EE547 (PMP) Midterm - Winter 2015

Table of Contents

prepared by Paul Adams

Initialization

```
function midterm()

close all
orient landscape
opengl('save', 'software')
format shortG
set(0, 'defaultTextInterpreter', 'latex');
numerical_precision = 1e-9;
syms s x x_1 x_2 x_3 u t t_0
```

Problem 1

```
x = [x_1; x_2; x_3];
f = [-9*x_1 - 4*x_2 - (1+x_3)*x_3 + sin(x_3) + sin(u);...
    (x_2*x_3 - 4)*x_1 - 10*sin(x_2) + 3*cos(x_3) + x_3^2*sin(u);...
    9*x_1 + (x_1^2 - 4)*x_3 - 10*x_2 + u];
g = [x_1 + x_2*x_3 + sin(u);...
    x_2 + x_1*x_3 + u^2;...
    x_3 + x_2*x_3 + cos(u)];
render_latex(['f = ' latex(f)], 12, 0.7)
render_latex(['g = ' latex(g)], 12, 0.7)
```

$$f = \begin{bmatrix} \sin(u) - 4\,x_2 - 9\,x_1 + \sin(x_3) - x_3\,(x_3+1) \\ 3\,\cos(x_3) - 10\,\sin(x_2) + {x_3}^2\,\sin(u) + x_1\,(x_2\,x_3 - 4) \\ u + 9\,x_1 - 10\,x_2 + x_3\,({x_1}^2 - 4) \end{bmatrix}$$

$$g = \begin{bmatrix} x_1 + \sin(u) + x_2\,x_3 \\ u^2 + x_2 + x_1\,x_3 \\ x_3 + \cos(u) + x_2\,x_3 \end{bmatrix}$$

The state-space matrices are found using

$$\mathbf{A} = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{x^{eq},u^{eq}} \quad \mathbf{B} = \left.\frac{\partial f}{\partial u}\right|_{x^{eq},u^{eq}} \quad \mathbf{C} = \left.\frac{\partial g}{\partial \mathbf{x}}\right|_{x^{eq},u^{eq}} \quad \mathbf{D} = \left.\frac{\partial g}{\partial u}\right|_{x^{eq},u^{eq}}$$

where,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

```
xeq = [-0.1 0.1 -0.2];
ueq = 0;
A = subs(jacobian(f, x), [x_1, x_2, x_3, u], [xeq, ueq]);
B = subs(jacobian(f, u), [x_1, x_2, x_3, u], [xeq, ueq]);
C = subs(jacobian(g, x), [x_1, x_2, x_3, u], [xeq, ueq]);
D = subs(jacobian(g, u), [x_1, x_2, x_3, u], [xeq, ueq]);
render_latex(['\mathbf{A} = ' latex(simplify(A))], 12, 0.75)
render_latex(['\mathbf{B} = ' latex(simplify(B))], 12, 0.75)
render_latex(['\mathbf{C} = ' latex(simplify(C))], 12, 0.75)
render_latex(['\mathbf{D} = ' latex(simplify(D))], 12, 0.75)
A = double(A);
B = double(B);
C = double(C);
D = double(D);
```

$$\mathbf{A} = \begin{bmatrix} -9 & -4 & \cos\left(\frac{1}{5}\right) - \frac{3}{5} \\ -\frac{201}{50} & \frac{1}{50} - 10\,\cos\left(\frac{1}{10}\right) & 3\,\sin\left(\frac{1}{5}\right) - \frac{1}{100} \\ \frac{226}{25} & -10 & -\frac{399}{100} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 \\ \frac{1}{25} \\ 1 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & -\frac{1}{5} & \frac{1}{10} \\ -\frac{1}{5} & 1 & -\frac{1}{10} \\ 0 & -\frac{1}{5} & \frac{11}{10} \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Given the state-space matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$. The transer function is found using

$$\hat{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$$

NOTE since the symbolic representation of G using

```
sI_A = s*eye(size(A)) - A;
G = C*inv(sI_A)*B;
```

results in unreadable expressions, use *ss2tf* instead.

```
[num, den] = ss2tf(A, B, C, D)


num =

            1      24.012      168.72      384.04
            0       -0.26     -8.4994     -34.743
            0       1.092       30.91      219.17


den =

            1       22.92      151.24      316.96
```

Proper rational? The degree of the numerators are, at most, equal to the degree of the denominator. Therefore, the transfer functions are proper rational.

