

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

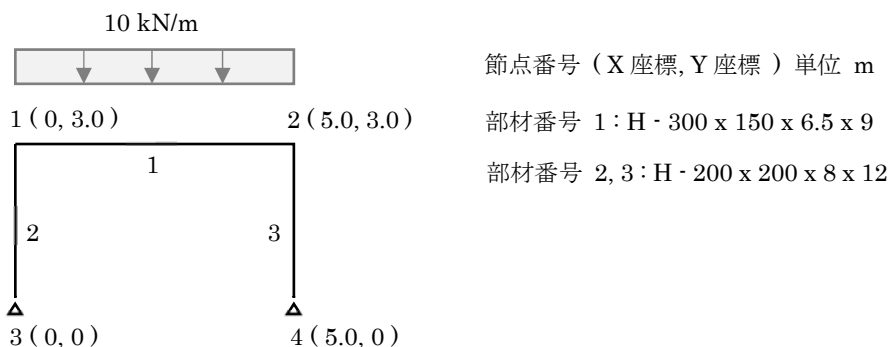
このプログラムを実行するためには Python の実行環境を整える必要がありますが、これについては説明を省略しています。小社のウェブサイト https://www.structure.jp/py_install.html などを参照してください。

● 概要

このプログラムでは、入力データの読み込み、ならびに計算結果の出力にマイクロソフト製の Excel を使用していますが、ファイルの拡張子は*.xlsx になります。したがって Excel 2010 以降のプログラムが別途必要です。

初期設定では、入力用ファイル名が `data.xlsx`、出力用ファイル名が `result.xlsx` で、これらが `stiff2D.py` と同じフォルダ内にあることを前提にしています(ファイルの場所と名前は変更可能)。

解凍されたファイルの中に `data.xlsx` がありますので、これを開いてください。ここにはサンプルとして、以下のような骨組のデータが書き込まれています。



上記のファイルをいったん閉じてから `stiff2D.py` を実行してみてください。コンソールに「正常終了」と表示されると、同じフォルダ内に `result.xlsx` というファイルが作成されます。これを開くと、上記の骨組の計算結果が書き込まれていることが確認できるはずです。

● 入力ファイル *.xlsx

ここには「節点」「部材」「節点荷重」「部材荷重」の 4 つのワークシートがあります。

節点

節点情報を入力するシート。1 行目はヘッダーで、2 行目以降に節点数分のデータを入力します。

1 列目 節点番号

節点を識別するための値で、必ずしも連続した番号である必要はありませんが、一般には 1 から始まる番号を連続的に振ります。

2-3 列目 X 座標 Y 座標 (m)

各節点の座標値。

4-6 列目 X 方向 Y 方向 回転方向

これは節点の拘束状態(支点)をあらわすもので、その方向が拘束されている場合は 1 とします。拘束がない場合

は 0 または空白とします。

部材

部材情報を入力するシート。1 行目はヘッダーで、2 行目以降に部材数分のデータを入力します。

1 列目 部材番号

部材を識別するための値で、必ずしも連続した番号である必要はありませんが、一般には 1 から始まる番号を連続的に振ります。

2-3 列目 始端番号 終端番号

部材の両側の節点番号（「節点」シートで指定したもの）。どちらを始端(i 端)または終端(j 端)とするかは任意。

4-6 列目 部材の断面性能

断面積 A ・断面 2 次モーメント I ・せん断断面積 A_s を cm 単位で入力します。せん断剛性を無視する場合は A_s は入力不要です。

7-8 列目 部材の材料定数 (N/mm^2)

ヤング係数 E ・せん断弾性係数 G を N/mm^2 単位で入力します。せん断剛性を無視する場合は G は入力不要です。

11-12 列目 部材の剛域長 (m)

始端または終端に剛域を考慮する場合は、その値を m 単位で入力します。

節点荷重

節点荷重の情報を入力するシート。1 行目はヘッダーで、2 行目以降に(節点荷重がある場合は)荷重数分のデータを入力します。

1 列目 節点番号

荷重が作用する節点番号（「節点」シートで指定したもの）。

2-4 列目 節点荷重の値

X 方向・ Y 方向に作用する荷重の値を kN 単位、回転方向に作用する曲げ荷重を $\text{kN}\cdot\text{m}$ 単位で入力します。曲げ荷重の符号は「時計回りが正」とします。

