LTI Discrete-Time Systems: Response Modes

Table of Contents

Introduction

In the case of a linear, time-invariant, discrete dynamical system,

$$\begin{cases} x(k+1) &=& Ax(k)+Bu(k)\,, \quad x(0)=x_0 \\ y(k) &=& Cx(k)+Du(k) \end{cases}$$

the state movement takes the form

$$x(k) \;=\; A^k x_0 + \sum_{j=0}^{k-1} A^{k-(j+1)} B\, u(j)\,, \quad k>0$$

Then, the output movement is

$$y(k) \;=\; C\,A^k x_0 + \sum_{j=0}^{k-1} C\,A^{k-(j+1)} B\, u(j) + D\, u(k)\,, \quad k>0$$

In both expressions, the matrix term A^k appears. It is possible to evaluate A^k by highlighting the contribution of each distinct eigenvalue of the matrix$A$, employing so-called **response modes** (a sort of "*matrix partial fraction expansion* ").

The Response Modes

Let $\lambda_1, \dots, \lambda_\sigma$ denote the σ distinct eigenvalues of the matrix A, and $n_i \quad (i = 1\,,\,2\,,\;\dots\sigma)$ is the algebraic multiplicity of such eigenvalues - of course $\sum_{i=1}^{\sigma} n_i = n$, where $n$is the degree of the matrix characteristic polynomial $p_A(\lambda) = \det(\lambda\, I - A)$.

Consider the following two cases:

1. There is no null eigenvalue;
2. At least one eigenvalue is zero.

First Scenario: No Null Eigenvalues

The following expression applies in this case

$$A^k = \sum_{i=1}^{\sigma}\left[A_{i0}\cdot\lambda_i^k\cdot 1(k) + \sum_{l=1}^{n_i-1} A_{il}\cdot k\cdot(k-1)\cdots(k-l+1)\cdot\lambda_i^{k-l}\cdot 1(k-l)\right] \quad (1)$$

where the matrices

$$A_{il} = \frac{1}{l!}\frac{1}{(n_i-1-l)!}\lim_{z\to\lambda_i}\left\{\frac{d^{n_i-1-l}}{dz^{n_i-1-l}}\left[(z-\lambda_i)^{n_i}(zI-A)^{-1}\right]\right\} \quad n_i \geq 1\,,\; l = 0\,,\,1\,,\;\dots n_i - 1 \quad (2)$$

are the **response modes**.

When all eigenvalues of A are distinct, i.e. $\sigma = n; n_i = 1, i = 1, \dots, n$, then

$$A^k = \sum_{i=1}^{n} A_i\lambda_i^k \qquad A_i = \lim_{z\to\lambda_i}\left[(z-\lambda_i)(zI-A)^{-1}\right]\; i = 1\,,\,2\,,\;\dots n \quad (3)$$

In the special case of distinct eigenvalues of A, a different characterisation of the response mode matrices exists

$$A_i = v_i\cdot\tilde{v}_i^{\top} \quad (4)$$

where

- $(\lambda_i I - A)\cdot v_i = 0$: v_i right eigenvector associated with λ_i;
- $\tilde{v}_i^{\top}\cdot(\lambda_i I - A) = 0$: $\tilde{v}_i^{\top}$ left eigenvector associated with λ_i .

Second Scenario: At Least One Null Eigenvalue

If an eigenvalue λ_j of A is zero, then the terms in the expression of A^k, corresponding to the zero eigenvalue must be computed as follows

$$A_{j0}\cdot\delta(k) + \sum_{l=1}^{n_j-1} A_{jl}\, l!\,\delta(k-l) \quad (5)$$

where

$$A_{jl} = \frac{1}{l!}\frac{1}{(n_j - 1 - l)!}\lim_{z\to 0}\left\{\frac{d^{n_j-1-l}}{dz^{n_j-1-l}}\left[z^{n_j}(zI - A)^{-1}\right]\right\} \quad n_j \geq 1\,,\ l = 0\,,\ 1\,,\ \ldots n_j - 1 \qquad (6)$$

References

For details on the response modes, how to derive the formulas in this live script and for further examples, please refer to:

Antsaklis, P. J., Michel A. N. , *Linear systems*, Birkhäuser Boston, 2006.

available also as e-book in the University Library.

