

Marc MentatのPython機能を介したトポロジー最適化の検討

Study on Topology Optimization Using the Python Function of Marc Mentat

横井宏昭¹⁾, 荒井皓一郎²⁾

Hiroaki Yokoi and Koichiro Arai

1) HEXAGON (〒101-0054 東京都千代田区神田錦町2-2-1 KANDA SQUARE 16F, E-mail: hiroaki.yokoi@hexagon.com)

2) HEXAGON (〒101-0054 東京都千代田区神田錦町2-2-1 KANDA SQUARE 16F, E-mail: koichiro.arai@hexagon.com)

Nonlinear finite element analysis commercial software Marc Mentat has a dedicated Python module as API. This Python module can automate pre-processing such as model creation and post-processing such as obtaining analysis results. On the other hand, Topology Optimization is a method for finding the optimal shape under constraints. Topology optimization requires iterative shape updates based on optimization methods and structural analysis such as finite element analysis. In this paper, we implemented a method to minimize weight under the constraint that stresses are below allowable stresses using Python module of Marc Mentat.

Key Words : Topology Optimization, Nonlinear finite element analysis

1. はじめに

製造業における設計業務では、事前検討の一環として強度計算が行われており、複雑な構造を対象とした強度評価には有限要素法解析が広く用いられている。一般に有限要素法による解析作業は以下の3つのフェーズに分かれる。

- (1)モデルの作成や解析条件を設定するプリ処理
- (2)設定した解析条件に基づく有限要素解析
- (3)解析結果を確認するポスト処理

同種の製品を対象とした解析を行う場合、モデルの作成や解析条件の設定、解析結果の評価などの手順が定型化される場合がある。そのため、商用の有限要素法解析ソフトウェアには業務効率化のために操作の自動化を行う機能が実装されている。汎用非線形有限要素法解析ソフトウェアMarcの専用GUIであるMentat[1]にはAPIとして専用のPythonモジュールが準備されており、このPythonモジュールを紹介することで解析モデルの作成・編集などのプリ処理や、解析結果の抽出などのポスト処理の自動化が可能である[2]。更にPythonの外部ライブラリと組み合わせることで、より高度な処理と組み合わせることも検討できる。例えば、Pandasを使用してExcelからパラメータを読み取り、そのパラメータに基づいて解析条件を自動設定することや、Matplotlibを用いて抽出した解析結果からグラフを描画すること、PyTorchと組み合わせることで計算結果から機械学習につなげることも検討できる。

一方、製品の設計においては、製品の使用状況の制約から生じる形状や配置などの領域に対する制約と、軽量化などの目的で課される重量に対する制約の範囲内で剛性を最大化できる形状を決定する場合がある。このような条件を満足する最適形状を探索する手法として、トポロジー最適化が注目されている。トポロジー最適化は有限要素法解析などの構造解析と、最適化手法に基づく形状

の更新を繰り返すことで最適形状を探索する手法である。例えば、商用の有限要素法解析ソフトウェアを使用し、計算結果の自動抽出と解析モデルの自動更新を行うシステムを構築すればトポロジー最適化の計算を行うことが可能である。

本論文ではMarc MentatのPython機能を介してトポロジー最適化の実装を試みた。2章でMarc MentatのPython機能の概要を示し、3章ではトポロジー最適化の理論の概要と実装した手法を示す。4章ではこれらを踏まえて実装したトポロジー最適化のフレームワークを示し、5章では実装したシステムで得られた最適形状の例を示す。

2. Marc MentatのPython機能の概要

Marc Mentatでは、インストール時にPythonの実行環境とMentatの自動操作や解析結果の抽出を行うための専用モジュールpy_mentat, py_postが自動構築される。このPython機能を活用することで以下のような処理を行うことが可能である。

(1) プリ・ポスト処理の自動化

MentatのGUI上での操作には、全てコマンドが割り振られている。例えば、原点に節点を作成する操作は以下のようなコマンドになる。

```
*add_nodes 0 0 0
```

このような操作コマンドは、Mentat用のPython関数である”py_send”を使用することで、Pythonコードから実行することが可能である。前述の節点を作成するコマンドを実行する場合は以下のようなPythonコードを記述する。

```
py_send(“*add_nodes 0 0 0”)
```

