構造実験・解析演習

副読本

解析演習の目的と概要

構造力学では基礎理論を学んできた．構造実験では基礎的な試験体や計測装置を用いた基礎実験を実施する．実験では未知なる特性を持つ構造・材料に対する試験を行い，その変形特性や破壊性状を知ることができる．その特性が明らかになってくると，実験データ等を元に解析モデルが作成できる．これによってはじめて構造解析というものを実施することできる．構造解析はあらかじめ材料や構造部材の特性が設定されていれば，実験を行うことなく，様々な境界条件や構造形式に対する構造物の応答を計算することができる．実際の構造物を構築する際には，このような構造解析を駆使することにより，要求される性能を満足する構造物を合理的に設計・建設することができる．
本解析演習では，マトリックス構造解析法を基礎とする基本的な構造解析プログラムを作成することを目的とする．連立一次方程式の解法，剛性行列の作成法を学び，荷重が作用した際の構造物の変形状態を求めるプログラムを作成する．このプログラムを用いて構造実験で実施したはりの試験を再現することができる．実験および解析結果を比較することで，実験・解析の両輪を持つ構造力学の理解を深めることを期待している．

Chapter 1

骨組み構造解析の基礎理論

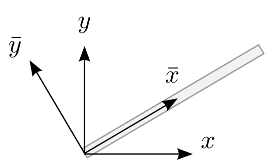
1.1　要素剛性マトリクスの作成

力と変位の関係を求める方法はいくつかある．

1. 力のつりあい式を直接解く
2. 仮想仕事の原理を用いる
3. カステリィアーノの定理を用いる

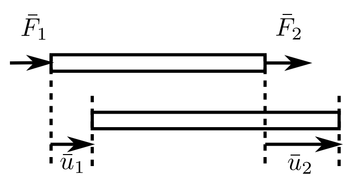
ここでは 1. の「力のつりあい式を直接解く」方法について説明する．.

最初に，簡単のため，部材上に局所座標系を考える $\bar{x}$ 軸ははりの方向に，$\bar{y}$ 軸ははりに直交する方向に取る．また，変形は「伸び」と「曲げ」に分けられるものとし，それぞれ独立に式を立てる．



1.1.1　伸びについて

まず，伸びについて考える．　両端における軸方向の変位を $\bar{u}_1, \bar{u}_2$，軸方向に作用する力を $\bar{F}_1, \bar{F}_2$ とする．



部材中で（伸び）ひずみが一様とする．

$$\bar{u}' = a_1 \tag{1.1}$$

$\bar{x}$ について積分する[1]．

$$\bar{u} = a_0 + a_1 \bar{x} \tag{1.2}$$

[1]節点における変位の情報が 2 つ (u_1, u_2) あるので，変位を 2 つの係数 (a_0, a_1) を持つ 1 次式で表すことは自然な発想であろう．

両端の変位から次式を得る．

$$\bar{u}(\bar{x} = L) = a_0 + a_1 L = \bar{u}_2 \quad \bar{u}(\bar{x} = 0) = a_0 = \bar{u}_1 \tag{1.3}$$

(1.4)

行列を用いて表現すると次のようになる．

$$\begin{bmatrix} 1 & 0 \\ 1 & L \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{Bmatrix} \tag{1.5}$$

また，部材両端における応力ひずみ関係式 $E\varepsilon = \sigma = \frac{P}{A}$ より

$$EA\bar{u}'(\bar{x} = 0) = EAa_1 = -\bar{F}_1, \tag{1.6}$$

$$EA\bar{u}'(\bar{x} = L) = EAa_1 = \bar{F}_2 \tag{1.7}$$

を得る[2]．　この式も同様に行列で表現する．

$$EA \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} \bar{F}_1 \\ \bar{F}_2 \end{Bmatrix} \tag{1.8}$$

式 (1.5) から a_0, a_1 を求める．

$$\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \frac{1}{L} \begin{bmatrix} L & 0 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{Bmatrix} \tag{1.9}$$

式 (1.8)，(1.9) をまとめると力と変位の関係式を得る．

$$\begin{Bmatrix} \bar{F}_1 \\ \bar{F}_2 \end{Bmatrix} = \frac{EA}{L} \begin{bmatrix} 0 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} L & 0 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{Bmatrix} \tag{1.10}$$

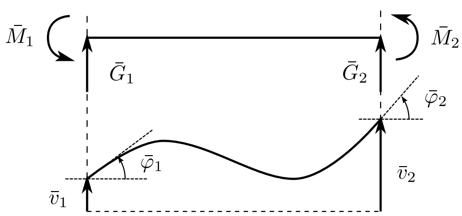
$$= \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{Bmatrix} \tag{1.11}$$

1.1.2　曲げについて

曲げでは，力と変位の関係を求める[3]．　両端のたわみを $\bar{v}_1, \bar{v}_2$，たわみ角を $\bar{\varphi}_1, \bar{\varphi}_2$ とする．また，両端に作用する鉛直上向きの力を $\bar{G}_1, \bar{G}_2$，曲げモーメントを $\bar{M}_1, \bar{M}_2$ とする．

[2]伸びを正としている．F_1 は圧縮に作用するためマイナスの符号がつく．

[3]ここで，力はせん断力とモーメントからなり，変位はたわみとたわみ角からなる．



はりのたわみの微分方程式

$$EI\bar{v}'''' = 0 \tag{1.12}$$

を $\bar{x}$ について 4 回積分する[4].

$$\bar{v} = c_0 + c_1\bar{x} + c_2\bar{x}^2 + c_3\bar{x}^3 \tag{1.13}$$

部材両端のたわみ，たわみ角より，次式を得る．

$$\bar{v}(\bar{x}=0) = c_0 = \bar{v}_1 \tag{1.14}$$
$$\bar{v}'(\bar{x}=0) = c_1 = \bar{\varphi}_1 \tag{1.15}$$
$$\bar{v}(\bar{x}=L) = c_0 + c_1L + c_2L^2 + c_3L^3 = \bar{v}_2 \tag{1.16}$$
$$\bar{v}'(\bar{x}=L) = c_1 + 2c_2L + 3c_3L^2 = \bar{\varphi}_2 \tag{1.17}$$