Before You Start

Work your way through the sections of this live scripts, using the links to navigate through the various addressed topics. You can also browse the live script as PDF or HTML document. Please pay attention: the PDF and the HTML version of a live script do not preserve the capability of inserting a MATLAB command or a piece of code directly in the live script, obtaining the result directly in the script itself.

What you will learn:

- How to compute the response mode matrices using MATLAB, in both the scenarios depicted above;
- How to proficiently solve either symbolic and numeric problems, involving the computation of the response modes using MATLAB.

```
clear
close all
clc
sympref('HeavisideAtOrigin',1); % initial setting
```

Computation of the Response Mode Matrices Using MATLAB

Special Case: All Eigenvalues Are Distinct

Consider the LTI discrete-time autonomous system

$$x(k+1) = A\,x(k)$$

where

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.5 & 1 \\ -1 & 0 & 3 \end{bmatrix}$$

Let us compute the response mode matrices associated with the matrix power A^k. Moreover, given a generic state

$$x_0 = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \quad \alpha, \beta, \gamma \in \mathbb{R}$$

compute the free state movement, starting from the initial state $x(0) = x_0$.

```
syms alpha beta gamma % the three parameters, used to define the generic initial state
x0 = [alpha; beta; gamma] % the initial state
```

x0 =

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$$

```
A = sym([1, 0, 0; 0, 1/2, 1;-1, 0, 3]) % assigning the matrix A as symbolic object
```

A =

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & 1 \\ -1 & 0 & 3 \end{pmatrix}$$

First check: does matrix$A$ admit diagonal form?

```
[eigVects, lambdaSET] = eig(A)
```

eigVects =

$$\begin{pmatrix} 0 & 2 & 0 \\ \frac{2}{5} & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

lambdaSET =

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$$

```
% extracting the main diagonal from the square matrix lambdaSET, creating
% a column vector and assigning the col. vector to lambdaSET
```

```
lambdaSET = diag(lambdaSET)
```

```
lambdaSET =
```

$$\begin{pmatrix} 3 \\ 1 \\ \frac{1}{2} \end{pmatrix}$$

How many distinct eigenvalues?

```
numel(unique(lambdaSET))
```

```
ans = 3
```

How many independent eigenvectors correspond to each distinct eigenvalue?

```
rank(eigVects)
```

```
ans = 3
```

The algebraic and the geometric multiplicity of each eigenvalue are equal each other and equal to one. So the matrix A does admit diagonal form.

Now we have two alternatives:

- to compute the response modes using the Z-transform, according to Eq. (3);
- to use the alternative characterisation, based on right and left eigenvectors, as reported in Eq. (4).

Using the Z-transform

```
lambda1 = lambdaSET(1);
lambda2 = lambdaSET(2);
lambda3 = lambdaSET(3);

syms z n
zIA = z*eye(size(A))-A;
zIA_inv = inv(zIA);
```

First response mode, according to Eq. (3)

```
% evaluation of a limit using the symbolic toolbox
A_1 = limit((z-lambda1)*zIA_inv,z, lambda1 ) %#ok<*MINV> % code suppressing warning mes
```

```
A_1 =
```

$$\begin{pmatrix} 0 & 0 & 0 \\ -\frac{1}{5} & 0 & \frac{2}{5} \\ -\frac{1}{2} & 0 & 1 \end{pmatrix}$$

The contribution $A_1\,\lambda_1^k$ is

```
A_1*lambda1^n
```

ans =

$$\begin{pmatrix} 0 & 0 & 0 \\ -\frac{3^n}{5} & 0 & \frac{2\,3^n}{5} \\ -\frac{3^n}{2} & 0 & 3^n \end{pmatrix}$$

The second response mode matrix and the corresponding contribution to the matrix power A^k are

```
A_2 = limit((z-lambda2)*zIA_inv,z, lambda2 )
```

A_2 =

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}$$

```
A_2*lambda2^n
```

ans =

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}$$

Finally, the third response mode and its contribution to A^k are

```
A_3 = limit((z-lambda3)*zIA_inv,z, lambda3 )
```