上記の関数を使用することで、例えば大量の節点座標データを保存したExcelファイルをPythonで読み込み、その座標値から自動で節点を生成する処理を実装することができる。このように、Mentatの操作をPythonから実行することで、モデル作成や解析条件の設定、解析結果の抽出など、業務の中で発生する繰り返し作業や定型作業を自動化することも可能である。

(2) Mentatの標準機能では困難なポスト処理の実装

専用モジュールを使用することで、解析結果のファイルから任意の物理量を抽出し、Pythonの変数として使用することが可能である。これを利用し、変数として格納した解析結果をPython側で処理し、整理して出力することや、次の解析で使用するパラメータにフィードバックすることが可能である。

例えば、インクリメント0における節点番号1の節点の解析結果の抽出には、以下のような記述を行う。

```
p = post_open("***filepath***")
p.moveto(0+1)
node_index = node_sequence(1)
x = p.node_scalar(node_index, 2)
```

ここでインクリメントはmovetoの引数として指定する。moveto(0)はモデル情報のみとなっているため、moveto(インクリメント+1)のようにインクリメントを指定する。また、解析結果を抽出するための関数node_scalarの1つ目の引数には、ユーザー定義の節点番号(node ID)ではなく、内部節点番号(node index)を指定する必要がある。node_sequenceでユーザー定義の節点番号を内部節点番号に変換して、node_scalarの1つ目の引数として指定する。node_scalarの2つ目の引数には、ポストファイルに出力されているポスト値のリストの中で、目的のポスト値が何番目のポスト値であるかを調べて記述する。

(3) 外部ライブラリと組み合わせた処理の実装

前述のMentat専用モジュールに加えて、Pandas, Numpy, Matplotlibなどの外部ライブラリを読み込み、これらを組み合わせた処理を実装することも可能である。例えば、整理した解析結果のデータからMatplotlibを使用して任意の形式のグラフを自動出力する機能を実装することも可能である(図1)。

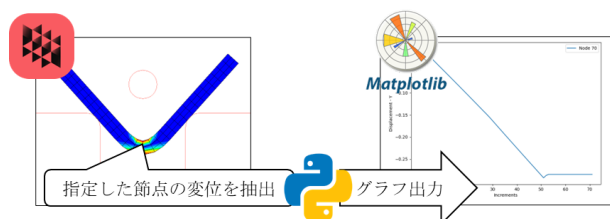


図-1 外部ライブラリMatplotlibによるグラフ出力例

3. トポロジー最適化の概要

(1) 材料分布の表現

トポロジー最適化は、図2に示すような固定設計領域内の材料分布の最適化問題を解く手法であり、材料分布の表現には均質化設計法や密度法、レベルセット法などが用いられる[3] [4]。

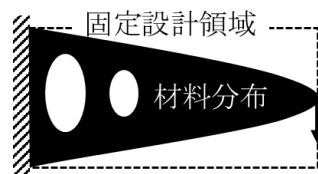


図-2 固定設計領域の導入

本論文では材料分布の表現として、既存の有限要素法解析ソフトウェアを使用した実装が容易な密度法を採用する。密度法は、固定設計領域内の各要素に0から1までの値をとる正規化された仮想的な無次元の材料密度を導入し、各要素の材料物性をこの材料密度の関数とする手法である。例えば、材料密度が図3に示すような分布の場合、 $\rho = 0$ の箇所には材料がなく、 $\rho = 1$ の箇所には材料が存在していることを意味する。