また，部材両端におけるせん断力，曲げモーメントより次式を得る [5]

$$EI\bar{v}'''(\bar{x}=0) = 6EIc_3 = \bar{G}_1 \tag{1.18}$$
$$EI\bar{v}''(\bar{x}=0) = 2EIc_2 = -\bar{M}_1 \tag{1.19}$$
$$EI\bar{v}'''(\bar{x}=L) = 6EIc_3 = -\bar{G}_2 \tag{1.20}$$
$$EI\bar{v}''(\bar{x}=L) = 2EIc_2 + 6EIc_3L = \bar{M}_2 \tag{1.21}$$

これらを行列表現する．

$$\begin{Bmatrix} \bar{v}_1 \\ \bar{\varphi}_1 \\ \bar{v}_2 \\ \bar{\varphi}_2 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & L & L^2 & L^3 \\ 0 & 1 & 2L & 3L^2 \end{bmatrix} \begin{Bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{Bmatrix} \tag{1.22}$$

$$\begin{Bmatrix} \bar{G}_1 \\ \bar{M}_1 \\ \bar{G}_2 \\ \bar{M}_2 \end{Bmatrix} = EI \begin{bmatrix} 0 & 0 & 0 & 6 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -6 \\ 0 & 0 & 2 & 6L \end{bmatrix} \begin{Bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{Bmatrix} \tag{1.23}$$

式 (1.22) を c_0，c_1，c_2，c_3 について解く．

$$\begin{Bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{Bmatrix} = \frac{1}{L^4} \begin{bmatrix} L^4 & 0 & 0 & 0 \\ 0 & L^4 & 0 & 0 \\ -3L^2 & -2L^3 & 3L^2 & -L^3 \\ 2L & L^2 & -2L & L^2 \end{bmatrix} \begin{Bmatrix} \bar{v}_1 \\ \bar{\varphi}_1 \\ \bar{v}_2 \\ \bar{\varphi}_2 \end{Bmatrix} \tag{1.24}$$

さらに，式 (1.23)，(1.24) をまとめると，一般化力と一般化変位の関係が得られる．

$$\begin{Bmatrix} \bar{G}_1 \\ \bar{M}_1 \\ \bar{G}_2 \\ \bar{M}_2 \end{Bmatrix} = \frac{EI}{L^4} \begin{bmatrix} 12L & 6L^2 & -12L & 6L^2 \\ 6L^2 & 4L^3 & -6L^2 & 2L^3 \\ -12L & -6L^2 & 12L & -6L^2 \\ 6L^2 & 2L^3 & -6L^2 & 4L^3 \end{bmatrix} \begin{Bmatrix} \bar{v}_1 \\ \bar{\varphi}_1 \\ \bar{v}_2 \\ \bar{\varphi}_2 \end{Bmatrix} \tag{1.25}$$

[4]一般化変位の未知数が 4 つなので係数を 4 つもつ 3 次式となる．
[5]正負に注意すること．

結局，式 (1.11), (1.25) をまとめると次式を得る．

$$\begin{Bmatrix} \bar{F}_1 \\ \bar{G}_1 \\ \bar{M}_1 \\ \bar{F}_2 \\ \bar{G}_2 \\ \bar{M}_2 \end{Bmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ & & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ & & & \frac{EA}{L} & 0 & 0 \\ & sym. & & & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ & & & & & \frac{4EI}{L} \end{bmatrix} \begin{Bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ \bar{\varphi}_1 \\ \bar{u}_2 \\ \bar{v}_2 \\ \bar{\varphi}_2 \end{Bmatrix} \tag{1.26}$$

これを簡単に以下のように書くこととしよう．

$$\bar{\boldsymbol{F}} = \bar{\boldsymbol{K}}_e \bar{\boldsymbol{u}} \tag{1.27}$$

1.2 座標の回転

局所座標系で考えていたが，これを全体座標系に置き換える．局所座標系の場合は変数の上に $^-$ を付けていたが，全体座標系の場合は何も付けず，座標系を区別する．

いま，部材が全体座標系に対して θ 傾いているとすると，変位 (u, v) と力 (F, G) について以下の式が成り立つ．

$$\begin{Bmatrix} \bar{u} \\ \bar{v} \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} u \\ v \end{Bmatrix} \tag{1.28}$$

$$\begin{Bmatrix} \bar{F} \\ \bar{G} \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} F \\ G \end{Bmatrix} \tag{1.29}$$

たわみ角 $\bar{\varphi}$ および曲げモーメント M については座標系によらない．

$$\bar{\varphi} = \varphi \qquad \bar{M} = M \tag{1.30}$$

部材両端における式 (1.28), (1.29), (1.30) の関係をまとめると次のようになる．

$$\begin{Bmatrix} \bar{u}_1 \\ \bar{v}_1 \\ \bar{\varphi}_1 \\ \bar{u}_2 \\ \bar{v}_2 \\ \bar{\varphi}_2 \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \varphi_1 \\ u_2 \\ v_2 \\ \varphi_2 \end{Bmatrix} \tag{1.31}$$

$$\begin{Bmatrix} \bar{F}_1 \\ \bar{G}_1 \\ \bar{M}_1 \\ \bar{F}_2 \\ \bar{G}_2 \\ \bar{M}_2 \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} F_1 \\ G_1 \\ M_1 \\ F_2 \\ G_2 \\ M_2 \end{Bmatrix} \tag{1.32}$$

式 (1.31), (1.32) を次のように表現する．

$$\bar{\boldsymbol{u}} = \boldsymbol{R}\boldsymbol{u} \qquad \bar{\boldsymbol{F}} = \boldsymbol{R}\boldsymbol{F} \tag{1.33}$$

