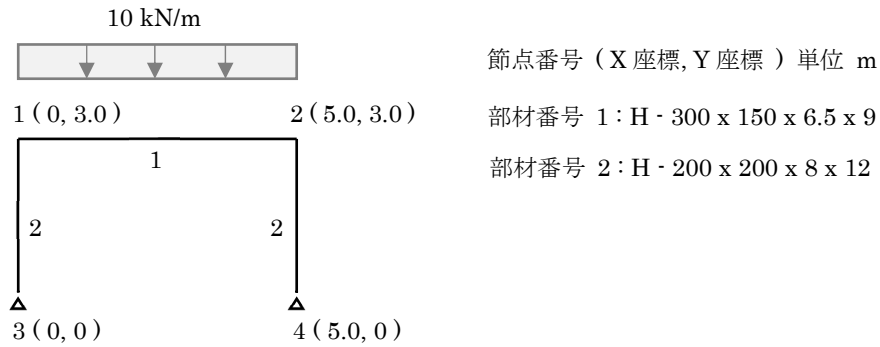


この文書は、平面骨組の応力計算を行うプログラムのソースコード stiff2D.js の内容について解説するものです。

● 例題

全体の流れを把握するために、以下のような例題を取り上げて本ソースコードの利用方法を紹介します。



行番号

```
001    let obj = new stiff2D({ e: 205000 });
002    obj.addNodeByName({ id: 1, x: 0, y: 3 });
003    obj.addNodeByName({ id: 2, x: 5, y: 3 });
004    obj.addNodeByName({ id: 3, x: 0, y: 0; xFix: 1, yFix: 1 });
005    obj.addNodeByName({ id: 4, x: 5, y: 0; xFix: 1, yFix: 1 });
006    obj.addMembByName({ id: 1, iNode: 1, jNode: 2, a: 46.8, ix: 7210 });
007    obj.addMembByName({ id: 2, iNode: 3, jNode: 1, a: 63.5, ix: 4720 });
008    obj.addMembByName({ id: 3, iNode: 4, jNode: 2, sectionId: 2 });
009    obj.addMembLoadByName ({ num: 1, ci: -20.83, cj: 20.83, m0: 31.25, qi: 25, qj: 25});
010    if (obj.stiffMain() > 0) {
011        console.log("ID  Mi  Mj  Mc  Qi  Qj  Ni  Nj");
012        console.log(obj.getAllMembStress());
013    } else {
014        console.log(obj.error);
015    }
```

コードの説明

行 001

stiff2D クラスのオブジェクト obj を作成。この時、e でヤング係数の初期値 205000(N/mm²) を設定している。ここではせん断変形は考慮しないので、せん断弾性係数の初期値等は省略。

行 002 - 005

節点情報の設定。id が節点番号、x が X 座標(m)、y が座標。ここでは節点番号を 1 から順に振っているが、重複

しない正の整数であれば任意の値でよい。その他の引数は支点拘束の指定になるが、節点 1 と 2 は何も拘束がないのですべて省略(行 002-003)。

節点 3 と 4 はピン支点で、X・Y 方向が拘束されているので **xFix** と **yFix** を 1 とする(行 004-005)。

行 006 - 008

部材情報の設定。**id** が部材番号で、重複のない正の整数であれば任意の値でよい。**iNode** が始端の節点番号、**jNode** が終端の節点番号、**a** が部材の断面積(cm^2)、**ix** が断面 2 次モーメント(cm^4)。ヤング係数は初期値を使用し、さらに両端剛接合で剛域長の指定もないので、その他の引数はすべて省略。

部材番号 3 は 2 と同じ断面性能なので、**sectionID** を 2 とし、断面性能の値は省略(行 008)。

行 009

梁の荷重項の設定。**num** は部材番号、**ci** は始端・**cj** は終端の固定端モーメント($\text{kN}\cdot\text{m}$)、**m0** は単純梁の中央モーメント、**qi** は始端・**qj** は終端のせん断力(kN)。ここでは軸力はないので、他の引数は省略。