A_3 =

$$\begin{pmatrix} 0 & 0 & 0 \\ -\frac{4}{5} & 1 & -\frac{2}{5} \\ 0 & 0 & 0 \end{pmatrix}$$

```
A_3*lambda3^n
```

ans =

$$\begin{pmatrix} 0 & 0 & 0 \\ -\frac{4\left(\frac{1}{2}\right)^n}{5} & \left(\frac{1}{2}\right)^n & -\frac{2\left(\frac{1}{2}\right)^n}{5} \\ 0 & 0 & 0 \end{pmatrix}$$

Now we can write down the matrix power A^k as sum of the response mode contriibutions, as follows

```
Ak = A_1 * lambda1^n + A_2*lambda2^n+A_3*lambda3^n
```

```
Ak =
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 1-\frac{3^n}{5}-\frac{4\left(\frac{1}{2}\right)^n}{5} & \left(\frac{1}{2}\right)^n & \frac{2\,3^n}{5}-\frac{2\left(\frac{1}{2}\right)^n}{5} \\ \frac{1}{2}-\frac{3^n}{2} & 0 & 3^n \end{pmatrix}$$

Let us compare with the matrix power obtained directly applying the inverse Z-transform

```
ApowZ = z*zIA_inv;
Apow = iztrans(ApowZ)
```

```
Apow =
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 1-\frac{3^n}{5}-\frac{4\left(\frac{1}{2}\right)^n}{5} & \left(\frac{1}{2}\right)^n & \frac{2\,3^n}{5}-\frac{2\left(\frac{1}{2}\right)^n}{5} \\ \frac{1}{2}-\frac{3^n}{2} & 0 & 3^n \end{pmatrix}$$

Let us compare the two resulting A^k expressions

```
isequaln(Ak, Apow)
```

```
ans = logical

   1
```

The Free State Movement

At this point, we can evaluate the expression of the free state movement, starting from the given (symbolic) initial state

```
freeMov_x = Apow * x0
```

```
freeMov_x =
```

$$\begin{pmatrix} \alpha \\ \left(\frac{1}{2}\right)^n \beta - \gamma \left(\frac{2\left(\frac{1}{2}\right)^n}{5} - \frac{2\,3^n}{5}\right) - \alpha \left(\frac{4\left(\frac{1}{2}\right)^n}{5} + \frac{3^n}{5} - 1\right) \\ 3^n \gamma - \alpha \left(\frac{3^n}{2} - \frac{1}{2}\right) \end{pmatrix}$$

Response Modes Computation Exploiting Left & Right Eigenvectors

Let us consider the same LTI system and determine the response modes using left and right eigenvectors, according to Eq. (4).

Let's start with the right eigenvectors.

```
[rV, ~] = eig(A); % the right eigenvectors

% let's verify: rV columns are the right eigenvectors iff
syms res [3 3]
for c = 1:3
    res(:,c) = A*rV(:,c) - lambdaSET(c)*rV(:,c);
end
disp(res)
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Now the left eigenvectors:

the following property holds true

$$\tilde{v}_i^\top (\lambda_i I - A) = 0 \quad \Longrightarrow \quad \left[\tilde{v}_i^\top (\lambda_i I - A)\right]^\top = 0 \quad \Longrightarrow \quad \left(\lambda_i I - A^\top\right)\tilde{v}_i = 0$$

```
% the left eigenvectors of A are the right eigenvectors of the transpose
% matrix A'

[lV, ~] = eig(A');
lV = lV'; % the right eigenvectors are row vectors
```

Verify that the `lV` column vectors are effectively the right eigenvectors of A:

```
syms resL [3 3]
for r = 1:3
    resL(r,:) = lV(r,:)*A - lambdaSET(r)*lV(r,:);
end
disp(res)
```

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Let's check the orthonormality

$$\tilde{v}_i^{\top} v_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and, in case, adjust the norm of the left (or right) eigenvectors