上式に式 (1.33) を代入すると以下の式を得る[6].

$$\boldsymbol{F} = \boldsymbol{R}^T \bar{\boldsymbol{K}}_e \boldsymbol{R}\, \boldsymbol{u} \tag{1.34}$$

式 (1.34) で,

$$\boldsymbol{K}_e = \boldsymbol{R}^T \bar{\boldsymbol{K}}_e \boldsymbol{R} \tag{1.35}$$

とおくと，式 (1.34) は次のように書ける．

$$\boldsymbol{F} = \boldsymbol{K}_e \boldsymbol{u} \tag{1.36}$$

1.3 全体剛性行列の組み立て

端点 i の変位，力を以下のように表記する．

$$\boldsymbol{u}_i = \begin{Bmatrix} u_i \\ v_i \\ \phi_i \end{Bmatrix}, \qquad \boldsymbol{F}_i = \begin{Bmatrix} F_i \\ G_i \\ M_i \end{Bmatrix} \tag{1.37}$$

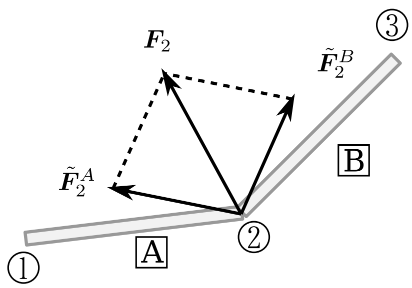
下図に示すように，部材 A と部材 B が連結している構造を考える．

$$\begin{bmatrix} \boldsymbol{K}_{11}^A & \boldsymbol{K}_{12}^A \\ \boldsymbol{K}_{21}^A & \boldsymbol{K}_{22}^A \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{Bmatrix} = \begin{Bmatrix} \tilde{\boldsymbol{F}}_1^A \\ \tilde{\boldsymbol{F}}_2^A \end{Bmatrix} \tag{1.38}$$

$$\begin{bmatrix} \boldsymbol{K}_{22}^B & \boldsymbol{K}_{23}^B \\ \boldsymbol{K}_{32}^B & \boldsymbol{K}_{33}^B \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \end{Bmatrix} = \begin{Bmatrix} \tilde{\boldsymbol{F}}_2^B \\ \tilde{\boldsymbol{F}}_3^B \end{Bmatrix} \tag{1.39}$$

$\tilde{\boldsymbol{F}}_2^A$ は部材 A に対して作用する節点 2 における力，$\tilde{\boldsymbol{F}}_2^B$ は部材 B に対して作用する節点 2 における力である． 節点 2 における力 $\boldsymbol{F}_2$ はこれらの和で表される．

$$\boldsymbol{F}_2 = \tilde{\boldsymbol{F}}_2^A + \tilde{\boldsymbol{F}}_2^B \tag{1.40}$$



式 (1.38), (1.39) を組み合わせると次式を得る．

$$\begin{bmatrix} \boldsymbol{K}_{11}^A & \boldsymbol{K}_{12}^A & 0 \\ \boldsymbol{K}_{21}^A & \boldsymbol{K}_{22}^A + \boldsymbol{K}_{22}^B & \boldsymbol{K}_{23}^B \\ 0 & \boldsymbol{K}_{32}^B & \boldsymbol{K}_{33}^B \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \\ \boldsymbol{u}_3 \end{Bmatrix} = \begin{Bmatrix} \tilde{\boldsymbol{F}}_1^A \\ \tilde{\boldsymbol{F}}_2^A + \tilde{\boldsymbol{F}}_2^B \\ \tilde{\boldsymbol{F}}_3^B \end{Bmatrix} \tag{1.41}$$

一般に，$\tilde{\boldsymbol{F}}_2^A$ と $\tilde{\boldsymbol{F}}_2^B$ の値はそれぞれ未知であるが，節点 2 における力 $\boldsymbol{F}_2 (= \tilde{\boldsymbol{F}}_2^A + \tilde{\boldsymbol{F}}_2^B)$ は既知である．

[6] $\boldsymbol{R}$ は直交行列であることに注意．すなわち，$\boldsymbol{R}^{-1} = \boldsymbol{R}^T$

1.4 境界条件の処理

変位や角度が固定されるとき，線形方程式 $\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F}$ において，その固定される変数に対応する解は 0 となる ($u_i = 0$) [7]． このとき，$u_i = 0$ を得るため，行列と右辺ベクトルを下記のように置き換える．この操作により，他の解に影響を与えることなく，$u_i = 0$ なる自明な解を得る．

$$\begin{cases} K_{ij} = 0 \quad (i \neq j) \\ K_{ii} = 1 \\ F_i = 0 \end{cases} \tag{1.42}$$

$$\begin{bmatrix} & & & 0 & & & \\ & \ddots & & \vdots & & & \\ & & K_{i-1,i-1} & 0 & K_{i-1,i+1} & & \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ & & K_{i+1,i-1} & 0 & K_{i+1,i+1} & & \\ & & & \vdots & & \ddots & \\ & & & 0 & & & \end{bmatrix} \begin{Bmatrix} \vdots \\ u_{i-1} \\ u_i \\ u_{i+1} \\ \vdots \end{Bmatrix} = \begin{Bmatrix} \vdots \\ F_{i-1} \\ 0 \\ F_{i+1} \\ \vdots \end{Bmatrix} \tag{1.43}$$

