

構造振動論 レポート

工学部航空宇宙工学科三年 野本陽平 (03-180332)

2019 年 2 月 5 日

はじめに

数値構造解析のときは、提出が遅くなったにも関わらず提出を受け入れていただき本当にありがとうございました。それを補うことができるののようなものだとは思っていませんが、今回は早めに提出させていただきます。

問題 1

下図に示すような断面が変化するはり (密度 ρ , ヤング率 E) の両端自由条件における曲げ振動解析 (紙面の面内曲げ振動のみを考える) をこころみる. 一様断面はりの固有振動モードを用い, 断面変化を有するはりの固有振動数および固有モード (図示すること) を 3 次モードまで求め, 高さ $2h$ を有する一様断面はりの場合と比較せよ. ただし, 各断面は幅一定の長方形形状をしており, 必要であれば $\rho = 2.7\text{g/cm}^3, E = 70\text{GPa}, h = 2\text{mm}, l = 1000\text{mm}$ とせよ.

諸物理量の定義

Solution. 形状が長方形である断面の断面二次モーメントは

$$I(x) = \frac{bh^3}{12} \left(1 + \frac{2x}{l} \right)$$

よって, 曲げ剛性 EI と線密度 μ は

$$EI = \frac{EBh^3}{12} \left(1 + \frac{2x}{l} \right)^3$$
$$\mu = \rho bh \left(1 + \frac{2x}{l} \right)$$

一方で, 高さ $2h$ を有する一様断面はりの場合の曲げ剛性 EI_{2h} と線密度 μ_{2h} は,

$$EI = \frac{Eb h^3}{12} \times 2^3 = \frac{2}{3} Eb h^3$$
$$\mu = 2\rho b h$$

である. 問題文の順序とは逆行することになるが, 以下では初めに一様断面はりの固有振動を検討し, 次に断面が変化するはりについて考えるものとする.

一様断面はり

運動方程式

$$\frac{\partial^2}{\partial x^2} \left(EI \frac{\partial^2 w}{\partial x^2} \right) + \mu \frac{\partial^2 w}{\partial t^2} - \frac{\partial}{\partial x} \left(\rho I \frac{\partial^3 w}{\partial t^2 \partial x} \right) = q_x + \frac{\partial m_x}{\partial x}$$

に対して、外力なし ($q_x = 0, m_x = 0$), 一様断面 ($EI, \rho I$ は x の関数ではない), $\rho I \ll EI, \mu$ の仮定を代入すると以下のような単純な形にできる.

$$EI \frac{\partial^4 w}{\partial x^4} + \mu \frac{\partial^2 w}{\partial t^2} = 0$$

理論解を $w(x, t) = u(x)v(t)$ なる変数分離でおき, 近似された EOM に代入すると,

$$EI u(t) \frac{d^4 u(x)}{dx^4} + \mu u(x) \frac{d^2 v(t)}{dt^2} = 0 \quad \Rightarrow \quad \frac{d^2 v}{dt^2} = -\frac{EI \frac{d^4 u}{dx^4}}{\mu u} \equiv -\omega^2$$

$$\therefore \frac{d^2 v}{dt^2} + \omega^2 v = 0, \quad \frac{d^4 u}{dx^4} - \left(\frac{\lambda}{l}\right)^4 u = 0$$

ただし解析的に計算しやすくするために $\lambda^4 \equiv \frac{\mu \omega^2 l^4}{EI}$ とおいている. それぞれを解くと,

$$u(x) = A \cos \omega t + B \sin \omega t$$

$$v(t) = C_1 \cosh \frac{\lambda x}{l} + C_2 \cos \frac{\lambda x}{l} + C_3 \sinh \frac{\lambda x}{l} + C_4 \sin \frac{\lambda x}{l}$$

である. 両端が自由端であるから, 剪断力及び曲げモーメントから境界条件 $x = 0, l: \frac{d^2 u}{dx^2} = \frac{d^3 u}{dx^3} = 0$ を得る. これを代入・整理して行列形式にすると,

$$\begin{bmatrix} \left(\frac{\lambda}{l}\right)^2 & -\left(\frac{\lambda}{l}\right)^2 & 0 & 0 \\ 0 & 0 & \left(\frac{\lambda}{l}\right)^2 & -\left(\frac{\lambda}{l}\right)^2 \\ \left(\frac{\lambda}{l}\right)^2 \cosh \lambda & -\left(\frac{\lambda}{l}\right)^2 \cos \lambda & \left(\frac{\lambda}{l}\right)^2 \sinh \lambda & -\left(\frac{\lambda}{l}\right)^2 \sin \lambda \\ \left(\frac{\lambda}{l}\right)^3 \sinh \lambda & \left(\frac{\lambda}{l}\right)^3 \sin \lambda & \left(\frac{\lambda}{l}\right)^3 \cosh \lambda & -\left(\frac{\lambda}{l}\right)^3 \cos \lambda \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