行 010

stiffMain を呼び出して応力計算を開始。正常終了した場合は 1、それ以外は 0 の戻り値が返る。

行 011

計算結果のヘッダーを出力。

行 012

全部材の応力を出力。各部材ごとに改行された計 3 行の値が出力される。

行 014

計算時に何らかのエラーがあった場合はその内容を画面上に出力。

● プロパティ

defaultE

ヤング係数の初期設定値。各部材の入力時に値が省略されたものについてはこの値を適用。入力単位は N/mm^2 だが、プログラム内では kN/cm^2 に変換している。

defaultG

せん断弾性係数の初期設定値。各部材の入力時に値が省略されたものについてはこの値を適用。せん断変形を考慮しないのであれば入力不要。単位については **defaultE** に同じ。

inpRotate

回転成分については、プログラム内では「反時計回りが正」としているが、荷重項の入力等では慣用的に「時計回りが正」とすることが多いので、この値が 0 (初期値)の場合は「時計回りが正」として入出力を行う。

node[] : 節点情報を格納する二次元配列。以下の **key** をもつ。

id : 節点番号 ≥ 1

x : X 座標 (cm) 入力値は m

y : Y 座標 (cm) 入力値は m

xFix : X 方向が拘束されている場合は 1

yFix : Y 方向が拘束されている場合は 1

rFix：回転方向が拘束されている場合は 1

以下は計算結果

dx：X 方向の変位量 (mm)

dy：Y 方向の変位量 (mm)

dr：回転方向の変位量 (rad)

memb[]：部材情報を格納する二次元配列。以下の key をもつ。

id：部材番号 ≥ 1

iNode：始端の節点番号 ≥ 1

jNode：終端の節点番号 ≥ 1

a：断面積 (cm^2)

ix：断面 2 次モーメント (cm^4)

as：せん断断面積 (cm^2) せん断変形を考慮する場合

e：ヤング係数 (kN/cm^2) 入力値は N/mm^2 省略時は defaultE

g：せん断弾性係数 (kN/cm^2) 入力値は N/mm^2 省略時は defaultG

iPin：始端がピンの場合は 1

jPin：終端がピンの場合は 1

iRigid：始端の剛域長 (cm) 入力値は m

jRigid：終端の剛域長 (cm) 入力値は m

以下は計算結果

mi：始端の曲げモーメント ($\text{kN}\cdot\text{m}$)

mj：終端の曲げモーメント ($\text{kN}\cdot\text{m}$)

mc：中央部の曲げモーメント ($\text{kN}\cdot\text{m}$)

qi：始端のせん断力 (kN)

qj：終端のせん断力 (kN)

ni：始端の軸力 (kN)

nj：終端の軸力 (kN)

nodeLoad[]：節点荷重情報を格納する二次元配列。以下の key をもつ。

num：荷重が作用する節点番号 ≥ 1

px：X 方向の荷重 (kN)

py：Y 方向の荷重 (kN)

pr：曲げモーメント ($\text{kN}\cdot\text{m}$)

membLoad[]：部材荷重情報を格納する二次元配列。以下の key をもつ。

num：荷重が作用する部材番号 ≥ 1

ci：始端の固定端モーメント ($\text{kN}\cdot\text{cm}$) 入力値は $\text{kN}\cdot\text{m}$

cj：終端の固定端モーメント ($\text{kN}\cdot\text{cm}$) 入力値は $\text{kN}\cdot\text{m}$

m0：単純梁の中央モーメント ($\text{kN}\cdot\text{cm}$) 入力値は $\text{kN}\cdot\text{m}$

qi：始端のせん断力 (kN)

qj：終端のせん断力 (kN)

ni：始端の軸力 (kN)

nj：終端の軸力 (kN)

nodeIndex[i]：入力された節点番号 i に対応する node[] の index

membIndex[i]：入力された部材番号 i に対応する memb[] の index

error：計算実行時に検出されたエラーをあらわす文字列

● メソッド

- ・ constructor (e = 0, g = 0, rotate = 0)

プロパティ defaultE, defaultG, inpRotate を設定するコンストラクタ

- ・ addNode (id = 0, x = 0, y = 0, xFix = 0, yFix = 0, rFix = 0)

節点情報 node[] を設定。addNodeByName はこれを名前付き引数で指定するもの。

- ・ addMemb (id = 0, iNode = 0, jNode = 0, a = 0, ix = 0, as = 0, e = 0, g = 0, iPin = 0, jPin = 0, iRigid = 0, jRigid = 0, sectionID = 0)

部材情報 memb[] を設定。addMembByName はこれを名前付き引数で指定するもの。sectionID に何らかの値がある場合は、その id 値をもつ部材と同じ断面性能がセットされる。

- ・ addNodeLoad (num = 0, px = 0, py = 0, pr = 0)

節点荷重情報 nodeLoad[] を設定。addNodeLoadByName はこれを名前付き引数で指定するもの。

- ・ addMembLoad (num = 0, ci = 0, cj = 0, m0 = 0, qi = 0, qj = 0, ni = 0, nj = 0, loadId = 0)