As noted above, the system is proper rational.

```
poles = roots(charpoly(A));     % eig is preferable, but is used below...
if all(real(poles) < 0)
    fprintf('Since the poles (eigenvalues) each have negative real parts, \n the s
else
    disp('The system is NOT BIBO stable.')
end
```

```
Since the poles (eigenvalues) each have negative real parts,
 the system is BIBO stable.
```

Find the eignevalues of A

NOTE use *eig* to get more accurate results

```
lambda = eig(A)
```

```
lambda =

      -13.396 +          0i
      -4.7621 +    0.99175i
      -4.7621 -    0.99175i
```

Since the eigenvalues of A are distinct, the Jordan form is simply the eigenvalues of A along the diagonal. The tranformation matrix, Q, is found by finding a solution to the homogenuous equation

$$(\mathbf{A} - \lambda_i \mathbf{I})\, q_i = \mathbf{0}$$

```
J = diag(lambda);
Q = zeros(size(A));
for i = 1:length(lambda)
    Q(:, i) = null(A - lambda(i)*eye(size(A)));
end
J_ = array2table(J)
Q_ = array2table(Q)
```

```
J_ =
```

J1	J2	J3
-13.396	0+0i	0+0i
0	-4.7621+0.99175i	0+0i
0	0+0i	-4.7621-0.99175i

```
Q_ =
```

```
       Q1              Q2                    Q3
    _______    __________________    __________________

    0.65802     0.050564+0i           0.050564+0i
    0.73756    -0.025377-0.1026i     -0.025377+0.1026i
    0.15173       0.2967-0.94776i       0.2967+0.94776i
```

For an LTI system, the State Transition Matrix $\Phi(t, t_0)$ using the Jordan form is given by

$$\Phi(t, t_0) = \mathbf{Q}e^{\mathbf{J}(t-t_0)}\mathbf{Q}^{-1}$$

```
Phi = Q*expm(J*(t))*inv(Q);
Phi = vpa(Phi, 2)
% render_latex(['\Phi(t, 0) = ' latex(Phi)], 12, 0.75)
```

```
Phi =

[      exp(-13.0*t)*(0.51 + 5.5e-17*i) + exp(-4.8*t)*(cos(0.99*t) + sin(0.99*t)*1.0
[ exp(-13.0*t)*(0.57 + 6.1e-17*i) + exp(-4.8*t)*(cos(0.99*t) + sin(0.99*t)*1.0*i)*
[ exp(-13.0*t)*(0.12 + 1.3e-17*i) + exp(-4.8*t)*(cos(0.99*t) + sin(0.99*t)*1.0*i)*
```

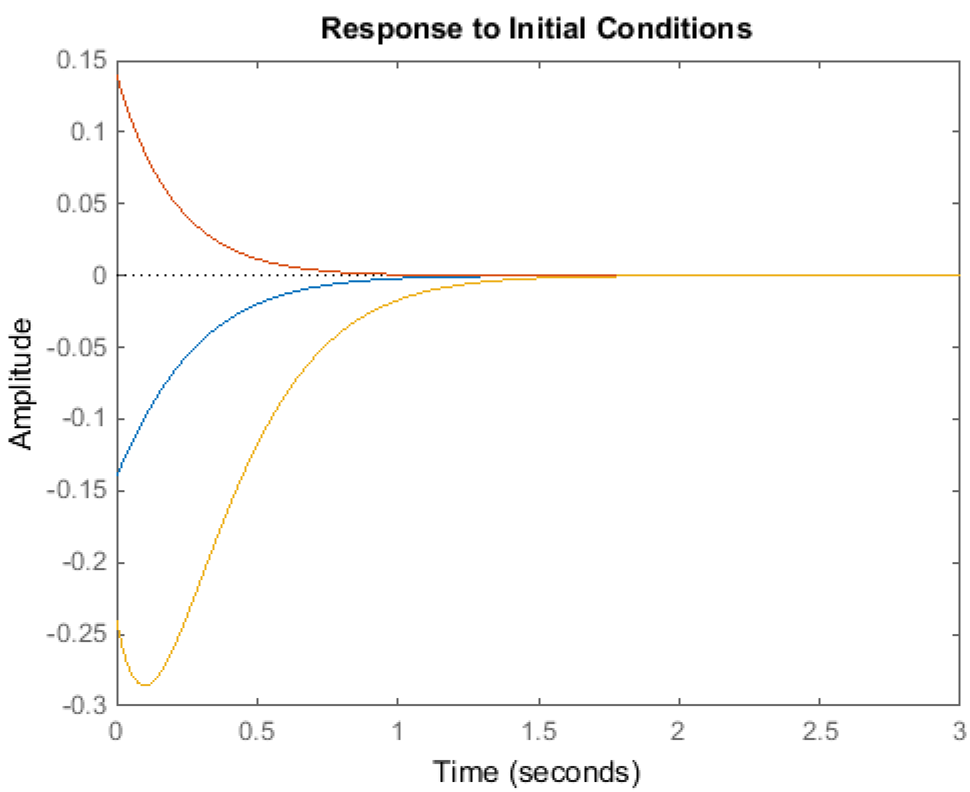
```
x0 = [-0.1; 0.1; -0.2];
sys = ss(A, B, C, D);
[y_zir, t, x_zir] = initial(sys, x0, 3);
```