上の行列が非自明解を持つのは行列の行列式が 0 になる時であり, 上の行列を A とおくと

$$\det A = 0 \Leftrightarrow \cosh \lambda \cos \lambda - 1 = 0$$

これらの条件を加味すると, $\begin{bmatrix} C_1 & C_2 & C_3 & C_4 \end{bmatrix}^T$ は以下のように表される.

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -\frac{\sinh \lambda + \sin \lambda}{\cosh \lambda - \cos \lambda} \\ -\frac{\sinh \lambda + \sin \lambda}{\cosh \lambda - \cos \lambda} \end{bmatrix} C_1$$

この C_1 は振幅を表すファクターなので, 固有振動の固有振動数を考える際には適当に決めて良い. 故にここでは 1 とする. よって $u(x)$ は

$$u_r(x) = \cosh \frac{\lambda_r x}{l} + \cos \frac{\lambda_r x}{l} - \beta \left(\sinh \frac{\lambda_r x}{l} + \sin \frac{\lambda_r x}{l} \right)$$

となる. ただし λ_r は $\cosh \lambda \cos \lambda - 1 = 0$ の解を小さい順 (=低次の振動モードを示す順) に並べたものを $\lambda_1, \lambda_2, \dots$ と定義したものであり, 各 λ_r は [2] より数値計算で求めるものとする. なお, これらの値はシステム学実験で扱った Reference[1] の情報とも概ね一致しているから正しいものと判断できる; 今回は wolfram で求めた. wolfram については Reference[2] を参照.

$$\begin{cases} \lambda_1 \approx 4.7300407448627 \\ \lambda_2 \approx 7.8532046240958 \\ \lambda_3 \approx 10.995607838002 \\ \lambda_4 \approx 14.137165491258 \end{cases}$$

一方, $\omega_r = (\frac{\lambda}{l})^2 \sqrt{\frac{EI}{\mu}} = (\frac{\lambda}{l})^2 \sqrt{\frac{Eh^2}{3\rho}}$ であるから, 固有角振動数 ω_r は

$$\begin{cases} \omega_1 \approx 131.54255401562 & [rad/s] \\ \omega_2 \approx 362.60211546633 & [rad/s] \\ \omega_3 \approx 710.84512705380 & [rad/s] \\ \omega_4 \approx 1175.0631042471 & [rad/s] \end{cases}$$

テーパー梁

Rayleigh-Ritz 法を用いて解く. はりの振動波形 $W(x)$ は変異境界条件を満たす関数 $\phi_i(x)$ を用いて

$$W^*(x) = \sum_{i=1}^4 C_i \phi_i(x)$$

によって表すことができる. ただし,

$$\phi_i = \cosh \frac{\lambda_i x}{l} + \cos \frac{\lambda_i x}{l} - \beta \left(\sinh \frac{\lambda_i x}{l} + \sin \frac{\lambda_i x}{l} \right)$$

とする. 質量マトリックス M^* , 剛性マトリックス K^*

$$M^* = \int_0^l \mu \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 \end{bmatrix} dx = \int_0^l \rho h \left(1 + \frac{2x}{l} \right) \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 \end{bmatrix} dx$$

$$K^* = \int_0^l EI \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 \end{bmatrix} dx = \int_0^l \frac{Eh^3}{12} \left(1 + \frac{2x}{l} \right)^3 \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \end{bmatrix} \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 \end{bmatrix} dx$$

より, M^{*-1} を左からかける変形を用いて,

$$\begin{bmatrix} K^* - \omega^2 M^* \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = 0 \quad \Leftrightarrow \quad \begin{bmatrix} M^{*-1} K^* - \omega^2 I \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = 0$$

これが非自明解を持つ条件は,

$$\det(M^{*-1} K^* - \omega^2 I) = 0$$

である. したがって ω は $M^{*-1} K^*$ の固有値として, 各モードに対応する $\begin{bmatrix} C_1 & C_2 & C_3 & C_4 \end{bmatrix}^T$ が $M^{*-1} K^*$ の固有ベクトルとして求めることができる. 今回これを台形近似による数値計算で求めると, モード 1 からモード 4 までの比較グラフとテーブルは図 1, 2, 3, 4, 5, 6 と表 1 のようになった. なお, テーパー梁のコードは Appendix の `trapezoid.py` を参照.

