MAE 290A HW4

Contents

Problem 1

```
A0 = [1 4 7; 2 5 8; 3 6 9];
err = 1;
u = [1; 1; 1];
mu = 0;
i = 0;
while err>1e-10, %power method
    v = A0*u;
    u = v/norm(v,2);
    mu1 = mu;
    mu = v'*A0*v;
    err = norm(mu-mu1,2);
    i = i+1;
end
s = u %associated eigenvector
lambdabig = u\v %largest eigenvalue
i %number of iterations
shift = 25; %shift to find smallest eigenvalue
err = 1;
u = [1; 1; 1];
i = 0;
while err>1e-10, %shifted power method
    v = (A0-shift*eye(3))*u;
    u = v/norm(v,2);
    mu1 = mu;
    mu = v'*A0*v;
    err = norm(mu - mu1,2);
    i = i+1;
end
s = u %associated eigenvector
lambdasmall = shift - u\v %smallest eigenvalue
i %number of iterations
```

```
s =

    0.4645
    0.5708
    0.6770


lambdabig =
```

```
   16.1168


i =

    13


s =

    0.8829
    0.2395
   -0.4039


lambdasmall =

   -1.1168


i =

    32
```

Problem 2

```
Ap = A0;
bit=1;
i = 0;
U = eye(3);
while bit > 0; %making sure resulting matrix is upper triangular
    [Q,R] = qr(Ap);
    Ap = R*Q;
    if abs(Ap(2,1)) > 1e-6
        partialbit1 = 1;
    else
        partialbit1 = 0;
    end
    if abs(Ap(3,1)) > 1e-6
        partialbit2 = 1;
    else
        partialbit2 = 0;
    end
    if abs(Ap(3,2)) > 1e-6
        partialbit3 = 1;
    else
        partialbit3 = 0;
    end
    bit = partialbit1+partialbit2+partialbit3;
    i = i+1;
    U = U*Q;
end
T = R*Q
```

```
U
Ap
i
eigval = diag(T);
```

```
T =

   16.1168    4.8990   -0.0000
   -0.0000   -1.1168    0.0000
   -0.0000   -0.0000    0.0000


U =

   -0.4645    0.7858    0.4082
   -0.5708    0.0868   -0.8165
   -0.6770   -0.6123    0.4082


Ap =

   16.1168    4.8990   -0.0000
   -0.0000   -1.1168    0.0000
   -0.0000   -0.0000    0.0000


i =

     7
```

Problem 3

```
[P,H]=hess(A0); %hessenberg decomposition
Ap = H; %
bit = 1;
i = 0;
while bit > 0; %making sure resulting matrix is upper triangular
    [Q,R] = qr(Ap);
    Ap = R*Q;
    if abs(Ap(2,1)) > 1e-6
        partialbit1 = 1;
    else
        partialbit1 = 0;
    end
    if abs(Ap(3,1)) > 1e-6
        partialbit2 = 1;
    else
        partialbit2 = 0;
    end
    if abs(Ap(3,2)) > 1e-6
        partialbit3 = 1;
    else
        partialbit3 = 0;
```

```
    end
    bit = partialbit1+partialbit2+partialbit3;
    i = i+1;
    U = U*Q;
end
T = R*Q
U = P*U
Ap
i
```

```
T =

   16.1168   -4.8990   -0.0000
    0.0000   -1.1168    0.0000
         0   -0.0000   -0.0000


U =

    0.9479    0.1706    0.2691
    0.0085   -0.8577    0.5140
   -0.3185    0.4850    0.8145


Ap =

   16.1168   -4.8990   -0.0000
    0.0000   -1.1168    0.0000
         0   -0.0000   -0.0000


i =

     7
```

Problem 4

```
for i=1:3
    err = 1;
    mu = eigval(i); %pulling eigenvalues from schur decomposition
    u = [1; 1; 1]; %initial guess vector
    while abs(err)>1e-6
        v = (A0 - mu*eye(3))\u; %calculating possible eigenvector
        u = v/norm(v,2); %normalizing possible eigenvector
        mu1 = u'*A0*u %calculating next possible eigenvalue
        err = mu - mu1;
        mu=mu1;
    end
    S(:,i) = u
    i;
end
S
Si = inv(S);
```

```
Adiag = Si*A0*S
```

```
mu1 =

   16.1168


S =

   -0.4645    0.8829   -0.7892
   -0.5708    0.2395    0.2133
   -0.6770   -0.4039    0.5759


mu1 =

   -1.1168


S =

   -0.4645    0.8829   -0.7892
   -0.5708    0.2395    0.2133
   -0.6770   -0.4039    0.5759

Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 3.700743e-17.

mu1 =

   3.5527e-15


S =

   -0.4645    0.8829   -0.7892
   -0.5708    0.2395    0.2133
   -0.6770   -0.4039    0.5759


S =

   -0.4645    0.8829   -0.7892
   -0.5708    0.2395    0.2133
   -0.6770   -0.4039    0.5759


Adiag =

   16.1168   -0.0000   -6.7098
    0.0000   -1.1168    1.1080
    0.0000    0.0000   -0.0000
```

Problem 5

```
A1 = [1 3; 2 4];
A2 = [1 4; 2 5; 3 6];
A3 = [1 2 3; 4 5 6];
[U1,S1,V1] = svd(A1)
[U2,S2,V2] = svd(A2)
[U3,S3,V3] = svd(A3)
```

```
U1 =

   -0.5760   -0.8174
   -0.8174    0.5760


S1 =

    5.4650         0
         0    0.3660


V1 =

   -0.4046    0.9145
   -0.9145   -0.4046


U2 =

   -0.4287    0.8060    0.4082
   -0.5663    0.1124   -0.8165
   -0.7039   -0.5812    0.4082


S2 =

    9.5080         0
         0    0.7729
         0         0


V2 =

   -0.3863   -0.9224
   -0.9224    0.3863


U3 =

   -0.3863   -0.9224
   -0.9224    0.3863


S3 =

    9.5080         0         0
```

```
         0    0.7729         0


V3 =

   -0.4287    0.8060    0.4082
   -0.5663    0.1124   -0.8165
   -0.7039   -0.5812    0.4082
```

Problem 6

```
%refer to fact 4.38 in numerical renaissance
%refer to definition of frobenius norm and singular values
```