

最適化手法CONLINを用いた 3 次元構造物の位相最適化

Topology Optimization of Three-Dimensional Structures Using CONLIN Optimizer

藤井大地*, 鈴木克幸**, 大坪英臣***

Daiji FUJII, Katsuyuki SUZUKI and Hideomi OHTSUBO

*博士(工学) 東京大学助手 工学系研究科 (〒113-8656 東京都文京区本郷7-3-1, dfujii@nasl.t.u-tokyo.ac.jp)**正会員 Ph.D. 東京大学助教授 新領域創成科学研究科 (〒113-8656 東京都文京区本郷7-3-1, katsu@k.u-tokyo.ac.jp)***工博 東京大学教授 工学系研究科 (〒113-8656 東京都文京区本郷7-3-1, ohtsubo@nasl.t.u-tokyo.ac.jp)

The Convex Linearization method (CONLIN), that is a dual optimizer based on the convex approximation concepts, is often used to solve the structural optimization problem, recently. The CONLIN optimizer has many benefits, for example, the handling of constraint conditions is easy as well as the SLP method, and the performance of convergence is excellent as well as the optimality criteria method (OC). In this paper, the CONLIN optimizer is applied to the topology optimization problem of three-dimensional structures. The density approach is used to obtain the optimum topology, and the filtering method proposed by the author is used to obtain the clearer topology. In the optimization problem, the total volume and the value of the gravity function which is used for the filtering are constrained. The validity of the CONLIN optimizer are clarified by comparing with the OC method. Some examples of bridge design are shown to demonstrate the effectiveness of the CONLIN optimizer for the topology optimization analysis.

Key Words : CONLIN, Dual method, Topology optimization, Density approach, Bridge structure

1. はじめに

最適化問題の解法の一つである凸線形化法 (Convex Linearization method, CONLIN) は, 1986 年に Fleury and Braibant¹⁾によって提案された方法で, 局所的に凸関数条件を満足させるために, 目的関数および制約条件を, 既知の設計変数の近傍で, 感度係数が正であれば設計変数で, 感度係数が負であれば設計変数の逆数でテーラー展開して解く方法である。また, この方法では, 凸関数条件を利用して, 制約条件にラグランジェ乗数を掛けて目的関数に加えるラグランジェ関数を定義し, ラグランジェ乗数を設計変数とする双対問題を解くことにより計算効率を改善している。その後, Fleury²⁾は, この方法をさらに発展させ, よりロバスト性の高いものになっている。また, この方法は, Beekers^{3),4)}によって, グランドストラクチャー法によるトラス構造の位相最適化問題や, 離散最適化問題として定式化された連続体の位相最適化問題に適用されている。

CONLIN は, OC 法 (Optimality Criteria method, 最適性規準法) と同様にラグランジェ関数の最適化を行うものであるが, OC 法においては複数制約条件の扱いが煩雑であるのに対して, CONLIN では複数制約条件を容易に扱うことができる。また, 設計変数が増えた場合の収束性も, OC 法同様優れている。

一方, 連続体の位相最適化問題では, 最適位相に現れる

チェッカーボード (要素の密度分布がチェッカーボードのようになる現象) やグレースケール (中間的な密度分布が広がる現象) がしばしば問題となる⁵⁾。したがって, 最近では, 位相最適化手法とチェッカーボードやグレースケールを防ぐフィルタリング法を組み合わせる方法が一般的となっている。しかしながら, 従来の位相最適化手法においてよく用いられている OC 法では, 複数の制約条件を扱うことが難しく, フィルタリングに必要な条件を制約条件として加える場合, 特別な工夫が必要となる⁶⁾。特に, チェッカーボードが頻繁に現れる密度法では, チェッカーボードを消去するフィルタリング法の制約条件の制御が難しい。

そこで, 本論文では, 密度法とフィルタリング法を組み合わせた位相最適化手法に, 複数制約条件を容易に扱える CONLIN を適用し, その有効性を検討する。本研究では, CONLIN のプログラムは, 文献 2)を参考に独自に開発し, すでに開発している密度法による 3 次元連続体の位相最適化プログラム⁷⁾に実装する。

なお, 同様の最適化手法として, CONLIN の考え方を発展させ, より収束安定性を高めた MMA (Moving asymptotes method)^{8),9)}等の手法も提案されているが, パラメータの扱いがより煩雑になることと, ここで対象とする位相最適化問題では, CONLIN 手法で十分良好な収束性が得られたため, 本論文ではこれらの手法の検討は行っていない。