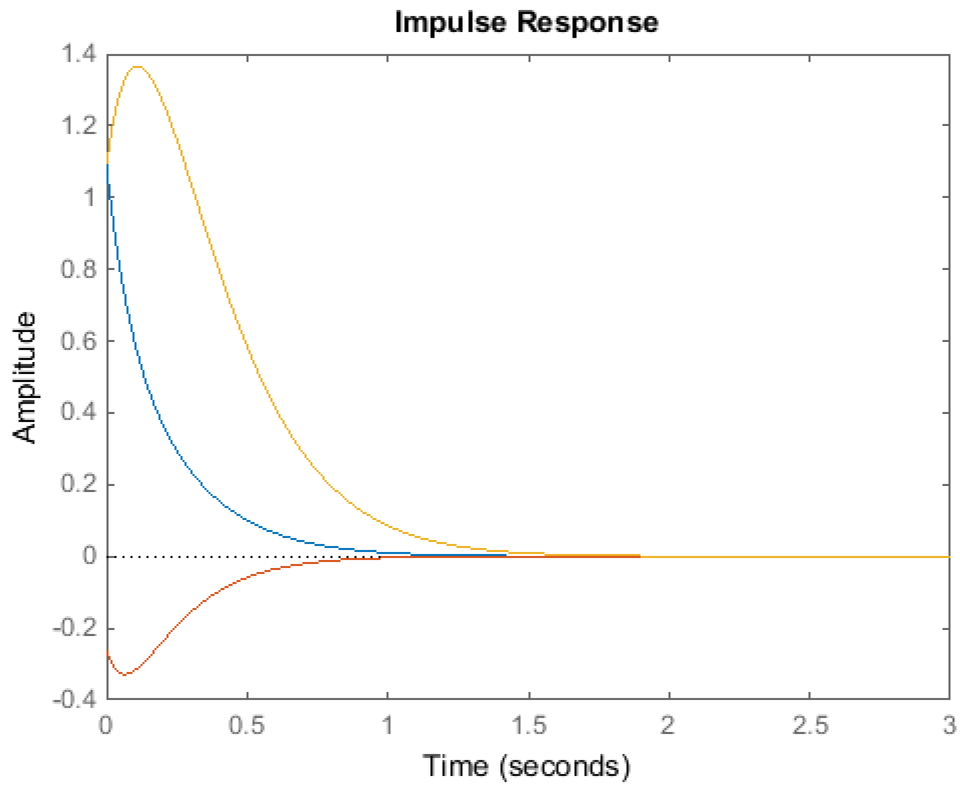
```
Q = eye(size(A, 1));
% The system is asymptotically stable if the solution to the Lyapunov
% equation, P, is positive definite.
P = lyap(A, Q)
% Choose the condition that every eigenvalue of P be positive as the
% condition for the positive-definiteness of P
if all(eig(P) > 0)
    disp('P is positive definite, therefore, the system is asymptotically stable')
else
    disp('The system is NOT asymptotically stable')
end
% compute the impulse response
[y_zsr, ~, x_zsr] = impulse(sys, 3);
```