質量マトリックスの要素が厚みに比例するのに対し, 剛性マトリックスの要素は厚みの 3 乗に比例するためテーパー梁は全体として剛になる (=固有角振動数が大きくなる) ことが予想されるが, 今回はそのような結果が得られていない. 果たしてこの結果は妥当なのだろうか. x 方向の分割数 N を大きくしてもいいのだが, それでは芸がなく python を使っている今回は計算時間もかなり長くなってしまったため, シンプソン法を用いる. シンプソン法に関する理論は Reference[3] を参照. モード 1 からモード 4 までの

比較グラフとテーブルは図 7, 8, 9, 10, 11, 12 と表 2 のようになった. これより確かに全ての固有角振動数において一様梁の理論解を上回っている. 具体的に用いたコードについては Appendix の Simpson.py を参照. 冗長になってしまうためコードは一部のみの抜粋としたが, この関数を trapezoid.py の同じ名前の関数と入れ替えれば実行できる. ■

表 1 テーパー梁の固有角振動数と固有ベクトル (台形近似)

	モード 1	モード 2	モード 3	モード 4
$\omega[\text{rad/s}]$	125.77	350.61	695.62	1303.06
C_1	-0.9962	-0.1570	0.1648	0.0024
C_2	0.0819	0.9856	-0.2809	-0.0353
C_3	0.0073	0.0615	0.9530	0.3054
C_4	0.0284	-0.0084	0.1123	-0.9516

表 2 テーパー梁の固有角振動数と固有ベクトル (シンプソン近似)

	モード 1	モード 2	モード 3	モード 4
$\omega[\text{rad/s}]$	154.04	429.40	851.95	1595.92
C_1	-0.9962	-0.1570	0.1648	0.0024
C_2	0.0819	0.9856	-0.2809	-0.0353
C_3	0.0073	0.0615	0.9530	0.3054
C_4	0.0284	-0.0084	0.1123	-0.9516