以下、本論文第2章では、文献2)を参考に、CONLINの理論を示し、第3章では、密度法による3次元連続体の位相最適化手法およびCONLINの適用法を示す。第4章では、アーチ橋の位相最適化問題を例題として、OC法との比較により、本方法の有効性を検討する。第5章では、以上で得られた知見を述べてまとめとする。

2. CONLINによる最適化問題の解法

ここでは、文献2)を参考に、著者らがプログラムの開発に実際に用いた定式化を示している。また、2.3節に示す解法は、増分を用いている点で文献2)とは若干異なる。

2.1 CONLINによる最適化問題の定式化

一般的な最適化問題を次式で表す。

$$\begin{aligned} \min \quad & c_0(\mathbf{x}) \\ \text{subject to} \quad & c_j(\mathbf{x}) \leq 0 \quad (j=1, \dots, m) \\ & \underline{x}_i \leq x_i \leq \bar{x}_i \quad (i=1, \dots, n) \end{aligned} \quad (1)$$

ここに、 $\mathbf{x} = \{x_1, \dots, x_n\}$ は設計変数、 $c_0(\mathbf{x})$ は目的関数、 $c_j(\mathbf{x})$ ($j \geq 1$) は制約条件、 $\underline{x}_i, \bar{x}_i$ は設計変数 x_i の下限値と上限値を表す。ただし、ここでは、 $\underline{x}_i \geq 0$ とする。

CONLIN では、(1)式の目的関数および制約条件を次式のように設計変数 $\mathbf{x}^t$ 点に関して Taylor 展開する。

$$\begin{aligned} c_j(\mathbf{x}) = c_j(\mathbf{x}^t) &+ \sum_{\frac{\partial c_j(\mathbf{x}^t)}{\partial x_i} > 0} \frac{\partial c_j(\mathbf{x}^t)}{\partial x_i} (x_i - x_i^t) \\ &- \sum_{\frac{\partial c_j(\mathbf{x}^t)}{\partial x_i} < 0} (x_i^t)^2 \frac{\partial c_j(\mathbf{x}^t)}{\partial x_i} \left(\frac{1}{x_i} - \frac{1}{x_i^t} \right) \end{aligned} \quad (2)$$

ここに、 $j=0, 1, \dots, m$ であり、右辺第2項は x_i に関する Taylor 展開の1次項、第3項は $1/x_i$ に関する Taylor 展開の1次項である。

さらに、数値計算誤差を小さくするために、(2)式の設計変数に次式のようなスケーリングをかける。

$$\bar{x}_i = \frac{x_i}{x_i^t} \Rightarrow \frac{\partial c_j(\mathbf{x}^t)}{\partial x_i} = \frac{\partial c_j(\mathbf{x}^t)}{\partial \bar{x}_i} \frac{1}{x_i^t} \quad (3)$$

(2),(3)式を用いて(1)式を書き直すと、

$$\begin{aligned} \min \quad & \sum_{\frac{\partial c_0(\mathbf{x}^t)}{\partial \bar{x}_i} > 0} \frac{\partial c_0(\mathbf{x}^t)}{\partial \bar{x}_i} \bar{x}_i - \sum_{\frac{\partial c_0(\mathbf{x}^t)}{\partial \bar{x}_i} < 0} \frac{\partial c_0(\mathbf{x}^t)}{\partial \bar{x}_i} \frac{1}{\bar{x}_i} - \bar{c}_0 \\ \text{subject to} \quad & \sum_{\frac{\partial c_j(\mathbf{x}^t)}{\partial \bar{x}_i} > 0} \frac{\partial c_j(\mathbf{x}^t)}{\partial \bar{x}_i} \bar{x}_i - \sum_{\frac{\partial c_j(\mathbf{x}^t)}{\partial \bar{x}_i} < 0} \frac{\partial c_j(\mathbf{x}^t)}{\partial \bar{x}_i} \frac{1}{\bar{x}_i} \leq \bar{c}_j \\ & \frac{\underline{x}_i}{x_i^t} \leq \bar{x}_i \leq \frac{\bar{x}_i}{x_i^t} \end{aligned} \quad (4)$$

ただし、

$$\bar{c}_j = \sum_i \left| \frac{\partial c_j(\mathbf{x}^t)}{\partial x_i} \right| x_i^t - c_j(\mathbf{x}^t) \quad (j=0, \dots, m) \quad (5)$$