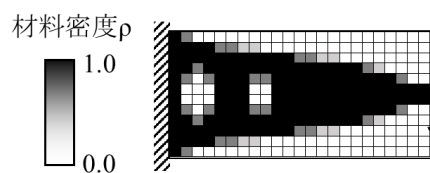


図-3 密度法の材料密度分布例

i 番目の要素における材料密度を ρ_i とし、 i 番目の要素のヤング率 E_i を以下のように表現する。

$$E_i(\rho_i) = (E_0 - E_{min})\rho_i^\lambda + E_{min} \quad (1)$$

ここで E_0 は材料固有のヤング率、 E_{min} はヤング率の下限値、 λ はペナルティパラメータである。

式(1)を用いて材料の有無を記述し、次節に示す最適化手法によって目的を満足する最適な形状の探索を行う。

(1) 代表的なトポロジー最適化

最適化(数理最適化)は、制約条件下で目的関数が最大または最小となる変数(設計変数)を求める手法である[5] [6]。トポロジー最適化では、設計変数を各要素の材料密度 ρ_i とし、重量 V を上限値 V_{allow} 以下に抑える制約条件 $g = V - V_{allow} \leq 0$ の下で、剛性の最大化を目的としてひずみエネルギーの最小化を目的関数 F とする最小化問題を解く場合がある。この問題を解く最適化手法には様々な手法があり、代表的な手法として式(2)で設計変数を更新する最適性基準法がある。

$$\rho_i^{(k+1)} = \rho_i^{(k)} \left(\frac{\frac{\partial F^{(k)}}{\partial \rho_i}}{-\Lambda^{(k)} \frac{\partial g^{(k)}}{\partial \rho_i}} \right)^\mu \quad (2)$$

ここで、 Λ はラグランジュ乗数、 μ は定数である。

しかしながら、強度設計への適用を考えた場合、前述の条件のみでは得られた形状の最大応力が許容応力以内であることは保証されておらず、直接適用することが困難である。また、設計目的によっては剛性の最大化ではなく、使用する材料の最小化や軽量化に重点を置くケースも多い。そこで本論文では前述の条件ではなく、次節に示す相当ミーゼス応力の最大値を許容応力 σ_{allow} 以下とする制約条件下で重量を最小化する手法を検討する。

(2) 実装した手法のコンセプト

最適化手法が使用されていないため厳密にはトポロジ最適化とは異なるが、前節の目的を満たす形状を求める最も単純な手法として全応力設計法がある。全応力設計法は構造物の各部が許容応力となるように設計変数を更新する手法であり、式(3)に示す式で設計変数 x を更新する。

$$x_i^{(k+1)} = x_i^{(k)} \left(\frac{\sigma_i}{\sigma_{allow}} \right)^\gamma \quad (3)$$

ここで、 γ は緩和係数、 σ_i は各要素 i の応力である。式(3)は許容応力以下の箇所の材料を減少させ、許容応力以上の箇所に材料を配置する処理を表す。しかしながら、設計変数に制約を課す密度法などの手法に適用する場合、全設計変数を許容応力とすることは難しく、このような条件に全応力設計法を適用することは困難である。

そこで本論文では全応力設計法と同様に、応力の低い箇所の材料密度を減少させる方法をとるが、変動可能な閾値 β を設け、これを基準に材料密度を増減させる方法を採用する。ここで導入した閾値 β を設計変数として最適化手法で更新し、材料密度分布を最適化する方法を検討した。また、最終的な材料密度の分布にグレースケールと呼ばれる $\rho_i = 0.5$ のような中間の材料密度が現れることを抑制する目的で、材料密度の更新には式(4)に示す閾値 β を基準に急激に材料密度を増減させる式を用いる。

$$\rho_i^{(k+1)} = \rho_i^{(k)} + \frac{2}{1 + \exp\left(-\alpha \left(\frac{\sigma_i}{\sigma_{allow}} - \beta\right)\right)} - 1 \quad (4)$$

上式の第2項と第3項の和は材料密度の更新量を表す。この材料密度の更新量と各要素の相当ミーゼス応力を許容応力で除した σ_i/σ_{allow} の関係を図4に示す。

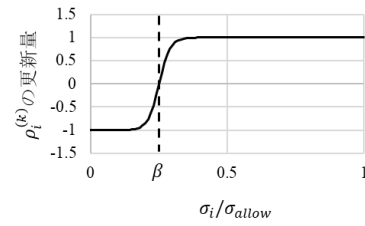


図-4 式(4)による材料密度の更新量(第2項と第3項の和)

図4に示すように、式(4)を用いた材料密度の更新量は β を境に増減が反転する。 α はこの反転する箇所における曲線の滑らかさを制御するパラメータである。

材料密度 ρ_i は式(4)を用いて更新を行うが、 ρ_i は0から1までの値とするため、式(5)による制約を設ける。

$$\rho_i^{(k+1)} = \begin{cases} 1 & \text{if } 1 < \rho_i^{(k+1)} \\ \rho_i^{(k+1)} & \text{if } 0 \leq \rho_i^{(k+1)} \leq 1 \\ 0 & \text{if } \rho_i^{(k+1)} < 0 \end{cases} \quad (5)$$