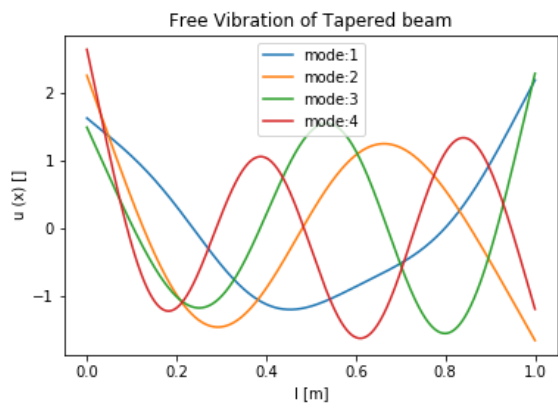


图 1

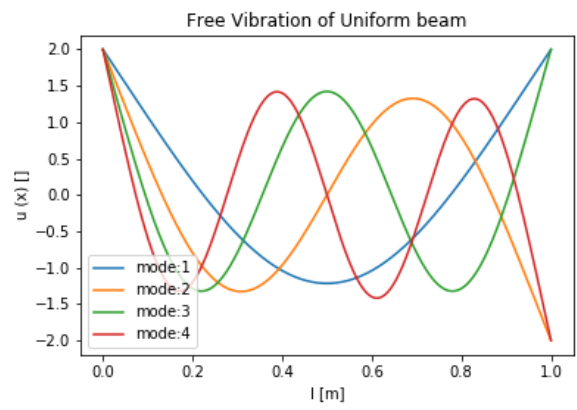


图 2

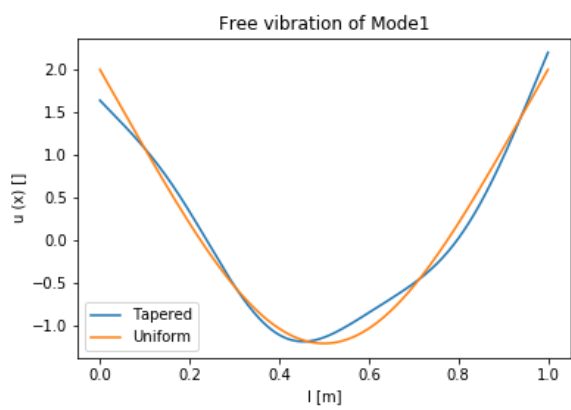


图 3

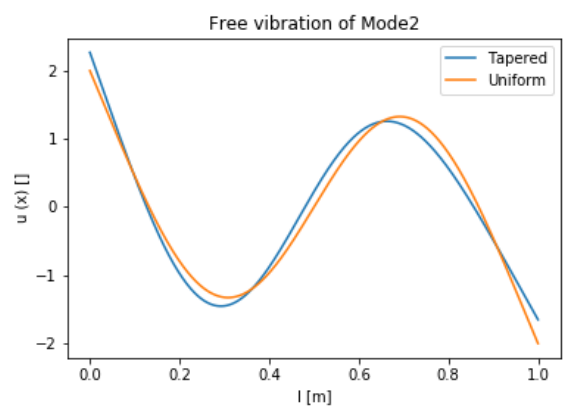


图 4

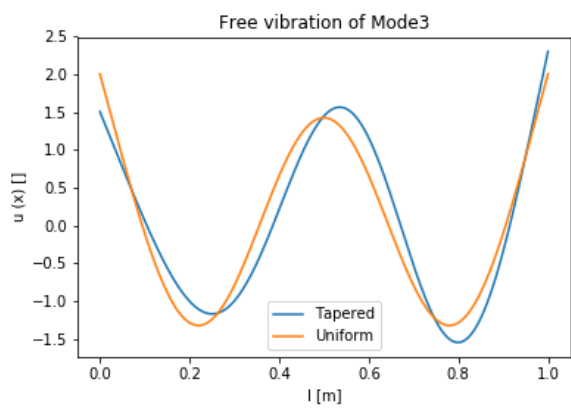


图 5

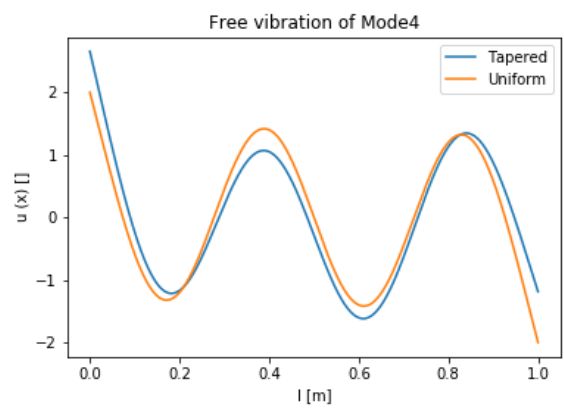


图 6

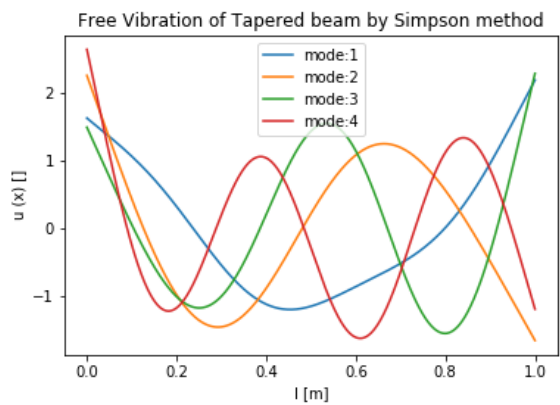


图 7

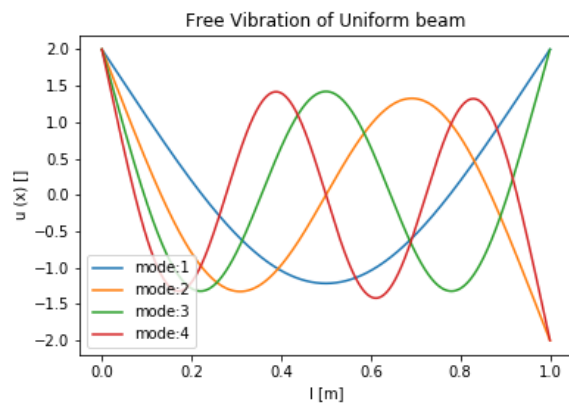


图 8

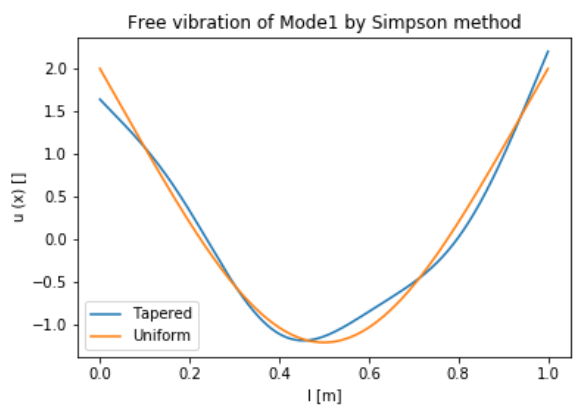


图 9

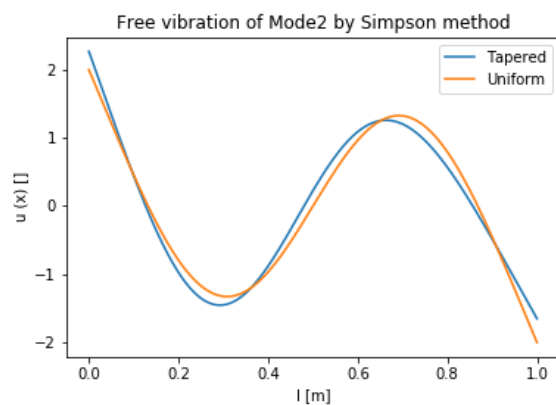


图 10

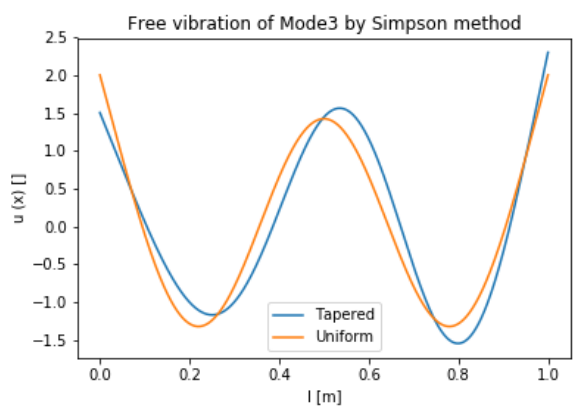


图 11

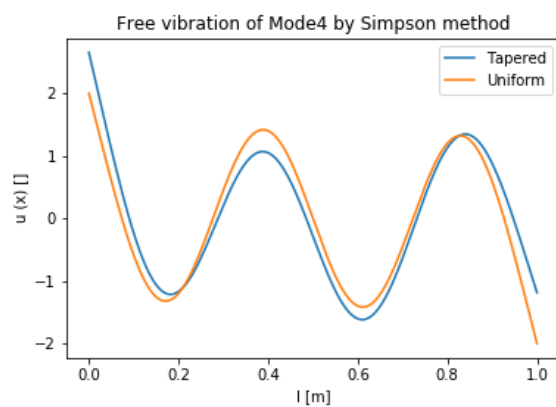


图 12

問題 2

下図に示すように、質量 m の剛体が速さ V_0 で棒 (密度 ρ , ヤング率 E , 断面積 A) に衝突したとする. このとき、以下の問いに答えよ. 尚、棒の左端を原点とする x 座標を図のように定める.

1. 棒を x の正方向に伝える縦波 $u = f(\xi) = f(x - at)$ を考える (a は位相速度). 棒に発生する応力と速度の関係を ρ, a を用いて表せ.

Solution. $u = f(\xi)$ を x と t で偏微分する.