2.2 双対法 (Dual method) の適用

(4)式を簡単のため次式のように表しておく。

$$\begin{aligned} \min \quad & \sum_+ \frac{\partial c_0}{\partial x_i} x_i - \sum_- \frac{\partial c_0}{\partial x_i} \frac{1}{x_i} - \bar{c}_0 \\ \text{subject to} \quad & \sum_+ \frac{\partial c_j}{\partial x_i} x_i - \sum_- \frac{\partial c_j}{\partial x_i} \frac{1}{x_i} - \bar{c}_j \leq 0 \\ & \underline{x}_i \leq x_i \leq \bar{x}_i \end{aligned} \quad (6)$$

(6)式はラグランジュ双対定理¹⁰⁾により、次のような問題に置き換えることができる。

$$\begin{aligned} \max \quad & l(\mathbf{r}) \\ \text{subject to} \quad & r_j \geq 0 \quad (j=0, \dots, m) \end{aligned} \quad (7)$$

ここに、

$$l(\mathbf{r}) = \min_{\underline{x}_i \leq x_i \leq \bar{x}_i} L(\mathbf{x}, \mathbf{r}) \quad (8)$$

ただし、

$$L(\mathbf{x}, \mathbf{r}) = \sum_{j=0}^m r_j \left(\sum_+ \frac{\partial c_j}{\partial x_i} x_i - \sum_- \frac{\partial c_j}{\partial x_i} \frac{1}{x_i} - \bar{c}_j \right) \quad (9)$$

ここで、 r_j はラグランジュ乗数で、目的関数にはラグランジュ乗数はかからないので、 $r_0=1$ である。したがって、(6)式は、(7)式と(8)式の2つ問題として解くことができる。

まず、(8)式は、変数 x_i に関する最適化問題であるが、これは n 個の個別の問題として解くことができる。すなわち、(8)式は次のように書ける。

$$l(\mathbf{r}) = \min_{\underline{x}_i \leq x_i \leq \bar{x}_i} \left[\sum_{i=1}^n L_i(x_i, \mathbf{r}) - \bar{c}'(\mathbf{r}) \right] \quad (10)$$

ここに、

$$\begin{aligned} L_i(x_i, \mathbf{r}) &= \sum_{j=0}^m r_j \left(\frac{\partial c_j}{\partial x_i} \right)_+ x_i - \sum_{j=0}^m r_j \left(\frac{\partial c_j}{\partial x_i} \right)_- \frac{1}{x_i} \\ \bar{c}'(\mathbf{r}) &= \sum_{j=0}^m r_j \bar{c}_j \end{aligned} \quad (11)$$

ただし、 $(\cdot)_+, (\cdot)_-$ は、括弧内が+または-の場合のみ採用されることを意味する。したがって、(10)式は次のような最適化問題となる。

$$\begin{aligned} \min \quad & L_i(x_i) = a_i x_i + \frac{b_i}{x_i} \\ \text{subject to} \quad & \underline{x}_i \leq x_i \leq \bar{x}_i \quad (i=1, \dots, n) \end{aligned} \quad (12)$$

ここに、

$$a_i = \sum_{j=0}^m r_j \left(\frac{\partial c_j}{\partial x_i} \right)_+ \geq 0, \quad b_i = - \sum_{j=0}^m r_j \left(\frac{\partial c_j}{\partial x_i} \right)_- \geq 0 \quad (13)$$

したがって、(12)式の解は次式から求まる。

$$\frac{\partial L_i(x_i)}{\partial x_i} = a_i - \frac{b_i}{x_i^2} = 0 \quad (14)$$

すなわち,

$$x_i = \left(\frac{b_i}{a_i} \right)^{\frac{1}{2}} \quad \text{if } x_i^2 \leq \frac{b_i}{a_i} \leq \bar{x}_i^2 \quad (15)$$

$$x_i = x_i \quad \text{if } \frac{b_i}{a_i} \leq x_i^2 \quad (16)$$

$$x_i = \bar{x}_i \quad \text{if } \bar{x}_i^2 \leq \frac{b_i}{a_i} \quad (17)$$

(15)~(17)式によって, x_i は $\mathbf{r}$ の関数として表せたことになる。したがって, (15)~(17)式の解を用いて, (7)式は次式のように表される。

$$\begin{aligned} \max \quad & l(\mathbf{r}) = \sum_{j=0}^m r_j \left(\sum_{+} \frac{\partial c_j}{\partial x_i} x_i(\mathbf{r}) - \sum_{-} \frac{\partial c_j}{\partial x_i} \frac{1}{x_i(\mathbf{r})} - \bar{c}_j \right) \\ \text{subject to} \quad & r_j \geq 0 \quad (j=0, \dots, m) \end{aligned} \quad (18)$$