```
shouldBeEyematrix = lV*rV
```

```
shouldBeEyematrix =
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -\frac{5}{2} \end{pmatrix}$$

```
diagElm = diag(shouldBeEyematrix); % extract the main diagonal as a vector
shouldBeNullMatrix = shouldBeEyematrix- diag(diagElm); % subtract the main diagonal

% check: if shouldBeNullMatrix is not a null matrix, then through an error
% and stop the execution of the code
assert(isequal(shouldBeNullMatrix, ...
        sym(zeros(size(shouldBeNullMatrix)))));
% please note: shouldBeNullMatrix is a symbolic matrix, so also the null
%              matrix has to be a symbolic variable (otherwise, the
%              comparison fails! Remove the symbolic casting command
%              sym().. and see the error message! In fact, in that case you would
%              compare a symbolic variable with a NOT symbolic one, so the
%              comparison fails.

if ~isequal(diagElm, ones(size(diagElm)))
    for r=1:3
        lV(r,:) = lV(r,:)/diagElm(r);
    end % for r
end
% let us check one more time
lV*rV
```

```
ans =
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

```
lV
```

```
lV =
```

$$\begin{pmatrix} -\frac{1}{2} & 0 & 1 \\ \frac{1}{2} & 0 & 0 \\ -\frac{4}{5} & 1 & -\frac{2}{5} \end{pmatrix}$$

Remark

According to

$$Q := [v_1 \mid v_2 \mid \cdots \mid v_n] \implies P = Q^{-1} = \begin{bmatrix} \widetilde{v}_1^\top \\ \vdots \\ \widetilde{v}_n^\top \end{bmatrix}; \widetilde{v}_i^\top v_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

(please, refer to the class material, L3-p19), a more effiicient and quick way to obtain the left eigenvectors $\widetilde{v}_i^\top$ as orthonormal vectors with respect to the right eigenvectors v_j is simply

```
lV_alt = inv(rV)
```

```
lV_alt =
```

$$\begin{pmatrix} -\frac{1}{2} & 0 & 1 \\ \frac{1}{2} & 0 & 0 \\ -\frac{4}{5} & 1 & -\frac{2}{5} \end{pmatrix}$$

Finally, we can compute the response mode matrices, by simply evaluating

```
A1_alt = rV(:,1)*lV(1,:) %#ok<*NASGU>
```

```
A1_alt =
```

$$\begin{pmatrix} 0 & 0 & 0 \\ -\frac{1}{5} & 0 & \frac{2}{5} \\ -\frac{1}{2} & 0 & 1 \end{pmatrix}$$

```
A2_alt = rV(:,2)*lV(2,:)
```

```
A2_alt =
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{pmatrix}$$

```
A3_alt = rV(:,3)*lV(3,:)
```

```
A3_alt =
```

$$\begin{pmatrix} 0 & 0 & 0 \\ -\frac{4}{5} & 1 & -\frac{2}{5} \\ 0 & 0 & 0 \end{pmatrix}$$

As you can easily check by comparison, the response modes `A1`, `A2`, and `A3` are equal respectively to `A1_alt`, `A2_alt`, and `A3_alt`.

General Case: Multiple Eigenvalues

Consider the LTI discrete-time autonomous system

$$x(k+1) = A\,x(k)$$

where

$$A = \begin{bmatrix} 3 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Let us compute the response mode matrices associated with the matrix power A^k. Moreover, given a generic state

$$x_0 = \begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} \quad \alpha\,,\,\beta\,,\,\gamma\,,\,\delta \in \mathbb{R}$$

compute the free state movement, starting from the initial state $x(0) = x_0$.

```
clear
clc
syms alpha beta gamma delta  % the parameters, used to define
                             % the generic initial state
x0 = [alpha; beta; gamma; delta] % the initial state
```

```
x0 =
```

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{pmatrix}$$

```
A = sym([3, -1, 1, 1 ;...
         1, 1, -1, -1;...
         0, 0, 2, 0; ...
         0, 0, 0, 0])
```

A =

$$\begin{pmatrix} 3 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Is the Matrix Diagonalisable?

First check: does matrix$A$ admit diagonal form?

```
[eigVects, lambdaSET, p] = eig(A);
% extracting the main diagonal from the square matrix lambdaSET, creating
% a column vector and assigning the col. vector to lambdaSET
lambdaSET = diag(lambdaSET)
```

lambdaSET =

$$\begin{pmatrix} 0 \\ 2 \\ 2 \\ 2 \end{pmatrix}$$

What is the vector `p`?

Please, refer to the MATLAB `help`

```
help symbolic/eig
```

and refer also to this section of the current document, for a practical use of the vector p.