$$\begin{aligned}\frac{\partial u}{\partial x} &= \frac{\partial \xi}{\partial x} \frac{\partial f}{\partial \xi} = \frac{\partial(x - at)}{\partial x} \frac{\partial f}{\partial \xi} = \frac{\partial f}{\partial \xi} \\ \frac{\partial u}{\partial t} &= \frac{\partial \xi}{\partial t} \frac{\partial f}{\partial \xi} = \frac{\partial(x - at)}{\partial t} \frac{\partial f}{\partial \xi} = -a \frac{\partial f}{\partial \xi}\end{aligned}$$

棒に発生する応力を $\sigma(x, t)$ と表すと、

$$\sigma(x, t) = E \frac{\partial u}{\partial x} = E \frac{\partial f}{\partial \xi} = -\frac{E}{a} \frac{\partial u}{\partial t} = \underline{\underline{-\rho a \frac{\partial u}{\partial t}}}$$

但し最後の変形では位相速度 a についての式 $a^2 = \frac{E}{\rho}$ を用いた. ■

2. 剛体と棒が接触している際、剛体と左端面は同じ速度を持つと考えられる. このとき、棒左端 ($x = 0$) の応力 $\sigma_0(t)$ と剛体速度 $V(t)$ の関係を示せ. また、棒からの反力を考え、接触している間の剛体の運動方程式を示せ.