(18)式の問題を逐次2次計画法で解く。逐次2次計画法では, 設計変数に関する1階微分と2階微分が必要となるが, まず, (18)式の目的関数 l の r_j に関する1階微分は次式となる。

$$g_j = \frac{dl(\mathbf{r})}{dr_j} = \sum_{+} \frac{\partial c_j}{\partial x_i} x_i - \sum_{-} \frac{\partial c_j}{\partial x_i} \frac{1}{x_i} - \bar{c}_j \quad (19)$$

ただし $j \geq 1$ 。また, 2階微分は次式となる。

$$H_{jk} = \frac{d^2 l}{dr_j dr_k} = \frac{dg_j}{dr_k} = \sum_{+} \frac{\partial c_j}{\partial x_i} \frac{dx_i}{dr_k} + \sum_{-} \frac{\partial c_j}{\partial x_i} \frac{1}{x_i^2} \frac{dx_i}{dr_k} \quad (20)$$

ただし $k \geq 1$ 。ここで, (15)~(17)式により,

$$\frac{dx_i}{dr_k} = \frac{\frac{db_i}{dr_k} - x_i^2 \frac{da_i}{dr_k}}{2x_i a_i} \quad \text{if } x_i^2 < \frac{b_i}{a_i} < \bar{x}_i^2 \quad (21)$$

$$\frac{dx_i}{dr_k} = 0 \quad \text{if } \frac{b_i}{a_i} \leq x_i^2 \quad (22)$$

$$\frac{dx_i}{dr_k} = 0 \quad \text{if } \bar{x}_i^2 \leq \frac{b_i}{a_i} \quad (23)$$

また, (13)式より,

$$\frac{da_i}{dr_k} = \left(\frac{\partial c_k}{\partial x_i} \right)_{+}, \quad \frac{db_i}{dr_k} = - \left(\frac{\partial c_k}{\partial x_i} \right)_{-} \quad (24)$$

であるから, (21)式は,

$$\frac{dx_i}{dr_k} = - \frac{x_i}{2a_i} \frac{\partial c_k}{\partial x_i} \quad \text{if } \frac{\partial c_k}{\partial x_i} > 0 \quad (25)$$

$$\frac{dx_i}{dr_k} = - \frac{1}{2x_i a_i} \frac{\partial c_k}{\partial x_i} \quad \text{if } \frac{\partial c_k}{\partial x_i} < 0 \quad (26)$$

(25),(26)式を(20)式に代入すると, $x_i^2 < b_i/a_i < \bar{x}_i^2$ の範囲では, 最終的に次式が得られる。

$$H_{jk} = - \frac{1}{2} \left(\sum_{+} \frac{\partial c_j}{\partial x_i} \frac{\partial c_k}{\partial x_i} \frac{x_i}{a_i} + \sum_{-} \frac{\partial c_j}{\partial x_i} \frac{\partial c_k}{\partial x_i} \frac{1}{x_i a_i} \right) \quad \text{if } \frac{\partial c_k}{\partial x_i} > 0$$

$$H_{jk} = - \frac{1}{2} \left(\sum_{+} \frac{\partial c_j}{\partial x_i} \frac{\partial c_k}{\partial x_i} \frac{1}{x_i a_i} + \sum_{-} \frac{\partial c_j}{\partial x_i} \frac{\partial c_k}{\partial x_i} \frac{1}{x_i^3 a_i} \right) \quad \text{if } \frac{\partial c_k}{\partial x_i} < 0$$

(27),(28)

2.3 逐次2次計画法による解法

(18)式を逐次2次計画法で解くことを考える。まず, 設計変数 r_j の更新式を次式のように置く。

$$r_j^{k+1} = r_j^k + \Delta r_j \quad (j=1, \dots, m) \quad (29)$$

(18)式の $l(\mathbf{r})$ を r_j^k に関してテーラー展開し2次項まで採用すると,

$$l(\mathbf{r}^{k+1}) = l(\mathbf{r}^k) + \sum_{j=1}^m g_j \Delta r_j + \frac{1}{2} \sum_{j=1}^m \sum_{k=1}^m H_{jk} \Delta r_j \Delta r_k \quad (30)$$

(30)式をベクトル・マトリックス表示すると,

$$l(\mathbf{r}^{k+1}) = l(\mathbf{r}^k) + \Delta \mathbf{r}^T \mathbf{g} + \frac{1}{2} \Delta \mathbf{r}^T \mathbf{H} \Delta \mathbf{r} \quad (31)$$

したがって, (18)式は次式の問題を逐次解くことになる。

$$\begin{aligned} \max \quad & q(\Delta \mathbf{r}) = \Delta \mathbf{r}^T \mathbf{g} + \frac{1}{2} \Delta \mathbf{r}^T \mathbf{H} \Delta \mathbf{r} \\ \text{subject to} \quad & r_j^k + \Delta r_j \geq 0 \quad (j=1, \dots, m) \end{aligned} \quad (32)$$

