1) Investigate the local truncation error for the difference formula

$$f''(x) \approx \frac{f(x+3h) - 4f(x) + 3f(x-h)}{6h^2}.$$

We can find the local truncation error as $f''(x) - f''_{\text{approx}}(x)$. For this, it is most straightfoward to perform the taylor exansions, form the linear combinations according to the approximation formula, and take the difference between the exact derivative and the approximation. The taylor expansions are shown below:

$$\begin{aligned} f(x+3h) &= f(x) + 3hf'(x) + \frac{9}{2}h^2 f''(x) + \frac{9}{2}h^3 f^{(3)}(\xi_1) \\ f(x) &= f(x) \\ f(x-h) &= f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) - \frac{1}{6}h^3 f^{(3)}(\xi_2). \end{aligned}$$

Then, we see that

$$\frac{f(x+3h) - 4f(x) + 3f(x-h)}{6h^2} = f''(x) + h\left[\frac{3}{4}f^{(3)}(\xi_1) - \frac{1}{12}f^{(3)}(\xi_2)\right].$$

2) Consider a quadrature of the form

$$\int_{-1}^{1} |x|\, f(x)\, \mathrm{d}x = \frac{1}{4}\left[f(-1) + 2f(0) + f(1)\right].$$

Show that it is exact for any polynomial $f(x)$ of degree at most 3.

We want to show that for any polynomial of at most degree 3 the quadrature formula is exact. Since the integral operation is linear, we must check the polynomial basis $1, x, x^2, x^3, \ldots$ as $f(x)$ and determine when the integral is no longer exact. Notice that

$$\frac{1}{4}\left[(-1)^k + 2(0)^k + (1)^k\right] = \begin{cases} 1 & f(x) = 1 \\ 1/2 & k \equiv 0 \mod 2 \\ 0 & k \equiv 1 \mod 1 \end{cases}.$$

It is clear that the integral is zero for all odd power of x since $|x|\, x^k$ is an odd function being integrated over an even range. Thus, it remains to evaluate the integral for even powers of x.

$$\begin{aligned} \int_{-1}^{1} |x|\, 1\, \mathrm{d}x &= 1 \\ \int_{-1}^{1} |x|\, x^2\, \mathrm{d}x &= \frac{1}{2} \\ \int_{-1}^{1} |x|\, x^4\, \mathrm{d}x &= \frac{1}{3}. \end{aligned}$$

Indeed, it is seen that the quadrature formula is no longer exact for $k = 4$, meaning that it is only guranteed to be exact for polynomials of at most degree 3.

3) For the function $f(x) = \sin x$, use the forward-difference formula and backward-difference formula to determine $f'(a)$ at $a = 0.5$, , 0.6, 0.7 for $h = 0.1$, 0.05, 0.025, 0.0125, 0.00625. Calculate the exact derivatives exactly by directly calculating the derivative function. We define the order to be

$$\text{order}(h) = \log_2 \frac{\text{error}\,(2h)}{\text{error}\,(h)}.$$

The calculation is done in python using the following formulae. The forward and backward difference equations are

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$
$$f'(x) = \frac{f(x) - f(x-h)}{h},$$

respectively. The error is simply calculated as the absolute difference between the approximation for $f'(x)$ and the exact derivative $f'(x) = \cos x$.

The code to produce the tables is shown below, and the tables are shown below the code.

```
#!/usr/bin/env python

import numpy as np
import pandas as pd

def deriv(f,a,h):
    faph = f(a+h)
    fa   = f(a)
    famh = f(a-h)
    forward  = (faph - fa) / h
    backward = (fa - famh) / h
    return np.array([forward,backward])

def order(errors):
    orders = []
    for i in range(1,len(errors)):
        orders.append(np.log2(errors[i-1]/errors[i]))
    return orders

if __name__ == '__main__':

    f  = np.sin
    fp = np.cos
    h = np.array([0.1,0.05,0.025,0.0125,0.00625])

    results = {}

    for A in [0.5,0.6,0.7]:
        derivs = np.array([deriv(f,A,H) for H in h])
        exact  = fp(A)
        errors = abs(exact - derivs)
        forward_order  = order(errors[:,0])
        backward_order = order(errors[:,1])

        results = {}
```

```
    results['$h$'] = ['0.1','0.05','0.025','0.0125','0.00625']

    results[r'\makecell{Forward \\ Difference}'] = ['%.5f'%_ for _ in
derivs[:,0]]
    results[r'\makecell{Forward \\ Error}']  = ['%.5f'%_ for _ in
errors[:,0]]
    results[r'\makecell{Forward \\ Order}']  = ['-'] + ['%.5f'%_ for _
in order(errors[:,0])]

    results[r'\makecell{Backward \\ Difference}'] = ['%.5f'%_ for _ in
derivs[:,1]]
    results[r'\makecell{Backward \\ Error}'] = ['%.5f'%_ for _ in
errors[:,1]]
    results[r'\makecell{Backward \\ Order}']= ['-'] + ['%.5f'%_ for _
in  order(errors[:,1])]

    table = pd.DataFrame(results)

    A_str = str(A).replace('.','-')
    table.to_latex(buf='prob3_%s.tex'%A_str,
            index=False,
            escape=False,
            column_format='lcccccc',
            caption='Table for $a = %.1f$, where $f\'(%.1f) = %.5f$'%(A
,A,fp(A)),
            position='H')
```

Table 1: Table for $a = 0.5$, where $f'(0.5) = 0.87758$

h	Forward Difference	Forward Error	Forward Order	Backward Difference	Backward Error	Backward Order
0.1	0.85217	0.02541	-	0.90007	0.02249	-
0.05	0.86523	0.01235	1.04121	0.88920	0.01162	0.95294
0.025	0.87150	0.00608	1.02129	0.88348	0.00590	0.97725
0.0125	0.87456	0.00302	1.01082	0.88056	0.00297	0.98881
0.00625	0.87608	0.00150	1.00546	0.87908	0.00149	0.99445

Table 2: Table for $a = 0.6$, where $f'(0.6) = 0.82534$

h	Forward Difference	Forward Error	Forward Order	Backward Difference	Backward Error	Backward Order
0.1	0.79575	0.02958	-	0.85217	0.02683	-
0.05	0.81088	0.01446	1.03303	0.83910	0.01377	0.96260
0.025	0.81819	0.00714	1.01704	0.83231	0.00697	0.98187
0.0125	0.82179	0.00355	1.00865	0.82884	0.00351	0.99108
0.00625	0.82357	0.00177	1.00436	0.82709	0.00176	0.99557

Table 3: Table for $a = 0.7$, where $f'(0.7) = 0.76484$

h	Forward Difference	Forward Error	Forward Order	Backward Difference	Backward Error	Backward Order
0.1	0.73138	0.03346	-	0.79575	0.03091	-
0.05	0.74842	0.01642	1.02684	0.78063	0.01578	0.96966
0.025	0.75671	0.00813	1.01384	0.77281	0.00797	0.98528
0.0125	0.76080	0.00405	1.00703	0.76885	0.00401	0.99275
0.00625	0.76282	0.00202	1.00354	0.76685	0.00201	0.99640