$u_i = 0$ に該当する行と列をマトリクスと右辺ベクトルから削除する方法もあるが，本演習では上記の簡単な方法を取る．

1.5 縁ひずみの計算

部材の縁のひずみ ε と応力 σ は下記の式で与えられる．

$$\varepsilon = \frac{\sigma}{E}, \qquad \sigma = \frac{M\bar{y}}{I} \tag{1.44}$$

h ははりの高さであり，縁において $\bar{y} = \frac{h}{2}$ となる．曲げモーメント M は式 (1.13) から計算する．

例えば，**部材中央**の曲げモーメントは，

$$M\left(\bar{x} = \frac{L}{2}\right) = EI\bar{v}''\left(\bar{x} = \frac{L}{2}\right) \tag{1.45}$$

$$= 2EIc_2 + 3EIc_3 L \tag{1.46}$$

となる． 係数 c_2, c_3 は部材ごとに異なる値であり，式 (1.24) から得られる．

$$c_2 = -3\frac{\bar{v}_1}{L^2} - 2\frac{\bar{\varphi}_1}{L} + 3\frac{\bar{v}_2}{L^2} - \frac{\bar{\varphi}_2}{L} \tag{1.47}$$

$$c_3 = 2\frac{\bar{v}_1}{L^3} + \frac{\bar{\varphi}_1}{L^2} - 2\frac{\bar{v}_2}{L^3} + \frac{\bar{\varphi}_2}{L^2} \tag{1.48}$$

たわみ $\bar{v}$ は局所座標系における変位なので，全体座標系で計算した変位 u, v から式 (1.33) を用いて変換する．

$$\bar{v}_1 = -\sin\theta\, u_1 + \cos\theta\, v_1 \tag{1.49}$$

$$\bar{v}_2 = -\sin\theta\, u_2 + \cos\theta\, v_2 \tag{1.50}$$

角度 φ は式 $(1.30)_1$ に示す通り，座標系によらないので変換する必要はない．

[7] 例えば，節点 k の x 方向，y 方向，回転が固定されるとき，それぞれ $u_{3k-2} = 0$，$u_{3k-1} = 0$，$u_{3k} = 0$ となる．

1.6 可視化のための出力

得られた結果を視覚的に理解できるように，可視化する．部材を滑らかに表現するために，部材を細かく分割する．分割された点を接続し，部材を表現する．　各部材の変形は，伸びについては式 (1.2)，曲げについては式 (1.13) で表される．ともに局所座標系での変位を用いているので，全体座標系で得られた変位を局所座標系での変位に変換する．全体座標系から局所座標系への変換は式 $(1.33)_1$ を用いる．

次に，ビームを分割し，その分割した点における局所座標系における変形量を計算する．そして，それらを全体座標系に戻す．局所座標系からへ全体座標系の変換は次式で表される．

$$\boldsymbol{u} = \boldsymbol{R}^T \bar{\boldsymbol{u}} \tag{1.51}$$

Chapter 2

骨組み構造解析のプログラム

2.1 データの入力

今回作成する骨組み構造解析のプログラムは以下のような流れになる．演習ではこのプログラムを上から順に完成させる．括弧内はサブルーチン名を表しており，これらのサブルーチンも作成する．

プログラムの流れ

変数の宣言
データの入力 (`inputd`)
do 要素
　　要素剛性行列の作成 (`elemstiff`)
　　全体剛性行列の組み立て (`globstiff`)
end do
境界条件の設定 (`setbc`)
線形連立方程式の求解 (`solvele`)
ひずみの計算 (`compstrain`)
可視化用データの作成 (`outputd`)

演習 1
構造のデータを外部ファイルから読み取むプログラム[a]と外部ファイルを[b]作成せよ．また，プログラムが作動するか確認せよ．

説明
???は各自で考えて適当な式を入力せよ．プログラムのコンパイルは付録 A を参照すること．

[a]作成するファイルの名前は適当に決めて良いが，拡張子は `.f90` とすること．
[b]拡張子は指定しない．下記のプログラムでは `kunoji.dat`としている．

```
program frame_analysis
  implicit none
  integer nnode,nelem
  integer,allocatable:: ine(:,:),ibc(:,:),mapping(:,:)
  real(8),allocatable:: pos(:,:),force(:,:),disp(:,:)
  real(8),allocatable:: hght(:),wdth(:),ym(:),EA(:),EI(:)
  real(8),allocatable:: s(:,:),u(:)
  real(8) se(6,6)
  integer ielem,inode,i,j
```

```
10
11 !!!=== INPUT DATA ===
12   open(100,file='kunoji.dat')
13   read(100,*) nnode,nelem
14   allocate(pos(2,nnode),force(3,nnode),disp(3,nnode),mapping(3,nnode))
15   allocate(hght(nelem),wdth(nelem),ym(nelem),EA(nelem),EI(nelem))
16   allocate(s(3*nnode,3*nnode),u(3*nnode))
17   allocate(ine(2,nelem),ibc(3,nnode))
18
19   do inode=1,nnode
20      read(100,*) (pos(i,inode),i=1,2)
21   end do
22
23   j=0
24   do inode=1,nnode
25      do i=1,3
26         ???
27         mapping(i,inode)=j
28      end do
29   end do
30
31   do ielem=1,nelem
32      read(100,*) (ine(i,ielem),i=1,2)
33   end do
34   do ielem=1,nelem
35      read(100,*) ???
36      EA(ielem)=???
37      EI(ielem)=???
38   end do
39   do inode=1,nnode
40      read(100,*) ???
41   end do
42   do inode=1,nnode
43      read(100,*) ???
44   end do
45   close(100)
46
47 end program frame_analysis
```

プログラム例 2.1: メインプログラム

1 行目　プログラムの最初には`program`とプログラム名を書く.

2 行目　`implicit none`は宣言文にない変数の使用を禁止する命令である. タイプミスを防ぐことが可能となる.

4–7 行目　宣言文中の `allocatable` は配列の大きさをプログラム中で変更可能とする属性である. 配列の大きさが可変である場合に用いる. ダブルコロン`::`は変数に属性 (`allocatable` や `intent` など) がついている場合, あるいは宣言文中で変数に値を代入する場合に用いられる.