(27),(28)式より, H_{jj} は常に負になるため, (32)式の極大値では次式が満足される。

$$\frac{dq(\Delta \mathbf{r})}{d\Delta \mathbf{r}} = \mathbf{g} + \mathbf{H} \Delta \mathbf{r} = 0 \quad (33)$$

したがって, 次式の連立方程式を共役勾配法等 ($\mathbf{H}$ が特異行列になっても解ける方法) で解くことによって $\Delta \mathbf{r}$ が求められる。

$$\mathbf{H} \Delta \mathbf{r} = -\mathbf{g} \quad (34)$$

得られた $\Delta \mathbf{r}$ より, 次ステップの $\mathbf{r}^{k+1} (= \mathbf{r}^k + \Delta \mathbf{r})$ を求め, $j=1, \dots, m$ で次式が満足されるまで更新を繰り返す。

$$\begin{aligned} g_j &= 0 & \text{if } r_j > 0 \\ g_j &< 0 & \text{if } r_j = 0 \end{aligned} \quad (35)$$

ただし, この繰り返しでは, 感度係数等の更新は行わない。(35)式が満足されたら, 感度係数等を更新して再度繰り返しを行い, 最終的な収束解が得られるまで以上の演算を繰り返す。

2.4 制約条件の緩和

制約条件を満足する許容解が見つけれない現象を防ぐために, 文献2)にしたがって, 次式により制約条件の緩和を行う。

$$\begin{aligned} \min \quad & \sum_{+} \frac{\partial c_0}{\partial x_i} x_i - \sum_{-} \frac{\partial c_0}{\partial x_i} \frac{1}{x_i} - \bar{c}_0 + z_0 w \delta \\ \text{subject to} \quad & \sum_{+} \frac{\partial c_j}{\partial x_i} x_i - \sum_{-} \frac{\partial c_j}{\partial x_i} \frac{1}{x_i} \leq \bar{c}_j + z_j \left(1 - \frac{1}{\delta} \right) \quad (j=1, \dots, m) \\ & x_i \leq \bar{x}_i \leq \bar{x}_i \quad (i=1, \dots, n), \quad 1 \leq \delta \end{aligned} \quad (36)$$

ここに, w はユーザーによって与えられる重みで, 必要に応じて1~10程度の値として与える。また,

$$z_j = \sum_i \left| \frac{\partial c_j(\mathbf{x}^k)}{\partial x_i} \right| x_i^k \quad (j=0, \dots, m) \quad (37)$$

である。ただし、(37)式の x_i は(5)式と同様にスケーリングをかける前の変数である。

(36)式のラグランジアンで、 δ に関する項を抜き出すと次式となる。

$$\min_{\delta \geq 1} z_0 w \delta - (1 - 1/\delta) \sum_{j=1}^m r_j z_j \quad (38)$$

上式の解は次式のようになる。

$$\delta = \left(\sum_{j=1}^m r_j z_j / w z_0 \right)^{1/2} \quad \text{if } \sum_{j=1}^m r_j z_j > w z_0$$

$$\delta = 1 \quad \text{if } \sum_{j=1}^m r_j z_j < w z_0 \quad (39)$$

また、(19)式の g_j は、

$$g_j = \sum_+ \frac{\partial c_j}{\partial x_i} x_i - \sum_- \frac{\partial c_j}{\partial x_i} \frac{1}{x_i} - \bar{c}_j - z_j \left(1 - \frac{1}{\delta} \right) \quad (40)$$

となり、また、(20)式より H_{jk} は次式から計算される。

$$H_{jk} = \frac{dg_j}{dr_k} = \sum_+ \frac{\partial c_j}{\partial x_i} \frac{dx_i}{dr_k} + \sum_- \frac{\partial c_j}{\partial x_i} \frac{1}{x_i^2} \frac{dx_i}{dr_k} - \frac{z_j}{\delta^2} \frac{d\delta}{dr_k} \quad (41)$$

ここで、

$$\frac{d\delta}{dr_k} = \frac{1}{2\delta} \frac{z_k}{w z_0} \quad \text{if } \sum_{j=1}^m r_j z_j > w z_0$$

$$\frac{d\delta}{dr_k} = 0 \quad \text{if } \sum_{j=1}^m r_j z_j < w z_0 \quad (42)$$