Solution. 剛体と棒が接触しているとき、剛体と左端面は同じ速度を持つ. 故に剛体速度 $V(t)$ を用いると

$$\sigma_0(t) = \sigma(0, t) = -\rho a \left. \frac{\partial u}{\partial t} \right|_{x=0} = -\rho a V(t)$$

応力は引張が正であるから、

$$m \frac{dV}{dt} = \sigma_0(t) A = \underline{\underline{-\rho a A V(t)}}$$

■

3. (2) の 2 式から、棒左端の応力 $\sigma_0(t)$ の満たすべき微分方程式を求め、 $\sigma_0(t)$ を求めよ. ただし、衝突時を $t = 0$ とし、そのとき、棒左端の速度は V_0 と等しいと考えてよい.

Solution. $V(t)$ の式を剛体の運動方程式に代入すると、

$$\frac{d}{dt} \sigma_0(t) = -\frac{\rho a A}{m} \sigma_0(t)$$

これを解くことで、不定積分込みの $\sigma_0(t)$ を得る.

$$\sigma_0(t) = C \exp\left(-\frac{\rho a A}{m} t\right)$$

初期条件 $V(0) = V_0 \Rightarrow \sigma_0(0) = C = -\rho a V_0$ を代入して完全な $\sigma_0(t)$ を得る.

$$\sigma_0(t) = -\rho a V_0 \exp\left(-\frac{\rho a A}{m}t\right)$$

■

4. 縦波の運動方程式を棒内部の応力を用いて表し, x の正方向に伝わる応力波 $\sigma(x, t) = g(x - at)$ はこの運動方程式の解であることを示せ.

Proof. 今外力なしで一様断面の物体に対する縦波の運動方程式を考える. これを整理すると,

$$\begin{aligned}\frac{\partial}{\partial t}\left(EA\frac{\partial u}{\partial x}\right) + P_x &= \rho A \frac{\partial^2 u}{\partial t^2} \\ \Rightarrow EA \frac{\partial^2 u}{\partial x^2} &= \rho A \frac{\partial^2 u}{\partial t^2} \\ \Rightarrow \frac{\partial^2 u}{\partial t^2} &= a^2 \frac{\partial^2 u}{\partial x^2}\end{aligned}$$

また, E が定数であることから,

$$\begin{aligned}\sigma(x, t) &= E \frac{\partial u}{\partial x} \Rightarrow \frac{\partial \sigma}{\partial x} = E \frac{\partial^2 u}{\partial x^2} \\ \therefore \frac{\partial \sigma}{\partial x} &= \rho \frac{\partial^2 u}{\partial t^2}\end{aligned}$$

両辺を x で偏微分し, 偏微分を交代させると棒内部の応力による縦波の運動方程式を得る.

$$\begin{aligned}\frac{\partial^2 \sigma}{\partial x^2} &= \rho \frac{\partial}{\partial x} \left(\frac{\partial^2 u}{\partial t^2} \right) = \rho \frac{\partial^2}{\partial t^2} \left(\frac{\partial u}{\partial x} \right) = \frac{\rho}{E} \frac{\partial^2 \sigma}{\partial t^2} \\ \therefore \frac{\partial^2 \sigma}{\partial t^2} &= \frac{E}{\rho} \frac{\partial^2 \sigma}{\partial x^2} = a^2 \frac{\partial^2 \sigma}{\partial x^2}\end{aligned}$$

これに対し $\sigma(x, t) = g(x - at)$ を代入し, 運動方程式の解であることを示すことができる.

$$\begin{aligned}\frac{\partial^2 \sigma}{\partial t^2} &= \frac{\partial(x - at)}{\partial t} \frac{\partial}{\partial(x - at)} \left(\frac{\partial \sigma}{\partial t} \right) = -a \frac{\partial}{\partial(x - at)} \left(\frac{\partial(x - at)}{\partial t} \frac{\partial g}{\partial(x - at)} \right) = a^2 \frac{\partial^2 g}{\partial(x - at)^2} \\ \frac{\partial^2 \sigma}{\partial x^2} &= \frac{\partial(x - at)}{\partial x} \frac{\partial}{\partial(x - at)} \left(\frac{\partial \sigma}{\partial x} \right) = \frac{\partial}{\partial(x - at)} \left(\frac{\partial(x - at)}{\partial x} \frac{\partial g}{\partial(x - at)} \right) = \frac{\partial^2 g}{\partial(x - at)^2}\end{aligned}$$

以上より, 確かに運動方程式の解として正しい. □

5. 棒内部へはどのような縦波 (応力波) が伝搬するか. 衝突から t 秒後の棒内部の応力状態を図示せよ.

