Getting Eigenvalues with Shifted Inverse Iteration:

The inverse power iteration algorithm starts with a shift value that is close to a certain eigenvalue corresponding to the desired eigenvector.
The closer the approximation to the eigenvalue is chosen, the faster the algorithm converges; however, incorrect choice of can lead to slow convergence or to the convergence to an eigenvector other than the one desired.
In practice, the method is used when a good approximation for the eigenvalue is known.

MATLAB CODE:

```
function [eigval, x, nit] = invshiftedit(A, shift, etol,  maxit)
  % This function calculates the eigenvalues and the corresponding
  % eigenvectors by the shifted inverse power algorithm, given a
  % shift value the function will converge to the nearest eigenvalue.
  %     INPUTS:
  %         A = the given matrix
  %         shift = an istimated shifting value
  %         etol = error tolerance
  %         maxit = maximum number of iterations
  %     OUTPUTS:
  %         eigval = the nearest eigenvalue to the given shift
  %         x = the corresponding eigenvectors
  %         nit = actual number of iterations
  n = size(A)(1);
  shiftedA = A - shift*eye(n);
  [eigval, x, nit] = inverseit(shiftedA, etol, maxit);
  if(shift > eigval) eigval = shift - eigval;
  else eigval = shift + eigval; end
end
```

```
>> [eigval, x, nit] = invshiftedit(A, 10, 10^-10, 1000)
eigval =  3.8704
x =
  -0.87878
   0.12186
   0.46142
nit =  69
>> [eigval, x, nit] = invshiftedit(A, 20, 10^-10, 1000)
eigval =  17.290
x =
  -0.40450
  -0.70318
  -0.58475
nit =  7
>> [eigval, x, nit] = invshiftedit(A, 1, 10^-10, 1000)
eigval =  2.8392
x =
   0.25323
  -0.70050
   0.66721
nit =  26
```