よって、 H_{jk} の δ に関する寄与項は次式から計算される。

$$-\frac{1}{2} \frac{z_j z_k}{w z_0 \delta^3} \quad \text{if } \sum_{j=1}^m r_j z_j > w z_0$$

$$0 \quad \text{if } \sum_{j=1}^m r_j z_j < w z_0 \quad (43)$$

したがって、(27),(28)式に上式を加えれば、制約条件を緩和した場合のヘッセ行列が求められる。

3. 3次元連続体の位相最適化

3.1 最適化問題の定式化

連続体の位相最適化手法の一つである密度法は、材料の弾性剛性が密度のべき乗に比例すると仮定し、各有限要素の密度を設計変数として最適位相を求める方法である(図-1参照)。

なお、著者らが開発している密度法による3次元連続体の位相最適化手法では、有限要素として同形状の直方体要素(Voxel)を用い、連立方程式の解法として、Element-by-element 手法にもとづく共役勾配法を用いている⁷⁾。また、直方体要素としては、曲げ変形に関しても精度が良好となる応力仮定法要素¹¹⁾を採用している。

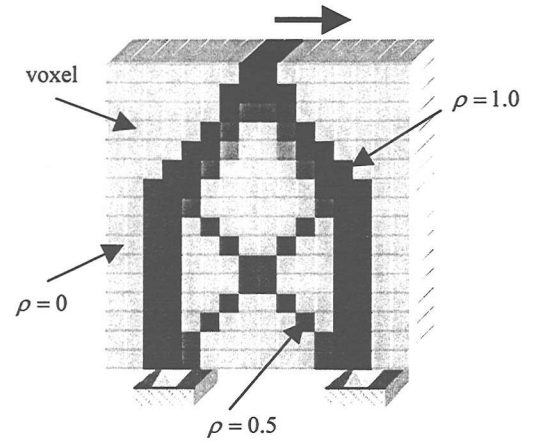


図-1 密度法を用いた位相最適化 (ρ: 密度比)

密度法による構造物の位相最適化問題は次式のように定式化される。

$$\min_{\rho \in L} [C(\rho)], \quad C(\rho) = \mathbf{U}^T \mathbf{K} \mathbf{U}$$

$$L = \left\{ \rho \mid 0 \leq \rho_i \leq 1 \quad (i=1, \dots, N), \quad \sum_{i=1}^N \rho_i \leq m_s \right\} \quad (44)$$

ここに、 ρ は設計変数 $\{\rho_1, \rho_2, \dots, \rho_N\}$ で、 ρ_i は i 番目要素の密度比を表し、 N は有限要素数である。また、(44)式の C は平均コンプライアンス、 $\mathbf{U}$ は節点変位ベクトル、 $\mathbf{K}$ は全体剛性マトリックス、 L は与えられた条件を満たす許容設計変数の集合、 m_s は密度比の制約値である。ここで、全体剛性マトリックス $\mathbf{K}$ は、要素密度比の関数として次式のように表す。

$$\mathbf{K}(\rho) = \sum_{i=1}^N \left(\rho_i^p \mathbf{K}_i^e \right) \quad (45)$$

ここに、 $\mathbf{A}$ は要素剛性マトリックスの重ね合わせを表す記号、 $\mathbf{K}_i^e$ は i 番目要素の要素剛性マトリックスである。また、 p はべき乗係数で、本論文では、文献12)およびいくつかの数値実験結果から $p=2$ に設定している。

3.2 フィルタリング法

密度法によるトポロジー最適化では、最適解の各所に要素密度比が交互に0と1となるチェッカーボード状の位相が現れる。これは、低次要素では節点を介してのみ力が伝達されるため、チェッカーボード状の密度分布の剛性が、数値計算上高くなってしまいうためである。なお、本研究で採用している応力仮定法要素は、応力が1次関数で仮定されるため、通常の8節点アイソパラメトリック要素に比較して精度は向上するが、変位は線形で仮定されており、要素辺上での高次変形を表せないため、チェッカーボードを防ぐことはできない。

このようなチェッカーボードを抑えるフィルタリング法として、最適位相の境界線長さを制約する Perimeter control 法⁵⁾があるが、この方法では、チェッカーボードの代わりに0と1の中間密度比の分布(グレースケール)が

広がるため、さらに別のフィルタリング法と組み合わせる必要がある¹³⁾。これに対して、著者らが提案している重力制御法^{7,14)}は、チェッカーボードとグレースケールの両方を同時に抑えることができる。

重力制御法では、次式の重力制御関数を定義する。

$$G(\rho) = \sum_{i=1}^N g_i(\rho) / \sum_{i=1}^N m_i \quad (46)$$

ここに、

$$g_i = \sum_{j=1}^{m_i} [\rho_i \rho_j + (1 - \rho_i)(1 - \rho_j)] \quad (47)$$

ただし、 m_i は i 番目の要素と辺を共有する要素数で、6 面体要素では、内部の要素では 6、端や隅の要素ではそれ以下となる。また、(46)式の G は $0 < G \leq 1$ となる。

