Calculate Δp

Using the higher-order relatedness approach of Ohtsuki (2014), the change in the proportion of Cooperators p is proportional to

$$\Delta p \propto \sum_{k=0}^{n-1} \sum_{l=k}^{n-1} (-1)^{l-k} \binom{l}{k} \binom{n-1}{l} \left[(1-\rho_1)\rho_{l+1} a_k - \rho_1(\rho_l - \rho_{l+1}) b_k \right].$$

This document illustrates how to use the functions in the code repository to calculate Δp for a variety of p and homophily levels h.

First, define a dictionary of parameter values to match the figure in the main text:

In [1]:

```
parD = {'n': 8, 'tau': 5, 'W': 2, 'X': -1, 'Y': 3, 'Z': 0}
```

The matrix M is used to convert the partition probabilities $F_{n->\mathbf{n}}$ to the n-relatednesses parameters $\theta_{l->m}$. Matrices have been stored in `/results/matrix_M/` and can be conveniently read using the `read_matrix_M()` function.

In [2]:

```
import sys
sys.path.append('../functions/')
```

In [3]:

```
from my_functions import read_matrix_M

lm, partns, M_num, M_den = read_matrix_M('../results/matrix_M/matrix_M' + str(pa
rD['n']) + '.csv')
M = M_num / M_den # the matrix for converting F to theta
```

Let's explore a range of q values in the members recruit group-formation model. For this model, $h \equiv 1 - q$.

In [5]:

```
import numpy as np

qV = np.array([1, 0.75, 0.5, 0.2, 0])
hV = 1-qV
```

We will plot Δp against p.

In [6]:

```
pV = np.linspace(0, 1, 50)
```

To calculate $F_{n\to\mathbf{n}}$ in the members recruit model, we use

$$F_{n\to\mathbf{n}} = P(\mathbf{n}) = \left(\frac{\prod_{i=1}^{\Phi}(n_i-1)!}{(n-1)!} q^{\Phi-1}(1-q)^{n-\Phi} \right) \sum_{\vec{\mathbf{n}}\in\mathcal{N}} \sum_{\mathbf{m}\in\mathcal{M}} C(\mathbf{m},\vec{\mathbf{n}}) \prod_{k=1}^{\Phi-1} m_k.$$

The $\sum_{\vec{\mathbf{n}}\in\mathcal{N}} \sum_{\mathbf{m}\in\mathcal{M}} C(\mathbf{m},\vec{\mathbf{n}}) \prod_{k=1}^{\Phi-1} m_k$ term has been preculculated and stored in the repository.

In [7]:

```
sum_prod_mistakes_file = '../results/members_recruit/sum_product_mistakes/sum_pr
od_mistakes' + str(parD['n']) + '.csv'
```

The function `calc_delta_p_members_recruit()` calculates Δp

In [8]:

```
from my_functions import calc_delta_p_members_recruit as calc_delta_p


deltapM = [ [ calc_delta_p(parD, lm, partns, M, q, p, sum_prod_mistakes_file=sum
_prod_mistakes_file) for p in pV ] for q in qV ]
```

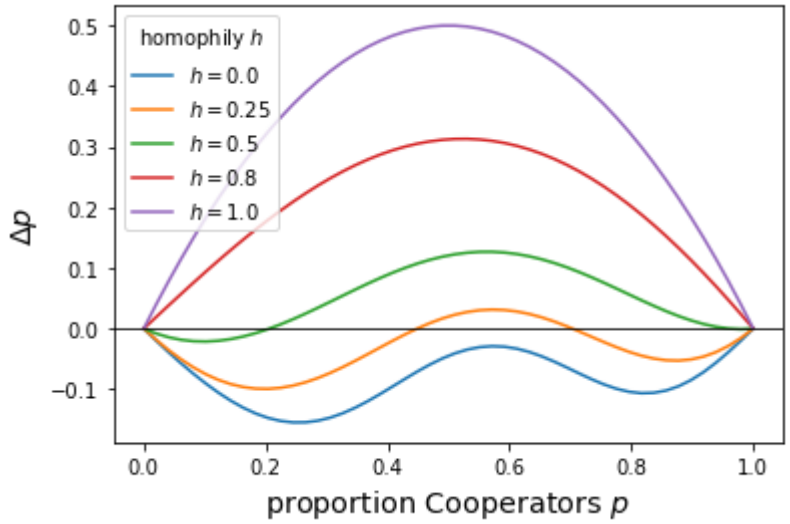
We can now plot the Δp vs p for our range of homophily levels.

In [9]:

```
import matplotlib.pyplot as plt

for h, deltapV in zip(hV, deltapM):
    plt.plot(pV, deltapV, label=r'$h = ' + str(h) + r'$')


plt.axhline(0, lw=1, alpha=0.8, color='black')
plt.xlabel(r'proportion Cooperators $p$', fontsize='x-large')
plt.ylabel(r'$\Delta p$', fontsize='x-large')
plt.legend(loc='best', title=r'homophily $h$')
plt.show()
```



In [ ]:

```
```