部材荷重情報 membLoad[] を設定。addMembLoadByName はこれを名前付き引数で指定するもの。LoadId に何らかの値がある場合は、その id 値をもつ部材と同じ断面性能がセットされる。

- ・ getNodeDisp (num)

節点番号 num の節点変位量を取得。

戻り値 result{}

.dx：X 方向変位量 (mm), .dy：Y 方向変位量 (kN), .dr：回轉變位量 (rad)

- ・ getAllNodeDisp ()

戻り値 全節点の変位量をあらわす文字列。

「節点番号 X 方向変位量 Y 方向変位量 回轉變位量」の書式で、節点ごとに改行される。

- ・ getMembStress (num)

部材番号 num の応力を取得。

戻り値 result{}

.mi：始端の曲げモーメント (kN.m), .mj：終端の曲げモーメント, .mc：中央部の曲げモーメント

.qi：始端のせん断力 (kN), .qj：終端のせん断力, .ni：始端の軸力 (kN), .nj：終端の軸力

- ・ getAllMembStress ()

戻り値 全部材の応力をあらわす文字列。

「部材番号 Mi Mj Mc Qi Qj Ni Nj」の書式で、部材ごとに改行される。

- **stiffMain ()**

応力計算の制御を行う。応力計算の実行時に呼び出される。

戻り値 1：正常終了， 0：異常終了

- **globalMatrix (free, diagNum, maxFree)**

全体剛性マトリクスの作成。

戻り値 bandMatrix[] 全体剛性マトリクスのバンドマトリクス

free[i][j]：節点 node[i] の自由度番号 (j = 0 : X 方向, 1 : Y 方向, 2 : 回転方向)

diagNum[i]：全体剛性マトリクスの i 列の対角要素の自由度番号

maxFree：全体剛性マトリクスの列数

- **membMatrix (m, code = 0)**

部材 memb[m] の部材剛性マトリクスの作成。

戻り値 stiff[6][6] 部材剛性マトリクス

code = 0：部材座標系, 1：全体座標系

- **membLength (m)**

部材 memb[m] の諸元。

戻り値 [al, cs, sn]

al：部材の長さ, cs : $\cos \theta$, sn : $\sin \theta$

- **membTrans (cs, sn)**

部材の座標変換マトリクスの作成。

戻り値 trans[6][6] 座標変換マトリクス

cs : $\cos \theta$, sn : $\sin \theta$

- **loadVector (free, maxFree, cmqIndex)**

外力ベクトルの作成。

戻り値 vect[] 外力ベクトル

free[i][j]：節点 node[i] の自由度番号 (j = 0 : X 方向, 1 : Y 方向, 2 : 回転方向)

maxFree：外力ベクトルの要素数

cmqIndex[i] : memb[i] に存在する部材荷重 membLoad[] の index (指定がない場合は -1)

- **getCmq (n)**

membLoad[n] の荷重項の値。

戻り値 cmq[i] 荷重項のベクトル (i = 0 : Ni, 1 : Qi, 2 : Mi, 3 : Nj, 4 : Qj, 5 : Mj, 6 : M0)

- **preGauss (a, diagNum, numMax)**

バンドマトリクス a[] を上三角マトリクスに変換。

戻り値 1：正常終了， 0：異常終了

diagNum[i]：バンドマトリクスの i 列の対角要素の自由度番号

numMax：バンドマトリクスの列数 - 1

- ・ `solveGauss (a, diagNum, numMax, r)`
 ガウスの消去法(後退代入)による連立方程式の解法。
 戻り値 `result[]` : 解のベクトル
`a[]` : バンドマトリクス
`diagNum[i]` : バンドマトリクスの `i` 列の対角要素の自由度番号
`numMax` : バンドマトリクスの列数 - 1
`r[]` : 係数ベクトル
- ・ `setDisp (free, disp)`
 節点の変位量を `node[]` に格納。
 戻り値 なし
`free[i][j]` : 節点 `node[i]` の自由度番号 (`j = 0` : X 方向, `1` : Y 方向, `2` : 回転方向)
`disp[]` : 全自由度の変位ベクトル
- ・ `setStress (free, disp, cmqIndex)`
 部材の応力を `memb[]` に格納。
 戻り値 なし
`free[i][j]` : 節点 `node[i]` の自由度番号 (`j = 0` : X 方向, `1` : Y 方向, `2` : 回転方向)
`disp[]` : 全自由度の変位ベクトル
`cmqIndex[i]` : `memb[i]` に存在する部材荷重 `membLoad[]` の index (指定がない場合は -1)
- ・ `preCheck()`
 入力データの検証。
 戻り値 `1` : エラーなし, `0` : エラーあり