図-2 は、2 次元問題で、密度比が 1, 0.5, 0 となる場合について、関数 g_i の値を示したものである。図からわかるように、この関数値は、 i 番目要素が黒で周辺要素が白の場合、または i 番目要素が白で周辺要素が黒の場合が最も低い値となる。チェッカーボード状の解は、このパターンに相当する。また、グレースケール（密度比が 0.5 の場合）が含まれる場合にもペナルティーが課され、例えば、すべてグレーの値は、すべて黒またはすべて白よりも低い値となる。したがって、(46)式の関数値が高くなれば、チェッカーボード状の密度分布が少なく、しかも白黒のはっきりした位相が得られる。

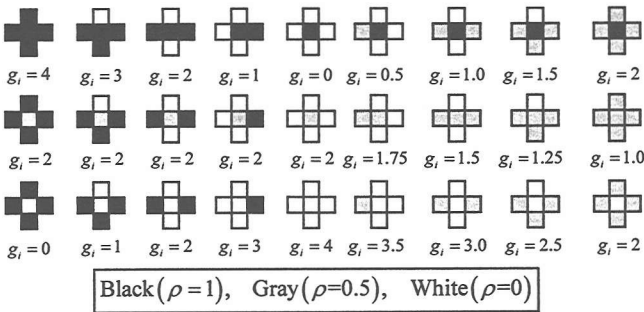


図-2 i 番目要素における関数 g_i の値 ($\rho_i, \rho_j : 0, 0.5, 1$)

そこで、ここでは、(46)式の G を与えられた値以上にする制約を課すことにする。この場合、(44)式は次のように書ける。

$$\begin{aligned} \min_{\alpha \in L} [C(\alpha)], \quad C(\alpha) &= \mathbf{U}^T \mathbf{K} \mathbf{U} \\ L &= \left\{ \alpha \mid 0 \leq \alpha_i \leq 1 \quad (i=1, \dots, N), \sum_{i=1}^N (1 - \alpha_i) \leq m_s, G(\alpha) \geq \bar{G} \right\} \end{aligned} \quad (48)$$

ただし、ここでは設計変数を ρ から $\alpha (= 1 - \rho)$ に変更している。この場合 $\rho_i = (1 - \alpha_i)$ となる。これは、比較に用いる最適性規準法の場合、このように設定する方がフィルタリングの効率が良いためである⁶⁾。また、(48)式の $\bar{G}$ は重力制御関数の制約値で、 $0 \leq \bar{G} \leq 1$ の範囲で与える。

3.3 CONLIN による解法

(48)式を CONLIN によって解く場合、次式のように書ける。

$$\begin{aligned} \min \quad & \sum_{+} \frac{\partial c_0}{\partial x_i} x_i - \sum_{-} \frac{\partial c_0}{\partial x_i} \frac{1}{x_i} \\ \text{subject to} \quad & \sum_{+} \frac{\partial c_j}{\partial x_i} x_i - \sum_{-} \frac{\partial c_j}{\partial x_i} \frac{1}{x_i} - \bar{c}_j \leq 0 \quad (j=1, 2) \\ & x_i \leq x_i \leq \bar{x}_i \quad (i=1, \dots, N) \end{aligned} \quad (49)$$

ここに、

$$\begin{aligned} x_i &= \frac{\alpha_i}{\alpha_i^k}, \quad x_i = 0, \quad \bar{x}_i = 1, \quad c_0 = C, \\ \frac{\partial c_0}{\partial x_i} &= \frac{\partial C}{\partial \alpha_i} \alpha_i^k = -\alpha_i^k \mathbf{U}^T \frac{\partial \mathbf{K}}{\partial \alpha_i} \mathbf{U} = -\alpha_i^k \mathbf{u}_i^T \frac{\partial \mathbf{k}_i}{\partial \alpha_i} \mathbf{u}_i \\ c_1 &= \sum_{i=1}^N (1 - \alpha_i), \quad \bar{c}_1 = \sum_{i=1}^N \alpha_i^k - \sum_{i=1}^N (1 - \alpha_i^k) + m_s \\ c_2 &= G, \quad \bar{c}_2 = \sum_{i=1}^N \left| \frac{\partial G(\alpha^k)}{\partial \alpha_i} \right| \alpha_i^k - G(\alpha^k) + \bar{G} \end{aligned} \quad (50)$$