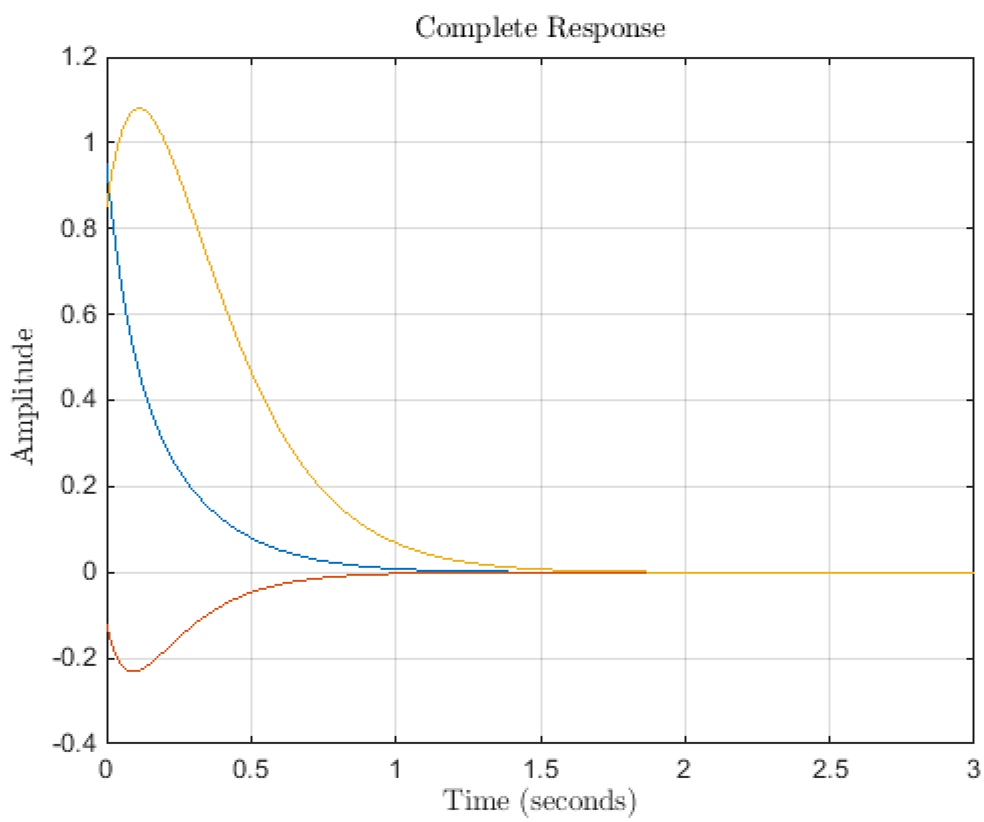
```
P =

     0.071146    -0.025168       0.10429
    -0.025168     0.056699     -0.065104
      0.10429    -0.065104       0.52477

P is positive definite, therefore, the system is asymptotically stable
```

Plots

```
figure, grid on
initial(sys(1), sys(2), sys(3), x0, 3);
figure, grid on
impulse(sys(1), sys(2), sys(3), 3);
figure, grid on
plot(t, y_zir + y_zsr)
title('Complete Response')
xlabel('Time (seconds)'); ylabel('Amplitude')
```




Impulse Response
Amplitude
Time (seconds)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
0.5
1
1.5
2
2.5
3


Complete Response
Amplitude
Time (seconds)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
0.5
1
1.5
2
2.5
3

Problem 2

Given the state-space matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$. The transer function is found using

$$\hat{G}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$$

```
A = [-5 -9 4; 2 -9 2; 9 -10 -8];
B = [-1; 2; -3];
C = eye(size(A));
D = zeros(3, 1);
sI_A = s*eye(size(A)) - A;
G = C*inv(sI_A)*B;
render_latex(['\hat{G_1}(s) = ' latex(simplify(G))], 16, 1.2)
```

$$\hat{G}_1(s) = \begin{bmatrix} -\frac{s^2+47\,s+370}{s^3+22\,s^2+159\,s+522} \\ \frac{2\,\left(s^2+9\,s-40\right)}{s^3+22\,s^2+159\,s+522} \\ -\frac{3\,s^2+71\,s+512}{s^3+22\,s^2+159\,s+522} \end{bmatrix}$$

```
A_ = [-82 -8 54; -174 -33 130.5; -138 -16 93];
B_ = [2; -7; 0];
C_ = [-4 -1 3.5; 1 0 -0.5; 2 1 -2];
D_ = zeros(3, 1);
sI_A = s*eye(size(A_)) - A_;
G_ = C_*inv(sI_A)*B_;
render_latex(['\hat{G_2}(s) = ' latex(simplify(G_))], 16, 1.2)
```

