Getting Min. Eigenvalue with Shifted Power Iteration:

The underpinning theorem is that if λ and a nonzero vector v *are* an eigenpair of A. If α is any constant, then $\lambda - \alpha$ and v are an eigenpair of the matrix $(A - \alpha I)$.

So, we choose a value of shift α greater than the maximum eigenvalue. Shift the matrix A so the shifted matrix = $(A - \alpha I)$, then perform power iterations on the shifted matrix to get its max. eigenvalue and the corresponding vector.

The minimum eigen value of A will be equal to (α – *the max. eigenvalue of the shifted matrix).*

<u>MATLAB CODE:</u>

```
function [eigval, x, nit] = powershift(A, shift, etol, maxit)
  %this function calculates the min eigenvalue of a given matrix
  % INPUTS:
  %     A = the matrix
  %     shift = the value of the shift
  %             (should be greater than the max eigenvalue
  %              to properly get the minimum)
  %     etol = the error tolerance
  %     maxit = max number of iterations
  % OUTPUTS:
  %     eigval = the minimum eigenvalue
  %     x = the corresponding eigenvector
  %     nit = actual no. of iterations

  n = size(A)(1);
  shiftedA = (A - shift*eye(n));
  [eigval, x, nit] = powerit(shiftedA, etol, maxit);
   eigval = shift - eigval;
end
```

```
>> [min, vec, nit] = powershift(A, 20, 10^-6, 100)
min =  2.8393
vec =

  -0.26320
   0.70184
  -0.66192

nit =  77
```