How many distinct eigenvalues?

```
lambdaU = unique(lambdaSET);
numel(lambdaU)
```

ans = 2

Evaluating the Algebraic Multiplicity of the Distinct Eigenvalues

What about the algebraic multiplicity?

```
algMult = zeros(numel(lambdaU),1); % preallocate an array
for j=1:numel(lambdaU)
    algMult(j) = sum(double(lambdaSET)==double(lambdaU(j)));
end % for j
disp(lambdaU)
```

$$\begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

```
disp(algMult)
```

```
     1
     3
```

Summarising, we have only two distinct eigenvalues:

- the first distinct eigenvalue $\lambda_1 = 0$ has algebraic multiplicity equal to 1;
- the second distinct eigenvalue $\lambda_2 = 2$ has algebraic multiplicity equal to 3.

Evaluating the Geometric Multiplicity of the Distinct Eigenvalues

What about the geometric multiplicity? How to use the vector p?

```
disp(eigVects)
```

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$$

```
disp(lambdaSET)
```

$$\begin{pmatrix} 0 \\ 2 \\ 2 \\ 2 \end{pmatrix}$$

```
disp(p)
```

```
     1     2
```

The **vector of indices** p shows that:

- `p(1) = 1`, so the first eigenvector (the first column of `eigVects`) corresponds to the first element of `lambdaSET`, with eigenvalue $\lambda_1 = 0$;
- `p(2) = 2`, so the second eigenvector (the second column of `eigVects`) corresponds to the second element of `lambdaSET` with eigenvalue $\lambda_2 = 2$.

Algebraic & Geometric Multiplicity of Eigenvalues vs Diagonal Form

The findings about the algebraic and geometric multiplicity of the eigenvalues allow us to state:

- The eigenvalue $\lambda_1 = 0$, that occurs once, has one linearly independent eigenvector (the eigenvalue λ_1 has algebraic multiplicity 1 and geometric multiplicity 1).
- The eigenvalue $\lambda_2 = 2$, that occurs three times, has one linearly independent eigenvectors (the eigenvalue $\lambda_2 = 2$ has algebraic multiplicity 3 and geometric multiplicity 1).

The matrix A **does not admit a diagonal form** (indeed, the matrix admits a Jordan canonical form). Moreover, there is a null eigenvalue. So, to determine the expression of the response mode matrices, we have to rely on the Eq. (2) and Eq. (6).

Computing the Response Modes

```
syms z n
zIA = z*eye(size(A))-A;
zIA_inv = inv(zIA);
```

The Response Mode Associated to the Null Eigenvalue

Associated to the eigenvalue $\lambda_1 = 0$, the response mode A_1 can be computed as follows (according to Eq. (6))

$$A_{10} = \lim_{z \to 0} [z \cdot (zI - A)^{-1}]$$

```
A_10 = limit(z*zIA_inv,z, lambdaU(1) )
```

A_10 =

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The contribution to the matrix power A^k is (according to Eq. (5))

```
A_10 * kroneckerDelta(n)
```

ans =

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_{n,0} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_{n,0} \end{pmatrix}$$

The Response Modes Associated to the Multiple Eigenvalue

According to Eq. (2), associated to the eigenvalue $\lambda_2 = 2$ we have 3 different response mode matrices

$$A_{20} = \frac{1}{0!}\frac{1}{(3-1)!}\lim_{z\to 2}\left\{\frac{d^{3-1}}{dz^{3-1}}[(z-2)^3(zI-A)^{-1}]\right\} = \frac{1}{2!}\lim_{z\to 2}\left\{\frac{d^2}{dz^2}[(z-2)^3(zI-A)^{-1}]\right\}$$

```
D0 = (((z-lambdaU(2))^3) * zIA_inv);
D1 = diff( D0, z, 1);
D2 = diff( D1, z, 1);
% D2 = diff( (((z-lambdaU(2))^3) * zIA_inv), z, 2);
A_20 = limit( D2 , z, lambdaU(2) )/ factorial(2)
```