$$\hat{G}_2(s) = \begin{bmatrix} -\frac{s^2+47\,s+370}{s^3+22\,s^2+159\,s+522} \\ \frac{2\,\left(s^2+9\,s-40\right)}{s^3+22\,s^2+159\,s+522} \\ -\frac{3\,s^2+71\,s+512}{s^3+22\,s^2+159\,s+522} \end{bmatrix}$$

```
G = ss2tf(A, B, C, D);
G_ = ss2tf(A_, B_, C_, D_);
if max(max(G - G_)) > numerical_precision
    disp('Systems are not zero-state equivalent')
else
    fprintf('Systems realize the same transfer function, therefore, \n they are ze
end
```

```
Systems realize the same transfer function, therefore,
 they are zero-state equivalent.
```

Problem 3

```
A = [-4 3 -1 10 3;...
    -6 -9 5 7 -6;...
    -8 -5 2 -9 2;...
    -4 -5 6 -1 2;...
    -6 8 -8 -4 2];
```

The characteristic polynomial of $\mathbf{A}$ is given by

$$\Delta(\lambda) = \det s\mathbf{I} - \mathbf{A}$$

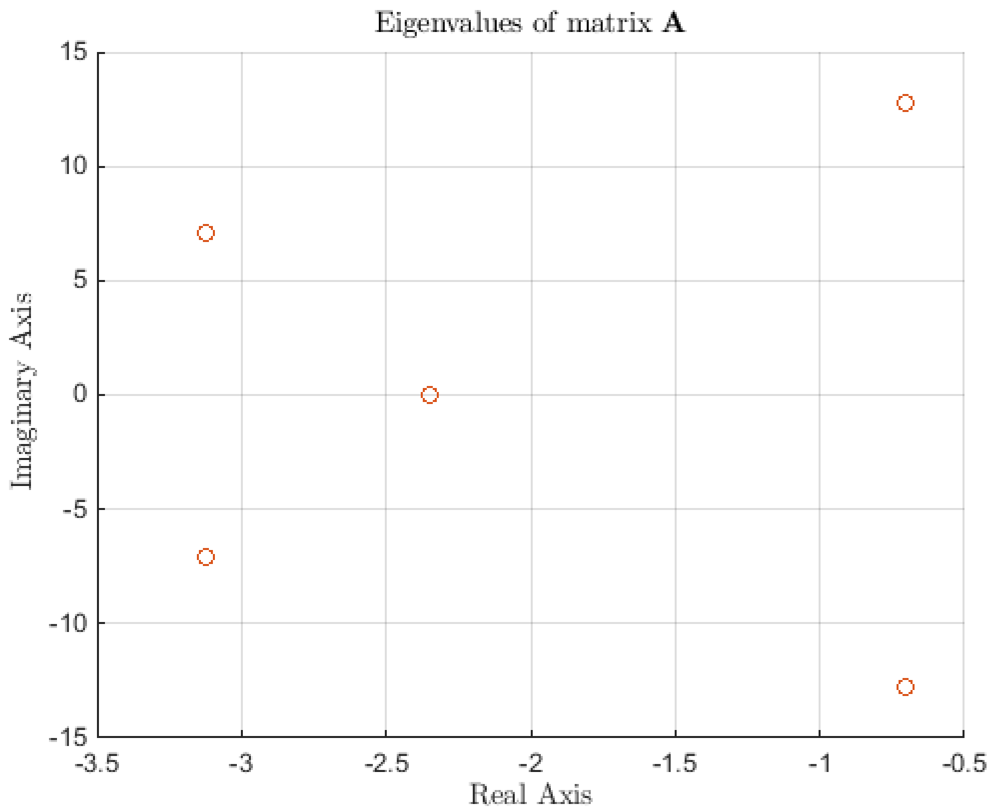
And the eigenvalues of $\mathbf{A}$ are the roots of the characteristic polynomial

```
sI_A = s*eye(size(A)) - A;
CharPoly = det(sI_A);
render_latex(['\Delta(\lambda) = ' latex(CharPoly)], 12, 0.5)
lambda = roots(sym2poly(CharPoly))
```

$$\Delta(\lambda) = s^5 + 10\,s^4 + 251\,s^3 + 1658\,s^2 + 12462\,s + 23160$$

```
lambda =

     -0.70106 +     12.797i
     -0.70106 -     12.797i
      -3.1238 +     7.0886i
      -3.1238 -     7.0886i
      -2.3504 +          0i
```

```
scatter(real(lambda), imag(lambda)); hold on
scatter(real(eig(A)), imag(eig(A))); hold off
xlabel('Real Axis'); ylabel('Imaginary Axis');
title('Eigenvalues of matrix $\mathbf{A}$')
```