部材荷重

部材荷重(部材に作用する中間荷重)の情報を入力するシート。1 行目はヘッダーで、2 行目以降に(部材荷重がある場合は)荷重数分のデータを入力します。なお、節点荷重・部材荷重のいずれかの指定が最低 1 つ必要です。

1 列目 部材番号

荷重が作用する部材番号（「部材」シートで指定したもの）。

2-8 列目 部材荷重の値

部材の荷重項の値。始端の固定端曲げ C_i ・終端の固定端曲げ C_j ・単純梁の中央曲げ M_0 ・始端の固定端せん断力 Q_i ・終端のせん断力 Q_j ・始端の軸力 N_i ・終端の軸力 N_j の値を、曲げについては $\text{kN}\cdot\text{m}$ 、その他は kN 単位で入力します。曲げ荷重の符号は「時計回りが正」とします。

● 出力ファイル *.xlsx

ここには「節点変位」「部材応力」の 2 つのワークシートがあります。

節点変位

1 列目 節点番号

「節点」シートで指定されたもの。

2-3 列目 X 方向変位 Y 方向変位 (mm) 回転変位 (rad)

X 方向・Y 方向の節点変位ならびに回転変位量を出力します。

部材応力

1 列目 部材番号

「部材」シートで指定されたもの。

2-8 列目 部材応力

始端の曲げ M_i ・終端の曲げ M_j ・中央部の曲げ M_c ・始端のせん断力 Q_i ・終端のせん断力 Q_j ・始端の軸力 N_i ・終端の軸力 N_j の値を、曲げについては $\text{kN}\cdot\text{m}$ 、その他は kN 単位で出力します。

ここでは `Stiff2D` という名称のクラスを作っていますので、以下、その内容について簡単に解説します。

● 初期化メソッド

・ `__init__` (`inp_path = 'data.xlsx'`, `out_path = 'result.xlsx'`)

`inp_path` : 入力ファイル(*.xlsx)のパスを `stiff2D.py` からの相対パス、または絶対パスで指定する。省略時は `stiff2D.py` と同じフォルダ内にある `data.xlsx`

`out_path` : 出力ファイル(*.xlsx)のパスを `stiff2D.py` からの相対パス、または絶対パスで指定する。省略時は `stiff2D.py` と同じフォルダ内にある `result.xlsx`

● プロパティ

`node[i][j]` : 節点情報を格納する二次元リスト

要素 j の値は以下の通り

[0] : 節点番号 ≥ 1

[1] : X 座標 (cm) 入力値は m

[2] : Y 座標 (cm) 入力値は m

[3] : X 方向が拘束されている場合は 1

[4] : Y 方向が拘束されている場合は 1

[5] : 回転方向が拘束されている場合は 1

以下は計算結果

[6] : X 方向の変位量 (mm)

[7] : Y 方向の変位量 (mm)

[8] : 回転方向の変位量 (rad)

`memb[i][j]`: 部材情報を格納する二次元リスト

要素 `j` の値は以下の通り

- [0]: 部材番号 ≥ 1
- [1]: 始端の節点番号 ≥ 1
- [2]: 終端の節点番号 ≥ 1
- [3]: 断面積 (cm^2)
- [4]: 断面 2 次モーメント (cm^4)
- [5]: せん断断面積 (cm^2) せん断変形を考慮する場合
- [6]: ヤング係数 (kN/cm^2) 入力値は N/mm^2
- [7]: せん断弾性係数 (kN/cm^2) 入力値は N/mm^2
- [8]: 始端がピンの場合は 1
- [9]: 終端がピンの場合は 1
- [10]: 始端の剛域長 (cm) 入力値は m
- [11]: 終端の剛域長 (cm) 入力値は m

以下は計算結果

- [12]: 始端の曲げモーメント (kN.m)
- [13]: 終端の曲げモーメント (kN.m)
- [14]: 中央部の曲げモーメント (kN.m)
- [15]: 始端のせん断力 (kN)
- [16]: 終端のせん断力 (kN)
- [17]: 始端の軸力 (kN)
- [18]: 終端の軸力 (kN)

`node_load[i][j]`: 節点荷重情報を格納する二次元リスト

要素 `j` の値は以下の通り

- [0]: 荷重が作用する節点番号 ≥ 1
- [1]: X 方向の荷重 (kN)
- [2]: Y 方向の荷重 (kN)
- [3]: 曲げモーメント (kN.m)

