Multi-Step Problem

Table of Contents

Setup: 1
Part b: 1
Part c: 3

Setup:

```
% Format long to show adequate number of decimal points:
format long;

% Define the matrix we will be working with:
A = [7, -6, 3; 9, -8, 3; 9, -6, 1];
```

Part b:

Using an initial b_0 = (1 2 2)^T. Code up the power iteration method in MATLAB

```
% Subpart (ii): To ten decimal places, what does the algorithm predict is
% the eigenvalue and eigenvector of A after 5 iterations? What about 10
% iterations?

% After 5 iterations:

% Initialize b:
b = [1; 2; 2];
for i = 1:5
    b = A*b / norm(A*b);
end

% Output b and mu:
b
mu = transpose(b)*A*b

% After 10 iterations:

% Initialize b:
b = [1; 2; 2];
for i = 1:10
    b = A*b / norm(A*b);
end

% Output b and mu:
b
mu = transpose(b)*A*b

% Subpart (iii): How many iterations are needed to guarentee that the
```

```
% eigenvalue is within 10^-6 error of the true eigenvalue? What about
% within 10^-9 error? Measuring the error of the eigenvetor as error_b =
% max(abs(b_k-v)); where v is the eigenvector found in part (i), what is
% the error of the eigenvector in each case?

% Define the eigenvector for use in computing error_b:
v = [1/sqrt(3); 1/sqrt(3); 1/sqrt(3)];

% Iterations for eigenvaue within 10^-6 error:

% Initialize b, mu, and the iteration count:
b_0 = [1; 2; 2];
mu_0 = transpose(b_0/norm(b_0))*A*(b_0/norm(b_0));
iterations = 0;
while abs(4-mu_0) > 10^-6
    b_0 = (A*b_0) / norm(A*b_0);
    mu_0 = transpose(b_0)*A*b_0;
    iterations = iterations + 1;
end

% Display the iteration count and error_b:
iterations
error_b = max(abs(b_0-v))

% Iterations for eigenvaue within 10^-9 error:

% Initialize b, mu, and the iteration count:
b_0 = [1; 2; 2];
mu_0 = transpose(b_0/norm(b_0))*A*(b_0/norm(b_0));
iterations = 0;
while abs(4-mu_0) > 10^-9
    b_0 = (A*b_0) / norm(A*b_0);
    mu_0 = transpose(b_0)*A*b_0;
    iterations = iterations + 1;
end

% Display the iteration count and error_b:
iterations
error_b = max(abs(b_0-v))

% Output:


b =

   0.602994168679282
   0.564091319087070
   0.564091319087070


mu =

   4.465380249716234
```

```
b =

   0.576599976837906
   0.577725049963444
   0.577725049963443


mu =

   3.986354155887673


iterations =

    24


error_b =

     4.588367141789007e-08


iterations =

    34


error_b =

     4.480837922926639e-11
```

Part c:

```
% Code up the QR eigenvalue alogrithm in MATLAB.

% Subpart (i): To ten decimal places, what does this algorithm predict are
% the eigenvalues of A after 5 iterations. What about after 10 iterations?

% After 5 iterations:

% initialize A_0
A_0 = A;
for i = 1:5
    [Q,R] = qr(A_0);
    A_0 = inv(Q)*A_0*Q;
end

% Display the eigenvalues of A_0 (the diagonal entries of A_0):
A_0

% After 10 iterations:
```

```
% initialize A_0
A_0 = A;
for i = 1:10
    [Q,R] = qr(A_0);
    A_0 = inv(Q)*A_0*Q;
end

% Display the eigenvalues of A_0 (the diagonal entries of A_0):
A_0

% Subpart (ii): How many iterations are needed to guarentee that all three
% eigenvalues compute by the QR algorithm are within 10^-6 error of the
% true eigenvalues? What about within 10^-9 error?

% Within 10^-6 error:

% initialize A_0 and the iteration count:
A_0 = A;
iterations = 0;
while (abs(4-A_0(1,1))>10^-6) & (abs(-2-A_0(2,2))>10^-6) & (abs(-2-
A_0(3,3))>10^-6)
    [Q,R] = qr(A_0);
    A_0 = inv(Q)*A_0*Q;
    iterations = iterations + 1;
end

% Display the number of iterations:
iterations

% Within 10^-9 error:

% initialize A_0 and the iteration count:
A_0 = A;
iterations = 0;
while (abs(4-A_0(1,1))>10^-9) & (abs(-2-A_0(2,2))>10^-9) & (abs(-2-
A_0(3,3))>10^-9)
    [Q,R] = qr(A_0);
    A_0 = inv(Q)*A_0*Q;
    iterations = iterations + 1;
end

% Display the number of iterations:
iterations

% Output:


A_0 =

   3.857088690496707  -7.796413543530077  16.818909075458247
   0.011303440533959  -2.015046092269428   0.032458367736341
   0.055007759858410  -0.073221230994027  -1.842042598227277
```

```
A_0 =

   4.004560189352390  -8.024901017915026 -16.659717432391378
  -0.000347795627530  -1.999535182361842   0.000964962744339
   0.001811132574674  -0.002420520268554  -2.005025006990547


iterations =

    19


iterations =

    29
```

Published with MATLAB® R2023a