A_20 =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The contribution to the matrix power A^k is (according to Eq. (1))

```
A_20*(lambdaU(2)^(n))
```

ans =

$$\begin{pmatrix} 2^n & 0 & 0 & 0 \\ 0 & 2^n & 0 & -2^n \\ 0 & 0 & 2^n & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A_{21} = \frac{1}{1!}\frac{1}{(3-2)!}\lim_{z\to 2}\left\{\frac{d^{3-2}}{dz^{3-2}}[(z-2)^3(zI-A)^{-1}]\right\} = \frac{1}{1!}\lim_{z\to 2}\left\{\frac{d}{d}[(z-2)^3(zI-A)^{-1}]\right\}$$

```
A_21 = limit( D1 , z, lambdaU(2) )/ factorial(1)
```

A_21 =

$$\begin{pmatrix} 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The contribution to the matrix power A^k is (according to Eq. (1))

```
simplify(A_21*(nchoosek(n, 1)*(lambdaU(2)^(n-1)))*factorial(1))
```

```
ans =
```

$$\begin{pmatrix} 2^{n-1}\,n & -2^{n-1}\,n & 2^{n-1}\,n & 2^{n-1}\,n \\ 2^{n-1}\,n & -2^{n-1}\,n & -2^{n-1}\,n & 2^{n-1}\,n \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$A_{22} = \frac{1}{2!}\frac{1}{(3-3)!}\lim_{z\to 2}\left\{\frac{d^{3-3}}{dz^{3-3}}[(z-2)^3(zI-A)^{-1}]\right\} = \frac{1}{2!}\lim_{z\to 2}\left[(z-2)^3(zI-A)^{-1}\right]$$

```
A_22 = limit( D0 , z, lambdaU(2) )/ (factorial(2))
```

```
A_22 =
```

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The contribution to the matrix power A^k is (according to Eq. (1))

```
simplify(A_22*(nchoosek(n, 2)*(lambdaU(2)^(n-2)))*factorial(2))
```

```
ans =
```

$$\begin{pmatrix} 0 & 0 & 2^{n-1}\binom{n}{2} & 0 \\ 0 & 0 & 2^{n-1}\binom{n}{2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The Matrix Power

Finally, the matrix power A^k is

```
Ak = simplify(...
     ( A_10 * kroneckerDelta(n) ) + ...
     ( A_20 * (lambdaU(2)^n ) )   + ...
     ( A_21 * (lambdaU(2)^(n-1) ) * nchoosek(n,1) *factorial(1) ) + ...
     ( A_22 * (lambdaU(2)^(n-2) ) * nchoosek(n,2) *factorial(2) ))
```

Ak =

$$\begin{pmatrix} \frac{2^n\,(n+2)}{2} & -\sigma_1 & \frac{2^n\left(n+\binom{n}{2}\right)}{2} & \sigma_1 \\ \sigma_1 & -\frac{2^n\,(n-2)}{2} & -\frac{2^n\left(n-\binom{n}{2}\right)}{2} & \frac{2^n\,n}{2}-2^n+\delta_{n,0} \\ 0 & 0 & 2^n & 0 \\ 0 & 0 & 0 & \delta_{n,0} \end{pmatrix}$$

where

$$\sigma_1 = 2^{n-1}\,n$$

Alternative way: using the Z-transform to compute directly A^k

```
Ak_alt = simplify(iztrans(z*zIA_inv))
```

Ak_alt =

$$\begin{pmatrix} \frac{2^n\,(n+2)}{2} & -\sigma_1 & \frac{2^n\left(2\,n+\binom{n-1}{2}-1\right)}{2} & \sigma_1 \\ \sigma_1 & -\frac{2^n\,(n-2)}{2} & \frac{2^n\left(\binom{n-1}{2}-1\right)}{2} & \sigma_1-2^n+\delta_{n,0} \\ 0 & 0 & 2^n & 0 \\ 0 & 0 & 0 & \delta_{n,0} \end{pmatrix}$$

where

$$\sigma_1 = \frac{2^n\,n}{2}$$

Are the two expressions equal?

```
simplify(expand(Ak-Ak_alt))
```

```
ans =
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

```
simplify(ztrans(Ak)-ztrans(Ak_alt))
```

```
ans =
```

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Yes, they are describing the same matrix power A^k.

Summary

Using this live script you learnt:

- How to compute the response mode matrices using MATLAB, either in the case of all distinct eigenvalues, either when multiple eigenvalues exist;
- How to proficiently solve either symbolic and numeric problems, involving the computation of the response modes using MATLAB.