IntroPython

November 6, 2023

[1]:
```
#Pierce Zhang, CMOR220, Fall 2023, Project 9: Intro to Python. Least Squares␣
↪Polynomial Fitting
#IntroPython.ipynb
#Uses the Vandermonde matrix to produce approximations for a curve
#Last modified: November 6, 2023
```

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
```

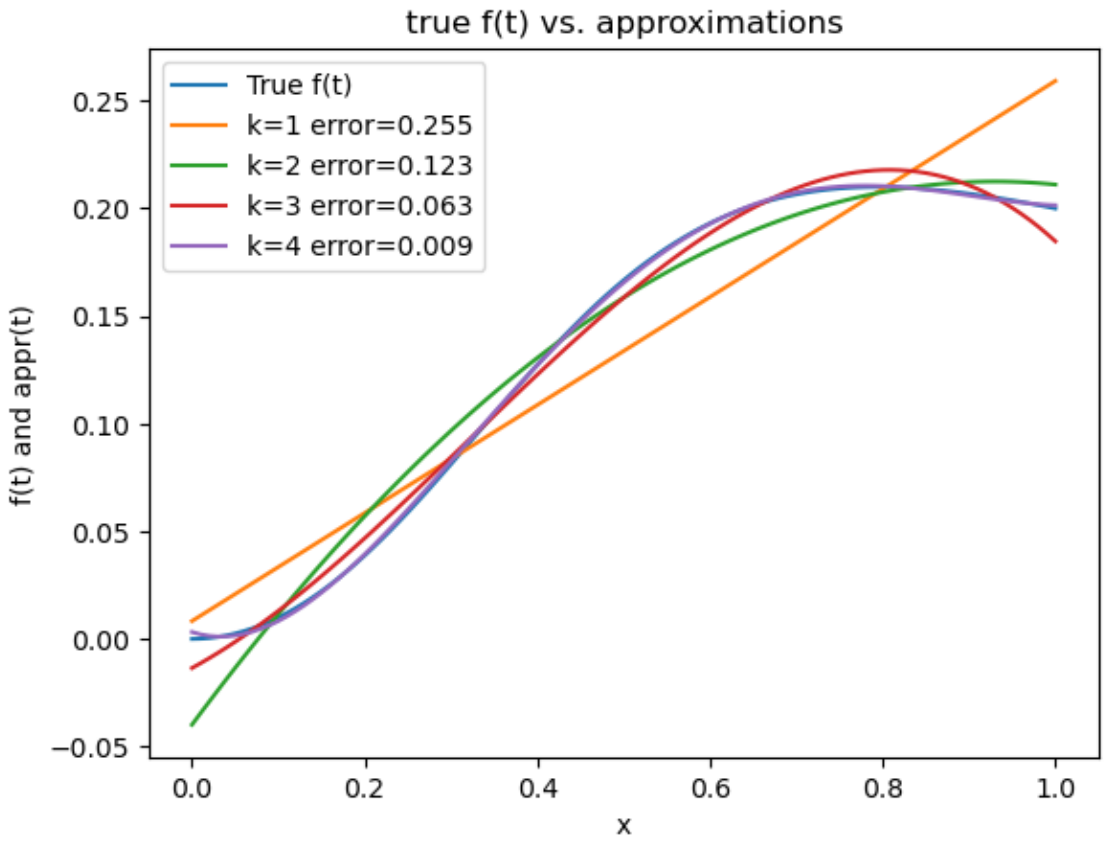
[3]:
```
# Inputs: t, the input to the specified function
# Outputs: value of f(t) as specified
def f(t):
    return t**2/(1 + 4*t**3)
```

[4]:
```
# Inputs: t, an array of the coefficients of t_i, which are used to define
# the Vandermonde matrix; and k, the highest power
# Outputs: A, the Vandermonde matrix itself
def vandermonde(t,k):
    A = np.zeros([len(t),k+1])
    for n1 in range(0,len(t)):
        for n2 in range(0,k+1):
            A[n1,n2]=t[n1]**n2
    return A
```

[5]:
```
# Inputs: t, an array of the coefficients of t_i, and f, the function
# to be used to load
# Output: b, the matrix f(t_0) ... f(t_last)
def load(t,f):
    b = np.zeros([len(t),1])
    for n in range(0,len(t)):
        b[n,0]=f(t[n])
    return b
```

[6]:
```
# Driver function
# Plot true function
plt.figure()
t = np.arange(0,1.01,0.01)
plt.plot(t,f(t),label="True f(t)")
```

```
# For each order to be used in the Vandermonde matrix
for k in [1,2,3,4]:
    # Let A = Vandermonde, b = solution
    A=vandermonde(t,k)
    b=load(t,f)
    # Find c using np.linalg.solve
    appr=np.zeros(len(t))
    c=np.linalg.solve(A.T@A,A.T@b)
    # Populate appr
    for n in range(0,k+1):
        appr=appr+c[n]*t**n
    # Calculate error using norm
    err=np.linalg.norm(A@c-b)
    err=np.round(err,3)
    string="k="+str(k)+" error="+str(err)
    # Plot appr for the current k
    plt.plot(t,appr,label=string)
plt.xlabel("x")
plt.ylabel("f(t) and appr(t)")
plt.title("true f(t) vs. approximations")
plt.legend()
plt.show()
```



[7]:
```
# Question answered: The value k=4 gives the best fit to the function as it␣
↪produces an approximation of the lowest measured error.
```