Practice 2.2

Practice sheets are not assessed. The intention is to use material from lectures in preparation for Competence tests and Assignments.

Here's some data to be used in this Practice sheet.

```
xdata = [-4.00, -3.11, -2.22, -1.33, -0.44, 0.44, 1.33, 2.22, 3.11, 4.00]
ydata = [9.37, 7.38, 4.36, 6.76, 7.17, 10.58, 18.31, 21.45, 30.99, 41.44]
```

Interpolation [CORE]

Create an interpolating function based on the data.

Plot this interpolating function and the data. (Hopefully this illustrates why interpolating is a bad idea!)

Curve fitting [CORE]

Find the best fit parameters a, b, c *and the associated uncertainties* $\sigma_a, \sigma_b, \sigma_c$ given the equation

$$y = ax^2 + bx + c$$

and the above data. Plot this fitted function and the data.

Integrating [EXTRA]

Integrate the interpolating function from -4 to $+4$.

Newton's method for complex arguments [HARD]

```
def newton_iters(f, x0, fprime=None, tol=1e-6, maxiter=50):
    if fprime is None:
        def fprime(x):
            h = 1e-6
            return (f(x+h)-f(x))/h
    x = x0
    for n in range(maxiter):
        x -= f(x)/fprime(x)
        if abs(f(x)) < tol:
            return n
    return n
```

The above implementation of Newton's method returns the **number of iterations** required to converge to a root. Find the number of iterations required to coverge for starting points in the *complex* plane $z = x + iy$. Use equation $z^6 + 4z^3 - 1$; interval $-2 < x < 2$ and $-2 < y < +2$; at least 500×500 points. Plot this using `plt.pcolormesh`, an example of which is given below:

```
def f(x,y):
    return np.exp(-x**2/10-y**2+x*y/2)

x = np.linspace(-4,4,101)
y = np.linspace(-3,3,101)

N, M = len(x), len(y)
A = np.zeros((N,M))
for n,xn in enumerate(x):
    for m, yn in enumerate(y):
        A[n,m] = f(xn,yn)

plt.pcolormesh(x,y,A.transpose())
plt.axis('equal'); plt.xlabel('x'); plt.ylabel('y'); plt.title('2D Gaussian')
plt.savefig('figs/pcolormesh.png'); plt.close('all')
```