式(5)により $k+1$ 回目の材料密度 $\rho_i^{(k+1)}$ が決定されたら、式(1)に基づいて各要素のヤング率の更新を行う。

(3) 制約条件付き最適化

相当ミーゼス応力の最大値 σ_{max} を許容応力 σ_{allow} 以下とする制約条件下で、総重量 V を最小化することを目的とする制約条件付きの最適化には様々な手法がある。例えば、ペナルティ関数法は、式(6)に示すように制約条件 $g < 0$ を満たす場合は0を、満たさない場合は正の値をとるペナルティ関数 $G(g)$ を本来の目的関数 F_{old} に加算して、最終的な目的関数を F_{new} とし、制約条件なし最適化問題として解く手法である。

$$\text{minimize } F_{new} = \begin{cases} F_{old} & \text{if } g < 0 \\ F_{old} + P G(g) & \text{if } 0 \leq g \end{cases} \quad (6)$$

ここで、 P はペナルティ関数の影響度合いを制御するための重みである。

本論文では相当ミーゼス応力の最大値 σ_{max} を許容応力 σ_{allow} 以下とする制約条件下で総重量 V を最小化することを目的とし、目的関数 F を以下のように定義する。

$$\text{minimize } F = \begin{cases} \frac{V}{V_{ini}} & \text{if } \sigma_{max} \leq \sigma_{allow} \\ \frac{V}{V_{ini}} + P \left(\frac{\sigma_{max}}{\sigma_{allow}} \right)^2 & \text{if } \sigma_{max} > \sigma_{allow} \end{cases} \quad (7)$$

ここで V_{ini} は初期の重量を表し、総重量 V を0から1に正規化する。 $\sigma_{max} > \sigma_{allow}$ となった場合はペナルティ関数を考慮して目的関数を増加させ、制約条件を満足させる。

(4) 最適化手法

目的関数の最小値を求める最適化手法には様々な手法がある。例えば、最急降下法は設計変数 x の関数である目的関数 $F(x)$ の勾配がゼロとなる点を最小値とし、式(8)のように現在の設計変数 $x_i^{(k)}$ を目的関数の勾配 $\partial F(x)/\partial x$ とステップサイズ η の積だけ更新して、目的関数が最小となる設計変数を探索する手法である[5] [6]。

$$x_i^{(k+1)} = x_i^{(k)} - \eta \frac{\partial F(x)}{\partial x_i^{(k)}} \quad (8)$$

本論文では閾値 β を設計変数として、式(9)に示す最急降下法で式(7)に示す目的関数 F の最小化を行う。

$$\beta^{(k+1)} = \beta^{(k)} - \eta \frac{F^{(k)} - F^{(k-1)}}{\beta^{(k)} - \beta^{(k-1)}} \quad (9)$$

ここで式(8)における目的関数の設計変数微分は、後退差分法により k 回目の値と一つ前の $k-1$ 回目の値の差分から $(F^{(k)} - F^{(k-1)})/(\beta^{(k)} - \beta^{(k-1)})$ とする。この計算には $k-1$ 回目と k 回目と2回分の値が必要となる。そのため、初回（初期形状）の閾値 $\beta^{(0)}$ と次の閾値 $\beta^{(1)}$ は任意の値を設定して計算を行う。

(5) 収束判定

前述の設計変数の更新を行う繰り返し処理は収束判定基準を満足した時点で終了する。一般的には目的関数や設計変数の変化量の最大値を収束判定に使用する方法が用いられる。本論文では、以下に示す材料密度の変化量の絶対値の最大値が閾値 S 以下となった場合に収束と判定する方法を採用した。

$$\max \left(\left| \rho_i^{(k+1)} - \rho_i^{(k)} \right| \right) \leq S \quad (10)$$

