Problem sheet 2

Please submit the question marked * to your tutor.

E2.1

Use elementary calculus to prove that

$$\max_{x\in[x_0,x_1]} (x-x_0)(x_1-x) = \frac{1}{4}(x_0-x_1)^2$$

and hence complete the proof of Corollary 2.2.

E2.2 *

Consider $f(x) = x^3 + 2x + 3$.

(a) Write down the linear polynomial p_1 that interpolates f.

(b) Use Corollary 2.2 to bound $|e(x)|$ over $x \in [0,1]$ where $e := f - p_1$. Compare your bound with the actual error, which can be found analytically in this case.

(c) Find the quadratic polynomial p_2 that interpolates f at x_0, x_1, and the additional point $x_2 = 2$.

(d) Next find the cubic polynomial p_3 which interpolates f at x_0, x_1, x_2 and the additional point $x_3 = -1$. Comment on the relation between f and p_3.

E2.3

The following function interpolates a given function f between the two points x_0 and x_1 (input arguments x_0 and x_1). p1 is a vector containing all the values of the function $p_1(x)$ at 100 equally spaced points in $[x_0, x_1]$.

In [1]:

```
import numpy as np
def f(x):
    return np.sqrt(x)

def linear_interp(x0,x1,mesh):
    vecone = np.ones(100,)
    p1 =  f(x0) + ((f(x1) - f(x0))/(x1-x0))*(mesh - x0)
    return p1

mesh = np.linspace(0,1,100)
p1 = linear_interp(0,1,mesh)
```

Write code to plot $p_1(x)$ and $f(x)$ (in different colours) on one graph, and then the error $e(x) := f(x) - p_1(x)$ on another graph.

In [2]:

```
import matplotlib.pyplot as plt
# fill in
```

E2.4*

In this exercise, numerically investigate

$$e_h := \max_{x\in[-h,h]} |f(x) - p_1(x)|$$

for $h > 0$, where the maximum is taken over 100 equally spaced points in $[-h, h]$.

Run your program for the case $f(x) = \exp(x)$ and draw up a table of e_h against h, for $h = 1/8, 1/16, 1/32, \cdots, 1/256$. Investigate the convergence rate as $h \to 0$ experimentally, by making the conjecture $e_h = Ch^\alpha$ where C and α are constants and then finding approximations to α.

Note that as in lectures, computing $\log_2(e_{2h}/e_h)$ will give approximations to α. This can be done either by editing the program or using a calculator.

Repeat the exercise for $f(x) = \sin x$. Explain your observations by appealing to the theory from lectures.

In [3]:

```
def f(x):
    return np.exp(x)

#takes as input h and returns the maximum error eh as defined above
def max_err_linear_interp(h):
    #fill in
    return eh
```