Solution. 3, 4 で得た結論によると,

$$\sigma_0(t) = -\rho a V_0 \exp\left(-\frac{\rho a A}{m}t\right) = g(-at)$$

g は $x - at$ による関数であることを踏まえると, 伝播を表す g と衝突を表す σ_0 の間に以下のような関係があると考えて矛盾はない.

$$g(x - at) = -\rho a V_0 \exp\left(\frac{\rho A}{m}(x - at)\right)$$

これを図示すると以下のようなになる. ただし, $at < x$ には衝撃が伝播していないため, 応力が存在しない.

■

Appendix

Listing 1 trapezoid.py

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Sun Jan 27 05:24:06 2019
5
6  @author: yohei
7  """
8
9  import numpy as np
10 import matplotlib.pyplot as plt
11
12 E = 70 * (10 ** 9) #youngs_modulus[Pa]
13 h = 0.002 #thickness[m]
14 rho = 2700 #desity[kg/m^3]
15 l = 1 #length[m]
16 lambda_r = np.array([4.7300407448627,
17                       7.8532046240958,
18                       10.995607838002,
19                       14.137165491258])
20 beta = np.zeros(4)
21 for i in range(len(beta)):
22     beta[i] += (np.cosh(lambda_r[i]) - np.cos(lambda_r[i])) / (np.sinh(lambda_r[i]) -
23                                                                np.sin(lambda_r[i]))
24
25 class BEF: #Both Edge Free
26     def __init__(self):
27         self.N = 1000 #division number
28         self.x_ = np.linspace(0, 1, self.N)
29
30     def phi(self, x):
31         phi = np.array([np.cosh(lambda_r[i]*x/l) + np.cos(lambda_r[i]*x/l) - beta
32                        [i]*(np.sinh(lambda_r[i]*x/l)+np.sin(lambda_r[i]*x/l)) for i in range
33                        (4)])
34         dif = np.array([(lambda_r[i]/l)**2 for i in range(4)])
35         ddphi = np.array([dif[i]*(np.cosh(lambda_r[i]*x/l) - np.cos(lambda_r[i]*x
36                               /l) - beta[i]*(np.sinh(lambda_r[i]*x/l)-np.sin(lambda_r[i]*x/l))) for
37                               i in range(4)])
38         return phi, ddphi
39
40     def f(self, x):
```

```

36     EI = E * h**3 * (1 + 2*x/l)**3 / 12
37     phi_for_EI = [[],[],[],[]]
38     ddp_phi = self.phi(x)[1]
39     for i in range(4):
40         for j in range(4):
41             phi_for_EI[i].append(ddp_phi[i]*ddp_phi[j]*EI)
42     return phi_for_EI
43
44 def g(self, x):
45     mu = rho*h*(1+2*x/l)
46     phi_for_mu = [[],[],[],[]]
47     phi = self.phi(x)[0]
48     for i in range(4):
49         for j in range(4):
50             phi_for_mu[i].append(phi[i]*phi[j]*mu)
51     return phi_for_mu
52
53 def K(self):
54     K = np.zeros((4,4))
55     for i in range(4):
56         for j in range(4):
57             for k in range(len(self.x_)-1):
58                 y=(self.f(self.x_[k])[i][j]+self.f(self.x_[k+1])[i][j])/2
59                 K[i][j]+=(self.x_[k+1]-self.x_[k])*y
60     return K
61
62 def M(self):
63     M = np.zeros((4,4))
64     for i in range(0,4):
65         for j in range(0,4):
66             for k in range(len(self.x_)-1):
67                 y=(self.g(self.x_[k])[i][j]+self.g(self.x_[k+1])[i][j])/2
68                 M[i][j]+=(self.x_[k+1]-self.x_[k])*y
69     return M
70
71 def eigen(self):
72     M_inv = np.linalg.inv(self.M())
73     dots = np.dot(M_inv, self.K())
74     eigenvalue, eigenvector = np.linalg.eig(dots.T)
75     return np.sqrt(eigenvalue), eigenvector
76
77 def discretization(self):
78     ector = self.eigen()[1]
79     y = np.zeros((4,len(self.x_)))
80     for k in range(4):

```

