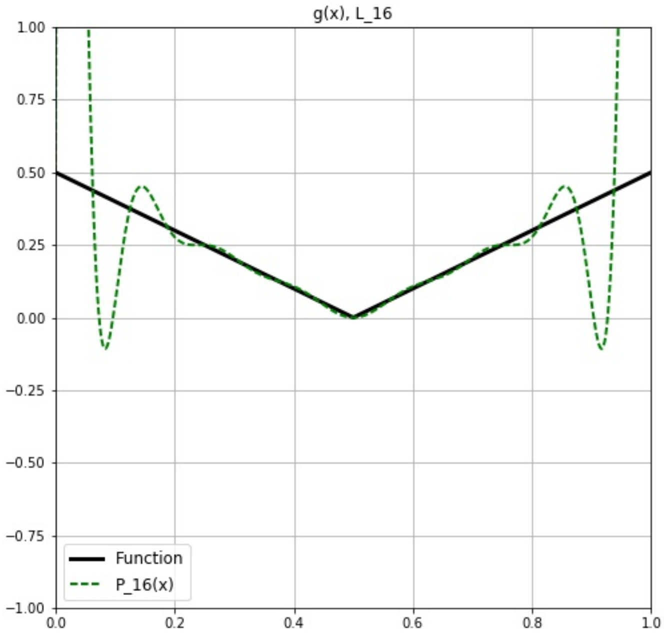




Part II

Interpolations for $g(x) = |x - 0.5|$





Part III

Code

```
# COLTON WILLIAMS
# NUMERICAL ANALYSIS
# LAGRANGE INTERPOLATION


import math from matplotlib
import pyplot
import numpy

def f(x): return math.sin(2.0*math.pi*x)

```

```python
def g(x): return abs(x - 0.5)

def lagrange(xs, n):

    xlist = numpy.append(xs[0:n], xs[n+1:])
    numerator = numpy.prod( numpy.asarray([ float(xs[n]-non) for non in xlist ]) )

    def L_n(x):
        return numpy.prod( numpy.asarray([ float(x - non) for non in xlist ]) ) / numerator

    return L_n

def construct(funct, n):

    xlist = numpy.asarray([float(i)/n for i in range(n+1)])
    l = [lagrange(xlist, i) for i in range(n+1)]
    v = numpy.vectorize(funct)
    ylist = v(xlist)

    def polynomial(x):
        return numpy.sum( numpy.asarray([ ylist[i]*l[i](x) for i in range(n+1)]) )

    return polynomial

def plot48(eq, title):

    lagrange4  = construct(eq, 4)
    lagrange8  = construct(eq, 8)

    origv = numpy.vectorize(eq)
    v4 = numpy.vectorize(lagrange4)
    v8 = numpy.vectorize(lagrange8)

    origx, origy = numpy.asarray([float((1.0/512)*i) for i in range(512)]), origv(numpy.asarray([float((1.0/512)*i) for i in range(512)]))
    x4, y4 = numpy.asarray([float((1.0/512)*i) for i in range(512)]), v4(numpy.asarray([float((1.0/512)*i) for i in range(512)]))    x8, y8 = numpy.asarray([float((1.0/512)*i) for i in range(512)]), v8(numpy.asarray([float((1.0/512)*i) for i in range(512)]))

    fig    = pyplot.figure(figsize=(8,8))

    pyplot.title(title)

    pyplot.plot(origx, origy, '-', color = 'k', linewidth=3., label='Function')
    pyplot.plot(x4, y4, '--', color = 'y', linewidth=2., label='P_4(x)')
    pyplot.plot(x8, y8, ':', color = 'r', linewidth=2., label='P_8(x)')
    pyplot.axis([0.0, 1.0, -1.0, 1.0])

    pyplot.grid()
    pyplot.legend(loc=3, fontsize='large')
    pyplot.show()

    return fig

def plot16(eq, title):

    lagrange16 = construct(eq, 16)
    origv = numpy.vectorize(eq)
    v16 = numpy.vectorize(lagrange16)
    origx, origy = numpy.asarray([float((1.0/512)*i) for i in range(512)]), origv(numpy.asarray([float((1.0/512)*i) for i in range(512)]))
    x16, y16 = numpy.asarray([float((1.0/512)*i) for i in range(512)]), v16(numpy.asarray([float((1.0/512)*i) for i in range(512)]))

    fig = pyplot.figure(figsize=(8,8))

    pyplot.title(title)

    pyplot.plot(origx, origy, '-', color = 'k', linewidth=3., label='Function')
    pyplot.plot(x16,y16, '--', color='g', linewidth=2., label='P_16(x)')
    pyplot.axis([0.0, 1.0, -1.0, 1.0])

    pyplot.grid()
    pyplot.legend(loc=3, fontsize='large')
    pyplot.show()

    return fig

def main():

    plot48(f, "f(x),␣L_4␣and␣L_8").savefig('sin48.jpg')
    plot16(f, "f(x),␣L_16").savefig('sin16.jpg')
    plot48(g, "g(x),␣L_4␣and␣L_8").savefig('abs48.jpg')
    plot16(g, "g(x),␣L_16").savefig('abs16.jpg')

if __name__ == "__main__":
    main()
```