12 行目　`open(100,`*filename*`)`は, ユニット 100 を *filename* というファイルに関連付けるという意味である.

13 行目　装置番号 100から, 値を読み込んでいる.

14–17 行目　`allocatable`な変数の配列の大きさを決める.

20 行目　`(pos(i,ielem),i=1,2)`は `do`を用いないループである. `pos(1,ielem)`, `pos(2,ielem)`と展開される.

47 行目 プログラムの終わりには `end program`を書く.

入力するデータは, 図 2.1 に示す平面骨組構造物とする. この構造物を記述するための情報は以下の通りとする.

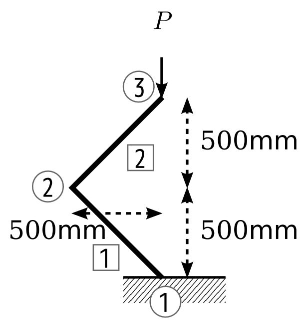


Figure 2.1: 平面骨組構造物

総節点数 = 3 , 総部材数 = 2

節点の座標

節点番号	x 座標	y 座標
1	0mm	0mm
2	-500mm	500mm
3	0mm	1000mm

部材両端における局所節点番号と全体節点番号の関係

部材番号	節点 1	節点 2
1	1	2
2	2	3

部材の物性値

部材番号	はりの高さ	はりの幅	ヤング係数
1	5mm	10mm	100GPa
2	5mm	10mm	100GPa

節点の拘束条件 (拘束あり:1, 拘束なし:0)

節点番号	x 方向の拘束	y 方向の拘束	回転方向の拘束
1	1	1	1
2	0	0	0
3	0	0	0

荷重

節点番号	x 方向の荷重	y 方向の荷重	モーメント荷重
1	0	0	0
2	0	0	0
3	0	-0.98N	0

これらの情報を一つのファイルにまとめる. ファイル名は適当に決めて良い.[1].

[1]単位は kg, m, s に統一すると混乱が少ない

入力データファイル kunoji.dat

```
3 2
0.d0 0.d0
-5.d-1 5.d-1
0.d0 1.d0
1 2
2 3
5.d-3 1.d-2 100.d9
5.d-3 1.d-2 100.d9
1 1 1
0 0 0
0 0 0
0.d0 0.d0 0.d0
0.d0 0.d0 0.d0
0.d0 -0.98d0 0.d0
```

2.2 要素剛性行列の作成

演習 2
要素剛性行列を計算するサブルーチンを作成せよ．

説明 配布されたプログラムでは，elemstiffの呼び出しはコメントされている．コメントを外してから，コンパイルせよ．

```
subroutine elemstiff(nelem,nnode,pos,ine,ielem,EA,EI,se)
  implicit none
  integer nelem,nnode
  real(8) pos(2,nnode), EA, EI,se(6,6)
  integer ine(2,nelem),inode,jnode,ielem,i,j,k
  real(8) lngth, dx(2), cs, sn, r(6,6), sr(6,6)
  inode=ine(1,ielem)
  jnode=ine(2,ielem)
  dx(1)=pos(1,jnode)-pos(1,inode)
  dx(2)=pos(2,jnode)-pos(2,inode)

  lngth=sqrt(dx(1)**2+dx(2)**2)

!!$   *** Compute [SM] ***
  se(1,1)=???
  ???

  do i = 1, 6
     do j = 1, i-1
        se(i,j) = se(j,i)
     end do
  end do

!!$   ***Compute [R] ***
  cs = dx(1) / lngth
  sn = dx(2) / lngth

  r(:,:)=0.d0
```

```
29   r(1,1)=???
30   ???
31
32 !!! S*R
33   sr(:,:) = 0.d0
34   do i = 1, 6
35      do j = 1, 6
36         do k = 1, 6
37            sr(i,j) =???
38         end do
39      end do
40   end do
41
42 !!! R^t*S*R
43   ???
44
45 end subroutine elemstiff
```

プログラム例 2.2: 要素剛性行列を作成するサブルーチン

33–40 行目 行列とベクトルの乗算をしている．単純に組み込み関数 matmulを用いて下記のように書くこともできる．

```
      sr(:,:)=matmul(se(:,:),r(:,:))
```

確認

プログラム例 2.2 の end subroutine の前に以下の文を追記すると，要素剛性行列を出力することができる[2]

```
    do i=1,6
       write(*,'(6e12.3e1)') (se(i,j),j=1,6)
    end do
    write(*,*)
```

kunoji.datを使えば以下のような結果を得るはずである．

```
  0.354E+7 -0.354E+7 -0.884E+2 -0.354E+7  0.354E+7 -0.884E+2
 -0.354E+7  0.354E+7 -0.884E+2  0.354E+7 -0.354E+7 -0.884E+2
 -0.884E+2 -0.884E+2  0.589E+2  0.884E+2  0.884E+2  0.295E+2
 -0.354E+7  0.354E+7  0.884E+2  0.354E+7 -0.354E+7  0.884E+2
  0.354E+7 -0.354E+7  0.884E+2 -0.354E+7  0.354E+7  0.884E+2
 -0.884E+2 -0.884E+2  0.295E+2  0.884E+2  0.884E+2  0.589E+2

  0.354E+7  0.354E+7 -0.884E+2 -0.354E+7 -0.354E+7 -0.884E+2
  0.354E+7  0.354E+7  0.884E+2 -0.354E+7 -0.354E+7  0.884E+2
 -0.884E+2  0.884E+2  0.589E+2  0.884E+2 -0.884E+2  0.295E+2
 -0.354E+7 -0.354E+7  0.884E+2  0.354E+7  0.354E+7  0.884E+2
 -0.354E+7 -0.354E+7 -0.884E+2  0.354E+7  0.354E+7 -0.884E+2
 -0.884E+2  0.884E+2  0.295E+2  0.884E+2 -0.884E+2  0.589E+2
```

