Module 3 - Eigenvalues and Eigenvectors

ME3001 - Mechanical Engineering Analysis

Mechanical Engineering

Tennessee Technological University

These notes came from Mike Renfro. Thanks Mike.

Solution of Matrix Eigenvalue Problem

- Standard Matrix Eigenvalue Problem
  - Introduction
  - Other Forms of the Standard Eigenvalue Problem
  - Solvability of the Standard Eigenvalue Problem
- General Eigenvalue Problem
  - Introduction
  - General Eigenvalue Problem Example
- Eigenvalue Solutions in MATLAB
  - Standard Eigenvalue Problems
  - General Eigenvalue Problems

Part I

Solution of Matrix Eigenvalue Problem

Standard Matrix Eigenvalue Problem: Introduction

Consider a system of equations in algebraic form

$$\begin{aligned}(a_{11}-\lambda)x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= 0\\ a_{21}x_1 + (a_{22}-\lambda)x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= 0\\ &\vdots\\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + (a_{nn}-\lambda)x_n &= 0\end{aligned}$$

This is not a normal system of linear algebraic equations we're used to. For one, there are n equations, but $n+1$ unknowns (the x_i values, and also λ). This particular system of equations is known as *the standard eigenvalue problem*.

Form 1

The three forms shown are all algebraically equivalent. Any system of equations that can be expressed in these forms is a standard eigenvalue problem.

$$\begin{bmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-\lambda \end{bmatrix} \left\{ \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix} \right\} = \left\{ \begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \end{matrix} \right\}$$

Form 2

$$\left(\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}\right) \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}$$

$$([A] - \lambda [I]) \{x\} = \{0\}$$

Form 3

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \lambda \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

$$[A]\{x\} = \lambda\{x\}$$

Solvability of the Standard Eigenvalue Problem

Recall form 2 of the standard eigenvalue problem:

$$([A] - \lambda [I]) \{x\} = \{0\}$$

This system of equations has a solution for values of λ that cause the determinant of the coefficient matrix to equal 0, that is:

$$|[A] - \lambda [I]| = 0$$

Characteristic Equation

Expanding out all the terms of the previous determinant

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}-\lambda \end{vmatrix} = 0$$

yields a long polynomial in terms of λ. This polynomial will be nth order, and will therefore have n roots, each of which may be real or complex.

General Eigenvalue Problem: Introduction

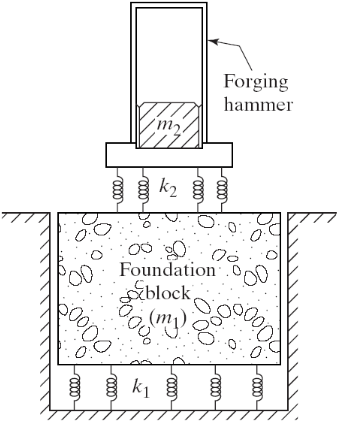
Many physical systems do not automatically present themselves as a standard eigenvalue problem, even though they can be reformatted as a standard eigenvalue problem. The form of a *general eigenvalue problem* is

$$[A]\{x\} = \lambda [B]\{x\}$$

where $[A]$ and $[B]$ are symmetric matrices of size $n \times n$.

General Eigenvalue Problem Example

A forging hammer of mass m_2 is mounted on a concrete foundation block of mass m_1. The stiffnesses of the springs underneath the forging hammer and the foundation block are given by k_2 and k_1, respectively.



General Eigenvalue Problem Example

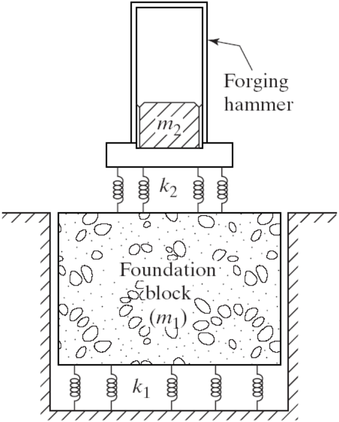
The system undergoes simple harmonic motion at one of its natural frequencies ω. That is:

$$x_1(t) = \cos(\omega t + \phi_1)$$

$$x_2(t) = \cos(\omega t + \phi_2)$$

$$a_1(t) = -\omega^2 x_1(t)$$

$$a_2(t) = -\omega^2 x_2(t)$$



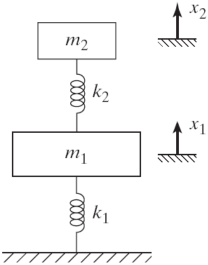
General Eigenvalue Problem Example

Each mass in the system obeys Newton's second law of motion, that is:

$$\Sigma F = ma$$

Forces on the foundation block:

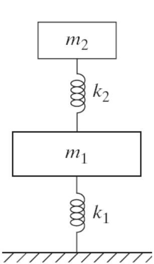
- forces from the lower springs, which counteracts motion in the x direction at an amount $-k_1 x_1$
- forces from the upper springs, which act according to the amount of relative displacement of the masses m_1 and m_2: $-k_2(x_1 - x_2)$



