Dobbs5

```
% create a matrix
rng(1000)
M = randi(21,100,100) - 10;
M = M - tril(M,-1) + triu(M,1)';

% get the eigenvalues
lambda_eig = eig(M)
```

```
lambda_eig = 100×1
10^2 ×
  -1.174005381039059
  -1.088601490800888
  -1.049637816344310
  -0.994771985235834
  -0.963741442378251
  -0.909105244320367
  -0.898432675043911
  -0.872927887217940
  -0.860407322462407
  -0.829970606317903
      ⋮
```

```
% approximate the eigenvalues
tridiagonal_M = phase1(M);
eig_M = phase2(tridiagonal_M);
lambda_approx = diag(eig_M)
```

```
lambda_approx = 100×1
10^2 ×
   1.438310013896559
  -1.174005381039053
   1.149386211731750
   1.142151445544184
  -1.088601490800890
  -1.049637816344309
   1.039732291156157
   1.032881409877130
   1.009559714248110
  -0.994771985235828
      ⋮
```

```
%check how these approximated eigenvalues compare to expected values...
num_similar = 0;
for i=1:100
    for j=1:100
        expected = lambda_eig(i);
        calculated = lambda_approx(j);
        if (abs(expected - calculated) < 1e-13*max(abs(expected),abs(calculated)))
```

```
            num_similar=num_similar+1;
        end
    end
end
num_similar % number of eigenvalues approximated to 1e-13 digits
```

```
num_similar =
   100
```

```
function M = phase1(M) % algorithm 26.1.
    m = size(M,1);
    for k=1:m-2
        x = M(k+1:m,k);
        e = eye(size(x,1));
        v_k = sign(x(1))*norm(x)*e(:,1) + x;
        v_k = v_k/norm(v_k);
        M(k+1:m,k:m) = M(k+1:m,k:m) - 2*v_k*(v_k' * M(k+1:m,k:m));
        M(1:m,k+1:m) = M(1:m,k+1:m) - 2*(M(1:m,k+1:m)*v_k)*v_k';
    end
end

function A = phase2(A) % algorithm 28.2
    [m,n] = size(A);
    if(m==1)
        return;
    end
    I = eye(m,n);
    for k=1:144
        mu = A(m,n);
        [Q,R] = qr(A-(mu.*I));
        A = R*Q + mu.*I;
        if(istriu(A) || istril(A))
            A = diag(diag(A));
            return;
        end
        for j=1:n-1
            if ((abs(A(j,j+1)) < 1e-15) || (abs(A(j+1,j)) < 1e-15))
                A(j,j+1) = 0;
                A(j+1,j) = A(j,j+1);
                A_k = A;
                A_k = [ phase2(A_k(1:j,1:j)), zeros(j,m-j) ;
                      zeros(m-j,j), phase2(A_k(j+1:m,j+1:m)) ];
                A = A_k;
                break;
            end
        end
    end
end
```