ここまでの処理の流れをフローチャートとしたものを図5に示す。

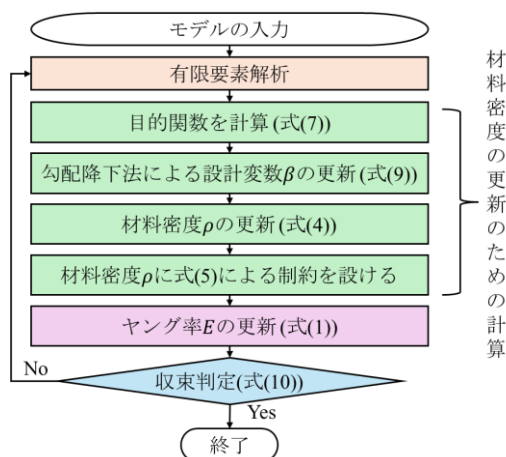


図-5 実装した手法の処理の流れ

4. MentatのPython機能を介してトポロジー最適化のフレームワークを実装する方法

本論文では3章で示した材料分布を密度法で表現し、許容応力以内で最軽量な材料分布を求める全応力設計法を参考にした手法をMentatのPython機能を介して実装した。全体的な処理の流れを図6に示す。本章では各処理ごとの具体的な実装方法について解説を行う。

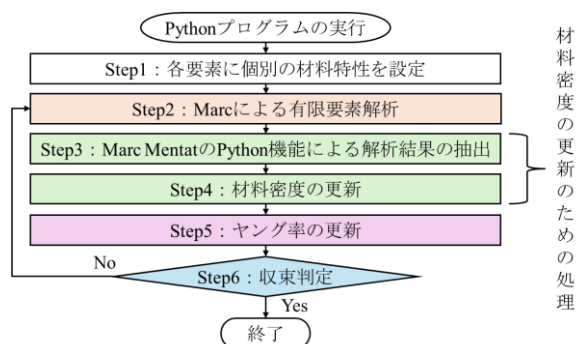


図-6 実装したPythonプログラムの流れ

(1) Step1:各要素に個別の材料特性を設定

密度法を用いた式(1)の操作は各要素のヤング率を制御することと等価である。Marcでこの処理を行う方法として、各要素にそれぞれ異なる材料特性を設定し、各材料特性中のヤング率の値を修正する方法が検討できる。Step1では前処理として固定設計領域内の各要素に対してそれぞれ異なる材料特性を定義する。この処理は膨大な繰り返し作業を要するため、Pythonで自動設定を行う処理を組み込んだ。

(2) Step2: Marcによる有限要素解析

Step1の処理の後、固定設計領域に材料が充填された状態 $(\rho_i = 1)$ を初期形状としてStep2でMarcによる解析を実行する。その後Step6の収束判定を満たすまでStep2~5の処理を繰り返す。

(3) Step3: Marc MentatのPython機能による解析結果の抽出

得られた有限要素法解析結果からMarc MentatのPython機能を使用して固定設計領域内の各要素の相当ミーゼス応力 σ_i を取得する。この値は以降の計算で使用するため、Pythonの変数として保持しておく。

