

Math 452, Numerical Methods:

Polynomial Interpolation Python Programs

```
from matplotlib.pyplot import *
from numpy import *

xpt = array([0,2,3,5])
ypt = array([1,5,0,8])

#Newton's Divided difference, produces coefficients of
#interpolating polynomial
def NDD(x,y):
    n = len(x)
    #Construct table and load xy pairs in first columns
    A = zeros((n,n+1))
    A[:,0]= x[:]
    A[:,1]= y[:]
    #Fill in Divided differences
    for j in range(2,n+1):
        for i in range(j-1,n):
            A[i,j] = (A[i,j-1]-A[i-1,j-1]) / (A[i,0]-A[i-j+1,0])
    #Copy diagonal elements into array for returning
    p = zeros(n)
    for k in range(0,n):
        p[k] = A[k,k+1]
    return p

#Evaluates polynomial at 't' given x-values and coefficients
def poly(t,x,p):
    n = len(x)
    out = p[n-1]
    for i in range(n-2,-1,-1):
        out = out*(t-x[i]) + p[i]
    return out

#Calculates correct coefficients and graphs interpolating polynomial
a = NDD(xpt,ypt)
tval = linspace(min(xpt)-1,max(xpt)+1,100)
yval = poly(tval,xpt,a)
plot(tval,yval,color='green',linestyle='--',label='poly')
```

```
#Annotates graph and plots points used for interpolation
title('Interpolation')
xlabel('x values')
ylabel('y values')
legend(loc='best')
plot(xpt,ypt,color='blue',marker='o',linestyle='')
```