Since the eigenvalues of A are distinct, the Jordan form is simply the eigenvalues of A along the diagonal. The tranformation matrix, Q, is found by finding a solution to the homogenuous equation

$$(\mathbf{A} - \lambda_i \mathbf{I})\, q_i = \mathbf{0}$$

```
lambda = eig(A);
J = diag(lambda);
Q = zeros(size(A));
for i = 1:length(lambda)
    Q(:, i) = null(A - lambda(i)*eye(size(A)));
end
J_ = array2table(J)
Q_ = array2table(Q)
```

J_ =

J1	J2	J3	J4	J5
-0.70175+12.798i	0+0i	0	0+0i	0+
0+0i	-0.70175-12.798i	0	0+0i	0+
0+0i	0+0i	-2.3508	0+0i	0+
0+0i	0+0i	0	-3.1229+7.0865i	0+
0+0i	0+0i	0	0+0i	-3.1229-

```
Q_ =

          Q1                     Q2                Q3               Q4
    __________________     __________________     ________     _________________     _

     -0.33956+0i            -0.33956+0i           0.066064       0.4554+0i
     -0.30505-0.32656i      -0.30505+0.32656i     -0.70522     -0.31672+0.56509i      -
      0.35779-0.49983i       0.35779+0.49983i     -0.64957     -0.21487+0.35643i      -
    -0.069349-0.43875i     -0.069349+0.43875i     0.077968     0.007789+0.11375i      0
      0.28216+0.1739i        0.28216-0.1739i       0.26512      0.35229+0.25029i
```

```
if max(max(A*Q - Q*J)) > numerical_precision
    disp('The transformation is not valid.')
else
    disp('The transformation is valid.')
end
```

```
The transformation is valid.
```

$$f_2(A) = A^5 + 10A^4 + 251A^3 + 1658A^2 + 12462^A + 23160I_{5x5}$$

The solution equates $f(\mathbf{A}) = f(\lambda)$ to $h(\lambda)$ where $h(\lambda) = \beta_0 + \beta_1\lambda + ... + \beta_{n-1}\lambda^{n-1}$

However, I discovered the hard way that solving linear equations with this $\mathbf{A}$ given the numeric precision available in Matlab leads to badly scaled matrices which factor with error.

Instead, use Jordan form of $\mathbf{A}$, then $\mathbf{A}^k = \mathbf{Q}^{-1}\mathbf{J}^k\mathbf{Q}$.

Since the eigenvalues of $\mathbf{A}$ are distinct, $\mathbf{J}^k = \lambda^k\mathbf{I}_5$, where $k = 5, 4, 3, 2, 1, 0$

```
k = 5:-1:0;
b = [1 10 251 1658 12462 23160];
f2 = zeros(size(A));
f2_ = zeros(size(A));
for i = 1:length(k)
    A_to_the_k = inv(Q)*(diag(lambda.^k(i)))*Q;
    f2 = f2 + A_to_the_k*b(i);
    f2_ = f2_ + A^k(i)*b(i); % the direct solution
end
array2table(f2)
```

```
ans =

            f21                          f22                          f23
    _______________________      _______________________      _______________________
```

```
  -1.6735e-10+2.2499e-12i     1.2733e-11-2.2792e-11i    -6.8157e-11+1.1963e-10i
   1.6371e-11+2.7497e-11i      3.638e-11-4.3833e-12i     1.1556e-10+4.3013e-11i
   7.7246e-11-1.5661e-10i    -7.2104e-11-1.3435e-10i    -8.3674e-11-1.0376e-11i
  -1.6564e-10+5.286e-11i     -3.5956e-11+2.1892e-10i    -7.7625e-11+4.1946e-11i
   8.8919e-11+3.2014e-10i     1.5312e-10-1.0695e-11i     1.2888e-10+4.3656e-11i
```

```
if max(max(f2_ - f2)) > numerical_precision
    disp('The transformation is not valid.')
else
    disp('The transformation is valid.')
end
```

```
The transformation is valid.
```

Published with MATLAB® R2014b