(4) Step4: 材料密度の更新

Step3で取得した σ_i を基に図5に示すフローチャート中の式(7)、式(9)、式(4)、式(5)の計算を順次行い、各要素の材料密度 ρ_i を更新する。

(5) Step5: ヤング率の更新

Step4で求めた材料密度 ρ_i を用いて式(1)の計算を行い、固定設計領域内の各要素のヤング率を更新する。

(6) Step6 : 収束判定

Step2からStep5までの繰り返し処理は式(10)の収束判定基準を満足した場合に終了させ、最終的な材料密度分布を出力する。

実装した手法を用いて簡単な検証計算を行った例を次章で示す。

5. 数値計算例

(1) 解析条件

図7に示すモデルに対して、実装した手法により最適化を行った。

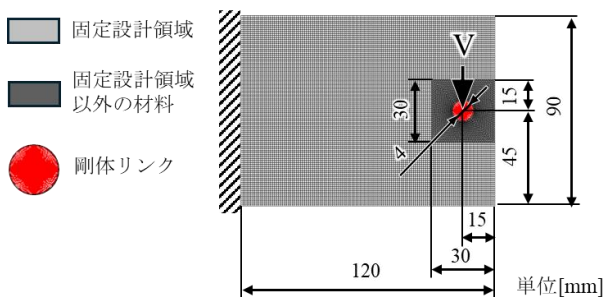


図-7 解析条件

ここでは、平面応力問題を仮定し、四辺形一次の低減積分要素（要素数：13312）を用いて解析を行った。また、ヤング率 $E_0 = 200,000[\text{MPa}]$ 、ポアソン比 $0.3[-]$ 、厚さ $1[\text{mm}]$ 、荷重 $V = 500[\text{N}]$ とし、最適化のパラメータは、許容応力 $\sigma_{allow} = 100$ 、 $E_{min} = 0.0001$ 、 $\lambda = 1$ 、 $S = 0.01$ 、 $P = 0.5$ 、 $\eta = 0.001$ 、 $\alpha = 100$ 、 $\beta^{(0)} = 0$ 、 $\beta^{(1)} = 0.1$ とした。

(2) 数値解析結果

構築したシステムを用い、前節に示す条件の下で最適化計算を行った。得られた材料密度分布を図8に示す。

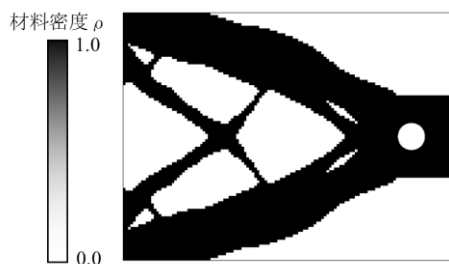


図-8 数値計算結果（材料密度分布）

図8の材料密度は、初期の材料密度分布である固定設計領域に比べて重量が56.45%減少した。

また、図8の材料密度分布における相当ミーゼス応力の分布を図9に示す。最大値は105.4[MPa]となっており、許容応力100[MPa]よりも5[MPa]ほど大きい値となっているが、概ね制約条件を満たしていることが確認できた。

制約条件を完全に満足しなかった原因としては、ペナルティ関数法により、制約条件無し最適化として計算したため完全には制約条件を満足できなかったと考えられる。これは目的関数へのペナルティ関数の影響度合いを制御するパラメータPを大きい値に調整することで改善されると推測される。

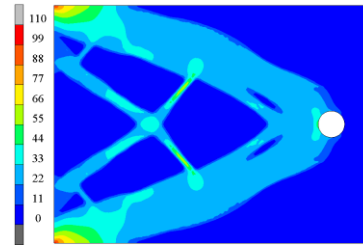


図-9 数値計算結果（相当ミーゼス応力分布）

6. 結言

本論文では相当ミーゼス応力の最大値を許容応力以下とする制約条件下で重量を最小化することを目的として、全応力設計法を参考にした手法をMarc MentatのPython機能で実装した。この検討結果から、トポロジー最適化のような複雑な繰り返し計算を伴う処理をMarc MentatのPython機能で実行可能であることが示された。

5章の数値計算では、目的関数である重量を削減した最適形状を得ることができた。一方で、今回実装した手法は材料密度を設計変数としていないため、一般的なトポロジー最適化手法とは異なる。そのため、得られた解が最適解である保証が困難である。今後は最適性基準法などの一般的なトポロジー最適化手法の実装を検討する。最適性基準法の実装は4章4節の材料密度の更新の処理を変更するのみであるため実装は容易である。

また、5章の数値計算結果は制約条件を完全には満足していなかったため、ペナルティ関数法のパラメータPなどの調整を検討する必要がある。その他、実用性を考えて製造方法を考慮した手法の検討も試みる。

参考文献

- [1] Hexagon : “Marc非線形有限要素法解析ソフトウェア”. <https://hexagon.com/ja/products/marc>, (参照2025-4-3)
- [2] Hexagon : Marc 2024.2 Python: Tutorial and Reference Manual, 2024.
- [3] 西脇 眞二, 泉井 一浩, 菊池 昇 : 計算力学レクチャーコース トポロジー最適化, 丸善出版, 2013.
- [4] 近藤 継男, 矢地 謙太郎, 西脇 眞二 : トポロジー最適化の基礎 弾性体ならびに熱流体関連工学諸問題への応用のために, コロナ社, 2024.
- [5] 梅谷 俊治 : しっかり学ぶ数値最適化 モデルからアルゴリズムまで, 講談社, 2020.
- [6] 斎藤 康毅 : ゼロから作るDeep Learning Pythonで学ぶディープラーニングの理論と実装, オーム社, 2022.