`memb_load[i][j]`: 部材荷重情報を格納する二次元リスト

要素 `j` の値は以下の通り

- [0]: 荷重が作用する部材番号 ≥ 1
- [1]: 始端の固定端モーメント (kN.cm) 入力値は kN.m
- [2]: 終端の固定端モーメント (kN.cm) 入力値は kN.m
- [3]: 単純梁の中央モーメント (kN.cm) 入力値は kN.m

[4] : 始端のせん断力 (kN)

[5] : 終端のせん断力 (kN)

[6] : 始端の軸力 (kN)

[7] : 終端の軸力 (kN)

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

memb_index[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

- add_node (id = 0, x = 0, y = 0, xfix = 0, yfix = 0, rfix = 0)
入力ファイルによらず、節点情報 node[] を直接設定する場合に使用。
引数の意味についてはソースコードのコメントを参照。
- add_all_node (list2)
入力ファイルから節点情報 node[] を取得する。
list2 は Excel から読み込まれたセルの値を格納した二次元リスト。list[0] はヘッダーの文字列。
- add_memb (id = 0, inode = 0, jnode = 0, a = 0, ix = 0, as = 0, e = 0, g = 0, ipin = 0, jpin = 0, irigid = 0, jrigid = 0)
入力ファイルによらず、部材情報 node[] を直接設定する場合に使用。
引数の意味についてはソースコードのコメントを参照。
- add_all_memb (list2)
入力ファイルから部材情報 memb[] を取得する。
list2 は Excel から読み込まれたセルの値を格納した二次元リスト。list[0] はヘッダーの文字列。
- add_node_load (num = 0, px = 0, py = 0, pr = 0)
入力ファイルによらず、節点荷重情報 node_load[] を直接設定する場合に使用。
引数の意味についてはソースコードのコメントを参照。
- add_all_node_load (list2)
入力ファイルから節点荷重情報 node_load[] を取得する。
list2 は Excel から読み込まれたセルの値を格納した二次元リスト。list[0] はヘッダーの文字列。
- add_memb_load (num = 0, ci = 0, cj = 0, m0 = 0, qi = 0, qj = 0, ni = 0, nj = 0)
入力ファイルによらず、部材荷重情報 memb_load[] を直接設定する場合に使用。
引数の意味についてはソースコードのコメントを参照。
- add_all_memb_load (list2)
入力ファイルから部材荷重情報 memb_load[] を取得する。
list2 は Excel から読み込まれたセルの値を格納した二次元リスト。list[0] はヘッダーの文字列。
- calc_main ()

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

- `global_matrix (free, diag_num, max_free)`

全体剛性マトリクスの作成。

戻り値 `band[]` 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)

`diag_num[i]`：全体剛性マトリクスの i 列の対角要素の自由度番号

`max_free`：全体剛性マトリクスの列数

- `memb_matrix (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

- `memb_trans (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` : $\sin \theta$

- `pre_gauss (a, diag_num, num_max)`

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

`diag_num[i]`：バンドマトリクスの i 列の対角要素の自由度番号

`num_max`：バンドマトリクスの列数 - 1

- `solve_gauss (a, diag_num, num_max, 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

`diag_num[i]`：バンドマトリクスの i 列の対角要素の自由度番号

`num_max`：バンドマトリクスの列数 - 1

`r[]`：係数ベクトル

- `load_vector (free, max_free, cmq_index)`

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)

`max_free`：外力ベクトルの要素数

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

- `cmq_vector (n , m)`

`memb_load[n]` `memb[m]` の荷重項の値。

戻り値 `cmq[i]` 荷重項のベクトル ($i = 0 : N_i, 1 : Q_i, 2 : M_i, 3 : N_j, 4 : Q_j, 5 : M_j, 6 : M_0$)

- `set_node_disp (free, disp)`

全節点の変位量を `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

- `set_memb_stress (free, disp, cmq_index)`

全部材の応力を `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

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

- `pre_check()`

入力データの検証。

戻り値 1: エラーなし, 0: エラーあり

- `memb_length (m, result)`

部材 `memb[m]` の諸元。

戻り値 (`al, cs, sn`) `al`: 部材の長さ, `cs`: $\cos \theta$, `sn`: $\sin \theta$

`result = 1` の場合は `al` のみ

- `read_data_xlsx ()`

入力ファイル `inp_path (*.xlsx)` からデータを読み込む。

- `write_result_xlsx ()`

出力ファイル `out_path (*.xlsx)` に計算結果を書き込む。