General Eigenvalue Problem Example

The equilibrium equation for the foundation mass is then

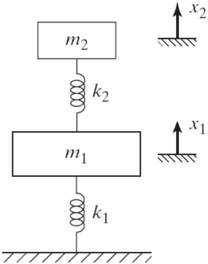
$$
\begin{aligned}
\Sigma F &= ma \\
-k_1 x_1 - k_2(x_1 - x_2) &= m_1 a \\
(-k_2 - k_1)x_1 + k_2 x_2 &= m_1 a \\
(-k_2 - k_1)x_1 + k_2 x_2 &= -m_1 \omega^2 x_1 \\
(k_1 + k_2)x_1 - k_2 x_2 &= m_1 \omega^2 x_1
\end{aligned}
$$



General Eigenvalue Problem Example

Similarly, the equilibrium equation for the forging hammer mass is

$$-k_2x_1 + k_2x_2 = m_2\omega^2x_2$$



General Eigenvalue Problem Example

So the two equations of motion are

$$\begin{aligned}(k_1+k_2)x_1 - k_2x_2 &= m_1\omega^2 x_1\\ -k_2x_1 + k_2x_2 &= m_2\omega^2 x_2\end{aligned}$$

or in matrix form

$$\begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \omega^2 \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}$$

This is a general eigenvalue problem

$$[A]\{x\} = \lambda[B]\{x\}$$

where $[A]$ is the spring matrix, $\{x\}$ is the vector of x values, λ is ω^2, and $[B]$ is the mass matrix.

Eigenvalue Solutions in MATLAB: Standard Problems

The design of a mechanical component requires that the maximum principal stress to be less than the material strength. For a component subjected to arbitrary loads, the principal stresses σ are given by the solution of the equation

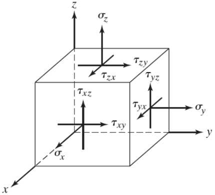
$$\begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} \end{bmatrix} \begin{Bmatrix} l_x \\ l_y \\ l_z \end{Bmatrix} = \sigma \begin{Bmatrix} l_x \\ l_y \\ l_z \end{Bmatrix}$$

where the σ values represent normal stresses in the x, y, and z directions, and the τ values represent shear stresses in the xy, xz, and yz planes. The l values represent direction cosines that define the principal planes on which the principal stress occurs.

Eigenvalue Solutions in MATLAB: Standard Problems

Determine the principal stresses and principal planes in a machine component for the following stress condition

$$\begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} \end{bmatrix} = \begin{bmatrix} 10 & 4 & -6 \\ 4 & -6 & 8 \\ -6 & 8 & 14 \end{bmatrix} \text{ MPa}$$



MATLAB Solution

```
clear all
sigma=[10 4 -6
       4 -6  8
      -6  8 14];
[dirs,stresses]=eig(sigma);
% diag(A) extracts the elements of the
% [A] matrix along the diagonal
principalStressList=diag(stresses)
principalDirs=dirs
```

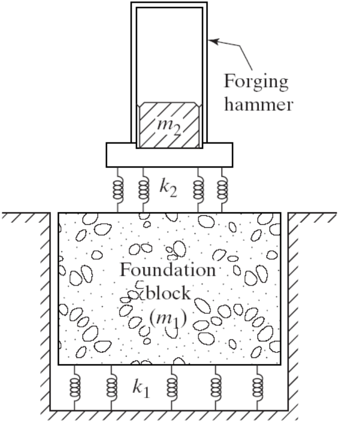
MATLAB Solution

```
>> rao_p431
principalStressList =
  -10.4828
    9.3181
   19.1647
principalDirs =
   -0.2792    0.8343   -0.4754
    0.8905    0.4102    0.1970
   -0.3594    0.3683    0.8574
```

Eigenvalue Solutions in MATLAB: General Problems

Solve the forging hammer problem for the following values:

- $m_1 = 20000$ kg
- $m_2 = 5000$ kg
- $k_1 = 1 \times 10^7$ N/m
- $k_2 = 5 \times 10^6$ N/m



Solving Eigenvalue Problems in MATLAB

Solving this eigenvalue problem will yield 2 eigenvalues equal to the square of the system's natural frequencies, and 2 corresponding x vector values that show the relative displacements of the m_1 and m_2 masses at those frequencies.

MATLAB Solution (Part 1)

```
clear all;
% Define spring constants and masses
% for hammer and foundation block
k1=1e7;
k2=5e6;
m1=20000;
m2=5000;

% Define system stiffness matrix
K=[k1+k2 -k2
    -k2  k2];
% Define system mass matrix
M=[m1 0
   0 m2];
```

MATLAB Solution (Part 2)

```
% Solve general eigenvalue problem
[X,Omega2]=eig(K,M);
% diag(A) extracts the elements of the
% [A] matrix along the diagonal
Omega=diag(sqrt(Omega2));
% Scale column 1 of the [X] matrix by
% the row 1, column 1 X value
X(:,1)=X(:,1)/X(1,1);
% Scale column 2 of the [X] matrix by
% the row 1, column 2 X value
X(:,2)=X(:,2)/X(1,2);

Omega
X
```

MATLAB Solution (Results)

```
>> rao_ex42
Omega =
   18.9634
   37.2879
X =
    1.0000    1.0000
    1.5616   -2.5616
```