Getting Min. Eigenvalue with Inverse Iteration:

The underlying concept in the inverse iteration is that: λmax of $[A]^{-1}$ = 1/λmin of [A].

λmax of [A]-1 can be determined by Power iterations on [A]-1. However, instead of constructing [A]-1, we solve a system of algebraic linear equations.

MATLAB CODE:

```
function [eigval, x, nit] = inverseit(A, etol,  maxit)
  % This function calculates the minimum eigen value and its
  % corresponding eigenvector by the inverse power iteration
  % algorithm
  % INPUTS:
  %     A = the matrix
  %     etol = error tolerance
  %     maxit = maximum number of iterations
  % OUTPUTS:
  %     eigval = the minimum eigenvalue
  %     x = the corrsponding eigenvector
  %     nit = the actual number of iterations

  [m,n] = size(A);
  x=ones(n,1);
  b=A\x;
  eigval=norm(b, 2);
  nit = 0;
  while nit < maxit
    nit = nit + 1;
    x = (1/eigval)*b;
    b = A\x;
    eigvalnew = norm(b, 2);
    if abs((eigvalnew - eigval)/eigvalnew) <= etol, break, end
    eigval = eigvalnew;
   end
   eigval = 1/eigval;
end
```

```
>> [eigval, x, nit] = inverseit(A, 10^-6, 100)
eigval =  2.8392
x =

   0.25498
  -0.70075
   0.66629

nit =  21
```