[2] '(6e12.3e1)'は出力のフォーマットを指定している．このフォーマットは「全体で 12 文字，小数点以下で 3 文字，指数部 1 文字を使用して，6 つの変数実数を表す」ことを意味している．

2.3　全体剛性行列の作成

演習　3
全体剛性行列を計算せよ．ただし，拘束条件の処理は考えなくてよい．

説明
要素剛性行列を重ねあわせて全体剛性行列を作成する．下記のプログラムにおいて kk は全体剛性行列の行番号あるいは列番号を表す.

```
subroutine globstiff(ielem,ine,s,se,nelem,nnode,mapping)
  integer nelem,ielem, ine(2,nelem),nnode,mapping(3,nnode)
  real(8) se(6,6),s(3*nnode,3*nnode)
  integer:: kk(6)
  integer:: i, j, k, l

  inode=ine(1,ielem)
  jnode=ine(2,ielem)
  kk(1)=???
  ???

  do i=1, 6
     k=???
     do j=1, 6
        l=???
        s(k,l)=s(k,l)+se(i,j)
     end do
  end do
end subroutine globstiff
```

プログラム例　2.3: 全体剛性行列を作成するサブルーチン

確認

全体剛性行列の計算後に，以下を追記すると全体剛性行列を確認することができる.

```
  do i=1,9
     write(*,'(9e10.3e1)') (s(i,j),j=1,9)
  end do
```

```
 0.354E+7 -0.354E+7 -0.884E+2 -0.354E+7  0.354E+7 -0.884E+2  0.000E+0  0.000E+0  0.000E+0
-0.354E+7  0.354E+7 -0.884E+2  0.354E+7 -0.354E+7 -0.884E+2  0.000E+0  0.000E+0  0.000E+0
-0.884E+2 -0.884E+2  0.589E+2  0.884E+2  0.884E+2  0.295E+2  0.000E+0  0.000E+0  0.000E+0
-0.354E+7  0.354E+7  0.884E+2  0.707E+7  0.000E+0  0.000E+0 -0.354E+7 -0.354E+7 -0.884E+2
 0.354E+7 -0.354E+7  0.884E+2  0.000E+0  0.707E+7  0.177E+3 -0.354E+7 -0.354E+7  0.884E+2
-0.884E+2 -0.884E+2  0.295E+2  0.000E+0  0.177E+3  0.118E+3  0.884E+2 -0.884E+2  0.295E+2
 0.000E+0  0.000E+0  0.000E+0 -0.354E+7 -0.354E+7  0.884E+2  0.354E+7  0.354E+7  0.884E+2
 0.000E+0  0.000E+0  0.000E+0 -0.354E+7 -0.354E+7 -0.884E+2  0.354E+7  0.354E+7 -0.884E+2
 0.000E+0  0.000E+0  0.000E+0 -0.884E+2  0.884E+2  0.295E+2  0.884E+2 -0.884E+2  0.589E+2
```

2.4 拘束条件の処理

演習 4
境界条件を考慮し， 全体剛性行列および力のベクトルを処理せよ.

説明
1.4 節を参考にせよ.

```
1  subroutine setbc(nnode,ibc,s,force,u,mapping)
2    implicit none
3    integer nnode,ibc(3,nnode),mapping(3,nnode)
4    real(8) force(3,nnode),s(3*nnode,3*nnode),u(3*nnode)
5    integer:: i, j,inode,k
6  
7    do inode=1,nnode
8       do i=1,3
9          k=???
10         u(k)=???
11      end do
12   end do
13 
14   do inode=1,nnode
15      do i=1,3
16         if(ibc(i,inode)==1) then
17            k=???
18            do j=1,3*nnode
19               s(k,j)=???
20               s(j,k)=???
21            end do
22            s(k,k)=???
23            u(k)=???
24         end if
25      end do
26   end do
27 
28 end subroutine setbc
```

プログラム例 2.4: 境界条件を処理するサブルーチン

10 行目 uは線形方程式の右辺ベクトルである.

確認

境界条件の処理の後に，全体剛性行列を出力すると，以下の結果を得る.

```
0.100E+1  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0
0.000E+0  0.100E+1  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0
0.000E+0  0.000E+0  0.100E+1  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.000E+0
0.000E+0  0.000E+0  0.000E+0  0.707E+7  0.000E+0  0.000E+0 -0.354E+7 -0.354E+7 -0.884E+2
0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.707E+7  0.177E+3 -0.354E+7 -0.354E+7  0.884E+2
0.000E+0  0.000E+0  0.000E+0  0.000E+0  0.177E+3  0.118E+3  0.884E+2 -0.884E+2  0.295E+2
0.000E+0  0.000E+0  0.000E+0 -0.354E+7 -0.354E+7  0.884E+2  0.354E+7  0.354E+7  0.884E+2
0.000E+0  0.000E+0  0.000E+0 -0.354E+7 -0.354E+7 -0.884E+2  0.354E+7  0.354E+7 -0.884E+2
0.000E+0  0.000E+0  0.000E+0 -0.884E+2  0.884E+2  0.295E+2  0.884E+2 -0.884E+2  0.589E+2
```

2.5 変位の計算

演習 5
ガウスの消去法を用いて各節点の変位を求めよ．メインプログラムで，サブルーチン solveleを呼び出せばよい．

説明
以下は，線形方程式 $\boldsymbol{Ax} = \boldsymbol{b}$ を解くサブルーチンである．サブルーチンを呼び出すとき，変数 xには右辺ベクトル $\boldsymbol{b}$ の値が入っている．サブルーチンから戻るとき，変数 xには方程式の解 $\boldsymbol{x}$ の値が入っている．call solvele (s,u,3*nnode)と subroutine solvele(a,x,n)でサブルーチンの引数が異なるが，変数の型が同じであれば問題ない．

```
subroutine solvele(a,x,n)
  implicit none
  integer n
  real(8) a(n,n),x(n)
  integer i,j,k

  do i=1,n-1
     do j=i+1, n
        x(j)=x(j)-x(i)*a(j,i)/a(i,i)
        do k=i+1,n
           a(j,k)=a(j,k)-a(i,k)*a(j,i)/a(i,i)
        end do
     end do
  end do

  x(n)=x(n)/a(n,n)
  do i=n-1,1,-1
     do j=i+1,n
        x(i)=x(i)-a(i,j)*x(j)
     end do
     x(i)=x(i)/a(i,i)
  end do
end subroutine solvele
```

