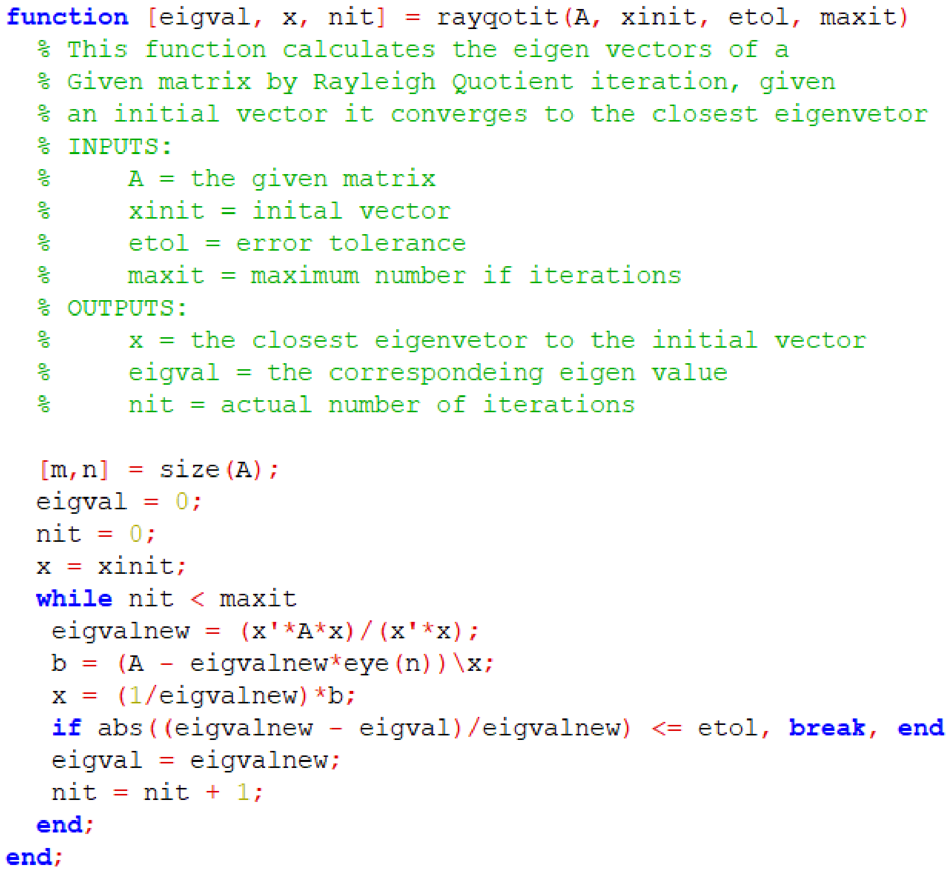
Getting Eigenvectors by Rayleigh Quotient Iterations:

Rayleigh quotient iteration is an eigenvalue algorithm which extends the idea of the inverse iteration by using the Rayleigh quotient to obtain increasingly accurate eigenvalue estimates.

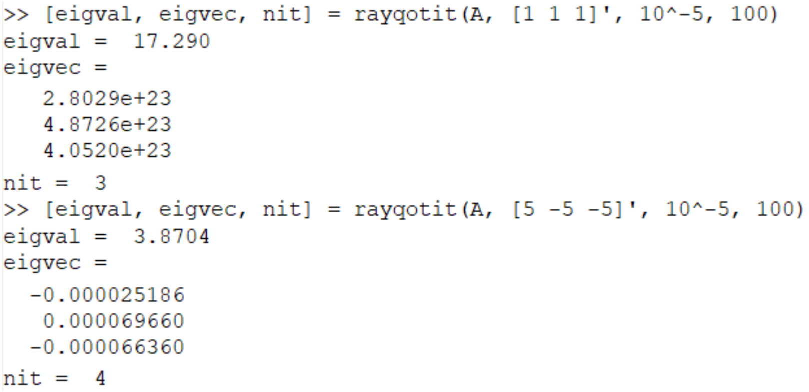
Rayleigh quotient iteration is an iterative method, that is, it delivers a sequence of approximate solutions that converges to a true solution in the limit. Very rapid convergence is guaranteed and no more than a few iterations are needed in practice to obtain a reasonable approximation. The Rayleigh quotient iteration algorithm converges cubically for Hermitian or symmetric matrices, **given** an initial vector that is sufficiently close to an eigenvector of the matrix that is being analyzed.

MATLAB CODE:



```
function [eigval, x, nit] = rayqotit(A, xinit, etol, maxit)
  % This function calculates the eigen vectors of a
  % Given matrix by Rayleigh Quotient iteration, given
  % an initial vector it converges to the closest eigenvetor
  % INPUTS:
  %     A = the given matrix
  %     xinit = inital vector
  %     etol = error tolerance
  %     maxit = maximum number if iterations
  % OUTPUTS:
  %     x = the closest eigenvetor to the initial vector
  %     eigval = the correspondeing eigen value
  %     nit = actual number of iterations

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



```
>> [eigval, eigvec, nit] = rayqotit(A, [1 1 1]', 10^-5, 100)
eigval =  17.290
eigvec =

   2.8029e+23
   4.8726e+23
   4.0520e+23

nit =  3
>> [eigval, eigvec, nit] = rayqotit(A, [5 -5 -5]', 10^-5, 100)
eigval =  3.8704
eigvec =

  -0.000025186
   0.000069660
  -0.000066360

nit =  4
```