~\Documents\documents_general\structured_courses\math564\evaluations\projects \p05\objective6.py

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import numpy as np

def pathtime(x,p,nargout):

    import numpy as np
    from scipy.interpolate import interpn

    '''
    parse the input data
    n is the number of sinusoid coefficients for x and for y
    w,z are the decision variable weights
    v is the velocity map array (rows are y, cols are x)
    A,B are the (x,y) coordinates of the beginning and ending points
    '''
    n=int(len(x)/2)
    w=x[0:n]
    z=x[n:2*n]
    v,A,B=p
    my,mx=v.shape

    '''
    construct a piecewise linear path approximation for computation
    xx,yy are the x and y coordinates along the path on [0,1]x[0,1]
    s is the variable that parametrically defines the path
    '''
    smp=1000
    s=np.linspace(0,1,smp)
    xx=(1-s)*A[0]+s*B[0]
    yy=(1-s)*A[1]+s*B[1]
    for k in range(n):
        S=np.sin((k+1)*np.pi*s)
        xx+=w[k]*S
        yy+=z[k]*S

    '''
    xxr,yyr are the coordinates of the midpoint of each line segement
    in the velocity array size units.  Any points outside of the array
    are set at the boundary using max/min functions
    '''
    xxr=xx*(mx-1)
    yyr=yy*(my-1)
    xxr=(xxr[1:smp+1]+xxr[0:-1])/2
    yyr=(yyr[1:smp+1]+yyr[0:-1])/2
    xxr=np.maximum(np.minimum(xxr,mx-1),0)
    yyr=np.maximum(np.minimum(yyr,my-1),0)

    '''
    compute the travel time.  dist is the distance on [0,1]x[0,1]
```

```
    between line segment end points -- summed.  vel is the velocity
    at the midpoint interpolated from array data.  f is travel time.
    '''
    dist=np.sqrt(np.diff(xx)**2+np.diff(yy)**2)
    vel=interpn((range(mx),range(my)),v,(xxr,yyr),method='linear')
    f=sum(dist/vel)

    if nargout==1:
        return f
    else:
        # compute the gradient by approximation
        sm=1E-8
        df=np.zeros((2*n,1))
        for j in range(2*n):
            y=x.copy()
            y[j]+=sm
            df[j]=pathtime(y,p,1)
        g=(df-f)/sm
        return f,g


```