```

81         for i in range(len(self.x_)):
82             temp = 0
83             n = self.x_[i]
84             for j in range(4):
85                 temp += ector[k][j]*self.phi(n)[0][j]
86             y[k][i] = temp
87     return y
88
89     def describe_tapered(self):
90         y = self.discretization()
91         plt.title("Free Vibration of Tapered beam")
92         plt.plot(self.x_, -y[3], label="mode:1")
93         plt.plot(self.x_, -y[0], label="mode:2")
94         plt.plot(self.x_, y[1], label="mode:3")
95         plt.plot(self.x_, y[2], label="mode:4")
96         plt.legend()
97         plt.xlabel("l [m]")
98         plt.ylabel("u (x) []")
99         plt.savefig('fig1.png')
100
101     def describe_uniform(self):
102         y = self.phi(self.x_)
103         plt.title("Free Vibration of Uniform beam")
104         for i in range(4):
105             plt.plot(self.x_, y[0][i], label="mode:"+ str(i+1))
106         plt.legend()
107         plt.xlabel("l [m]")
108         plt.ylabel("u (x) []")
109         plt.savefig('fig2.png')
110
111     def describe_model1(self):
112         y1 = self.discretization()
113         y2 = self.phi(self.x_)
114         plt.title("Free vibration of Model1")
115         plt.plot(self.x_, -y1[3], label="Tapered")
116         plt.plot(self.x_, y2[0][0], label="Uniform")
117         plt.legend()
118         plt.xlabel("l [m]")
119         plt.ylabel("u (x) []")
120         plt.savefig('fig3.png')
121
122     def describe_mode2(self):
123         y1 = self.discretization()
124         y2 = self.phi(self.x_)
125         plt.title("Free vibration of Mode2")

```

```

126     plt.plot(self.x_, -y1[0], label="Tapered")
127     plt.plot(self.x_, y2[0][1], label="Uniform")
128     plt.legend()
129     plt.xlabel("l [m]")
130     plt.ylabel("u (x) []")
131     plt.savefig('fig4.png')
132
133     def describe_mode3(self):
134         y1 = self.discretization()
135         y2 = self.phi(self.x_)
136         plt.title("Free vibration of Mode3")
137         plt.plot(self.x_, y1[1], label="Tapered")
138         plt.plot(self.x_, y2[0][2], label="Uniform")
139         plt.legend()
140         plt.xlabel("l [m]")
141         plt.ylabel("u (x) []")
142         plt.savefig('fig5.png')
143
144     def describe_mode4(self):
145         y1 = self.discretization()
146         y2 = self.phi(self.x_)
147         plt.title("Free vibration of Mode4")
148         plt.plot(self.x_, y1[2], label="Tapered")
149         plt.plot(self.x_, y2[0][3], label="Uniform")
150         plt.legend()
151         plt.xlabel("l [m]")
152         plt.ylabel("u (x) []")
153         plt.savefig('fig6.png')
154
155     def free_vibration():
156         root = np.sqrt((E*(h**2))/(3*rho))
157         temp = []
158         for i in range(len(lambda_r)):
159             temp.append(root*(lambda_r[i]**2)/l**2)
160         return temp
161
162     if __name__ == '__main__':
163         bef = BEF()
164         #uni = uniform()
165         print("Eigenvalue and Eigenvector", bef.eigen())
166         print("equivalent to Eigenvalue of free vibration", free_vibration())
167         #bef.describe_tapered()
168         #bef.describe_uniform()
169         #bef.describe_mode1()
170         #bef.describe_mode2()

```

```
171     #bef.describe_mode3()
172     bef.describe_mode4()
```

Listing 2 Simpson.py

```
1     def discretization(self):
2         ector = self.eigen()[1]
3         y = np.zeros((4,len(self.x_)))
4         for k in range(4):
5             for i in range(len(self.x_)):
6                 temp = 0
7                 n = self.x_[i]
8                 for j in range(4):
9                     temp += ector[k][j]*self.phi(n)[0][j]
10                y[k][i] = temp
11        return y
```

Reference

- [1] "Vibration of Redtangular Plates by the Ritz Method" by DANA YOUNG
- [2] <https://www.wolframalpha.com>
- [3] <https://qiita.com/popondeli/items/fbc602b80223227f3d0e>