プログラム例 2.5: ガウスの消去法のサブルーチン

確認

dispを求めた後に以下のコードを挿入し，各ノードでのひずみを確認する．

```
  do inode=1,nnode
     write(*,'(3e12.3e1)') (disp(i,inode),i=1,3)
  end do
```

```
   0.000E+0    0.000E+0    0.000E+0
   0.277E-2    0.277E-2   -0.166E-1
   0.166E-1   -0.111E-1   -0.333E-1
```

2.6 曲げモーメント，縁ひずみの計算

> **演習 6**
> はり要素中央における曲げモーメントと上縁ひずみを求めよ.

```
1  subroutine compstrain(nelem,ine,pos,disp,hght,EI,nnode)
2    implicit none
3    integer nelem,nnode,ine(2,nelem)
4    real(8) pos(2,nnode), disp(3,nnode), hght(nelem),EI(nelem)
5    real(8) lngth, dx(2), cs, sn
6    real(8) v(2),psi(2),c2,c3,bm,tops
7    integer inode,jnode,ielem
8    do ielem=1,nelem
9       inode=ine(1,ielem)
10      jnode=ine(2,ielem)
11      dx(1)=pos(1,jnode)-pos(1,inode)
12      dx(2)=pos(2,jnode)-pos(2,inode)
13      lngth = sqrt(dx(1)**2+dx(1)**2)
14      cs = dx(1) / lngth
15      sn = dx(2) / lngth
16      v(1)=-sn*disp(1,inode)+cs*disp(2,inode)
17      v(2)=-sn*disp(1,jnode)+cs*disp(2,jnode)
18      psi(1)=disp(3,inode)
19      psi(2)=disp(3,jnode)
20      c2=???
21      c3=???
22      bm=???
23      tops=???
24   end do
25 end subroutine compstrain
```

プログラム例 2.6: 縁ひずみの計算

22–23 行目 bm, tops はそれぞれ部材中央の曲げモーメント，上縁ひずみである.

確認

23 行目の後に以下のコードを挿入し，部材中央でのひずみを確認する.

```
write(*,*) ielem,tops
```

kunoji.datを使えば以下の結果を得る[3].

```
1        -5.8799999999647004E-005
2        -5.8799999999839599E-005
```

2.7 可視化

> **演習 7**
> 変形を可視化するため，データを出力するプログラムを作成せよ．可視化は gnuplot を用いる.

[3]誤差を含むので完全には一致しないことがある

```
1  subroutine outputd(nelem,nnode,ine,pos,disp)
2    implicit none
3    integer nelem,nnode, ine(2,nelem)
4    real(8) pos(2,nnode), disp(3,nnode)
5    real(8) lngth, dx(2), cs, sn
6    real(8) a0, a1, c0, c1, c2, c3
7    real(8) u(2), v(2), psi(2)
8    real(8) xbar, xini, yini, ubar, vbar, udef, vdef
9    integer j, inode, jnode, ielem
10   open(200,file='deformation.dat')
11
12   do ielem = 1, nelem
13      inode=ine(1,ielem)
14      jnode=ine(2,ielem)
15      dx(1)=pos(1,jnode)-pos(1,inode)
16      dx(2)=pos(2,jnode)-pos(2,inode)
17      lngth=sqrt(dx(1)**2+dx(2)**2)
18      cs=dx(1)/lngth
19      sn=dx(2)/lngth
20
21      u(1)=cs*disp(1,inode)+sn*disp(2,inode)
22      v(1)=-sn*disp(1,inode)+cs*disp(2,inode)
23      psi(1)=disp(3,inode)
24      u(2)=cs*disp(1,jnode)+sn*disp(2,jnode)
25      v(2)=-sn*disp(1,jnode)+cs*disp(2,jnode)
26      psi(2)=disp(3,jnode)
27
28      a0=u(1)
29      a1=-u(1)/lngth+u(2)/lngth
30      c0=v(1)
31      c1=psi(1)
32      c2=-3.d0*v(1)/lngth**2-2.d0*psi(1)/lngth &
33           & +3.d0*v(2)/lngth**2-psi(2)/lngth
34      c3=2.d0*v(1)/lngth**3+psi(1)/lngth**2 &
35           & -2.d0*v(2)/lngth**3+psi(2)/lngth**2
36      do j=1,11
37         xbar=lngth/10*(j-1)
38         xini=dx(1)/10*(j-1)+pos(1,inode)
39         yini=dx(2)/10*(j-1)+pos(2,inode)
40         ubar=a0+a1*xbar
41         vbar=c0+(c1+(c2+c3*xbar)*xbar)*xbar
42         udef=???
43         vdef=???
44         write(200,*) xini, yini, udef, vdef
45      end do
46      write(200,*)
47   end do
48   close(200)
49 end subroutine outputd
```

プログラム例 2.7: 可視化用のデータを出力するサブルーチン

13 行目 `write(200,*)`でユニット番号200に書き込みをする.

21–26 行目　全体変位から局所変位を計算する．

36–45 行目　はり要素を 10 分割し，変形量を計算する．`xbar` は局所座標系における $\bar{x}$ 座標, `xini`，`yini` はビームの初期位置 (x, y)，`ubar`，`vbar` は局所座標系における変位 $(\bar{u}, \bar{v})$，`udef`，`vdef` は全体座標系における変位 (u, v) である．

44 行目

`deformation.dat`の 1–4 列目には，それぞれ，はり要素の x 座標，y 座標，x 方向の変位，y 方向の変位が入っている．これを gnuplot で描画する場合，`gnuplot`を立ち上げてから，以下のコマンドを入力すれば良い[4]．

```
1 set size ratio -1
2 unset border
3 unset tics
4 unset key
5 a=1.
6 plot 'deformation.dat' using ($1+a*$3):($2+a*$4) with lines
7 set terminal postscript eps color
8 set output 'def.eps'
9 replot
```

1 行目　縦横比を 1:1 にする

2–4 行目　軸，メモリ，キーを表示しない

5 行目　`a`は変形の倍率を表す．変形を大きく表現したいときは `a`の値を大きくする．

6 行目　`deformation.dat`をプロットする．`using ($1+a*$3):($2+a*$4)`は 1 列目$+a\times$(3 列目の値) を x 軸に，2 列目$+a\times$(4 列目の値) を y 軸に取るという意味である．変形が小さくて見にくい場合は，5 行目に戻り `a`の値を大きくして再度 plot する．もとの変形と比較したい場合は，`plot 'deformation.dat' using ($1+a*$3):($2+a*$4) with lines,'deformation.dat' with lines` と入力すれば良い．

7–8 行目　6 行目ではディスプレイに出力していたが，出力先を eps ファイルに変更する．eps ファイルの作成が不要であれば，これ以降のコマンドは不要である．

9 行目　前に plot した内容を再びプロットする．出力先を eps ファイルに変更したので，eps ファイルにプロットする必要がある．

2.8　単純ばりの問題

演習　8
単純ばりの中央に鉛直下向きの集中荷重を作用させる問題を考える．この問題のデータファイルを作成し，プログラムを用いてはり中央のひずみと端点でのひずみ角を計算せよ．また，その求めた値が構造力学で学んだはり理論の解とあうか確認せよ．

簡単のため，はりの長さ $L = 1$，荷重 $P = 1$ とせよ．また，$EI = 1$ となるように，$h = 1, b = 12, E = 1$ とせよ．片持ちばりの自由端に集中荷重を作用させたとき，はり中央のたわみおよび端点でのたわみ角はそれぞれ次式で与えられる．

$$v = \frac{PL^3}{48EI}, \qquad v' = \frac{PL^2}{16EI} \tag{2.1}$$

図 2.2 のように 2 要素 3 節点を考える．`disp(2,2)` は部材中央の y 方向変位，`disp(3,1)` は自由端のたわみ角に対応する．線形計算をした後に以下のコードを挿入すれば，たわみとたわみ角が得られる．

```
      write(*,*) disp(2,2),disp(3,1)
```

[4] 別の方法として，四角の中の命令を `draw.gp` という名前で保存し，端末で `gnuplot draw.gp` とするやり方もある．

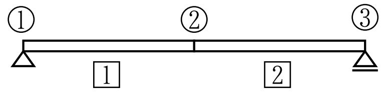


Figure 2.2: 単純ばり

2.9　フレーム実験との比較

演習　9
構造実験で実施したフレーム試験を数値計算により再現する．計測した変位やひずみと数値計算によって得られた値を比較せよ．

Appendix A

A.1 変数の意味

プログラムで用いる変数を以下に示す.

Table A.1: 解析に用いる主要な変数の説明

変数	変数の意味	meaning of variables
`nnode`	節点数	number of nodes
`nelem`	要素数	number of elements
`ine(:,:)`	接続のリスト	connectivity array
`ibc(:,:)`	境界条件 (0:自由, 1:固定)	boundary condition(0:free, 1:fix)
`pos(:,:)`	座標	coordinate
`force(:,:)`	外力	external force
`disp(:,:)`	変位	displacement
`hght(:)`	はりの高さ	height of a beam
`wdth(:)`	はりの幅	width of a beam
`ym(:)`	ヤング率	Young's modulus
`EA(:)`	EA	EA
`EI(:)`	EI	EI
`s(:,:)`	全体剛性行列	Global stiffness matrix
`se(6,6)`	要素剛性行列	Element stiffness matrix

A.2 Fortran の基礎

gfortran のコンパイル，実行は下記のように行う．

```
gfortran filename.f90
./a.out
```

Fortran のプログラムで知っておくべきことを以下に列挙する．

- 大文字と小文字は区別しない．
- `!`より右側はコメントとなり，無視される．
- べき乗の計算は`**2`，平方根は `sqrt()`を用いる．
- 6.17×10^{-3} という値は，倍精度を用いる場合，`6.17d-3`と表現する．

A.3 デバグ

デバグの際には以下のコンパイルオプションを付けると，エラーを発見しやすい．

Table A.2: gfortran のコンパイルオプション

意味	オプション
すべての警告メッセージを出す	-Wall
配列外参照の検出 [1]	-fbounds-check
初期化されていない変数の検出 [2]	-O -Wuninitialized
浮動小数点例外 [3]	-ffpe-trap=invalid,zero,overflow,underflow
問題となる行番号の表示	-fbacktrace

例えば，配列外参照，ゼロ割，行の表示のオプションを付ける場合，以下のようにコンパイルする．

```
gfortran -fbounds-check -ffpe-trap=zero -fbacktrace filename.f90
```

デバッガを用いてデバグする方法もある．デバッガ（gdb）を用いる場合は，コンパイル時に`-g` のオプションを付ける．gdb を終了するときは `quit` と入力する．

```
gfortran filename.f90 -g
gdb ./a.out
(gdb) run
```

[1]例えば，2 行 2 列の行列 `a(2,2)` に (2,3) 成分の `a(2,3)` を入れようとするとエラーを検出する．
[2]変数の初期値が代入されていない場合エラーを検出する．
[3]不適切な計算，0 による除算，非常に大きな値，非常に小さな値が得られた時にエラーを検出する