ただし、 α_i^k は第 k ステップの i 番目要素の設計変数の値、 $\mathbf{k}_i$ は i 番目要素の剛性マトリックス、 $\mathbf{u}_i$ は同要素に関する節点変位ベクトルである。

(49)式は第 2 章に示した方法で解くことができる。なお、収束を安定させるためには、設計変数のムーブリミットを設定する必要がある。すなわち、各ステップの設計変数の制約を次式のように設定する。

$$\max [x_i, x_i^k (1 - \bar{\varepsilon})] \leq x_i \leq \min [\bar{x}_i, x_i^k (1 + \bar{\varepsilon})] \quad (i=1, \dots, N) \quad (51)$$

ここに、 $\bar{\varepsilon}$ は設計変数のムーブリミットである。

4. 解析例

ここでは、CONLINを用いた密度法による位相最適化手法の有効性を検討するため、文献6)に示しているOC法による同様な方法との比較を行う。

4.1 2次元問題による比較

まず、計算負荷の少ない2次元の例題により、基本的特性を調査する。ただし、有限要素は8節点の応力仮定要素を用いている。図-3に設計領域と境界・荷重条件を示す。要素分割数は、 $50 \times 1 \times 10$ で、橋梁の道路に相当する部分は密度比1に固定している。総密度比の制約値は、設計領域の全要素の密度比を1とした場合の30%に設定した。また、設計変数のムーブリミットは0.1とし、30ステップ以上では、前ステップのムーブリミットを1.1で割ることにより絞り込みを行っている。

まず、図-4は、フィルタリングをかけない場合の解析結果を示したものである。図-4(a)はCONLINを用いた場合の結果、図-4(b)はOC法を用いた場合の結果である。なお、図においては、密度比0.3以下の要素は除いている。また、この場合、CONLINでは、制約条件の緩和は用いていない。

図より、両方法ともほぼ同様の位相が得られており、フィルタリングを用いていないため、チェッカーボード分布が現れていることがわかる。

また、図-5は、コンプライアンス値の収束状況を示したものである。図の横軸は設計変数の更新回数、縦軸は各ステップのコンプライアンス値をCONLINを用いた場合の40ステップのコンプライアンス値で割った値をプロットしたものである。図より、この場合は、CONLINを用いた場合の収束が、OC法を用いた場合に比較して速いことがわかる。

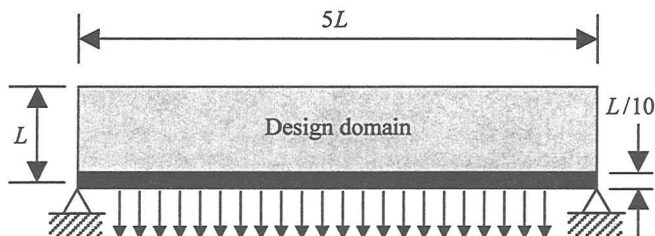
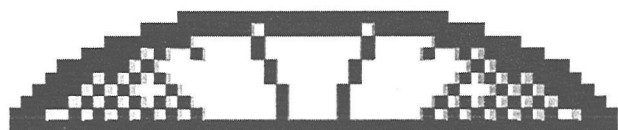
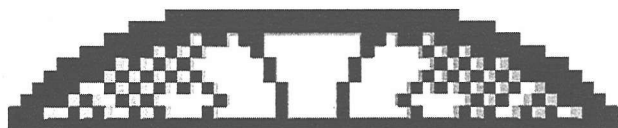


図-3 設計領域と境界条件および荷重条件



(a) CONLINを用いた場合の最適位相



(b) OC法を用いた場合の最適位相

図-4 フィルタリングを用いない場合の最適位相

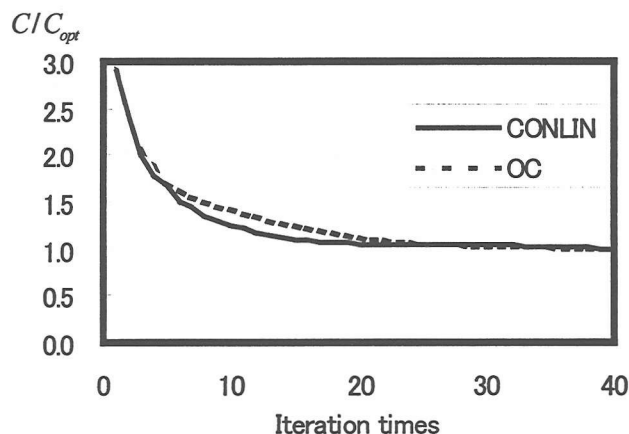
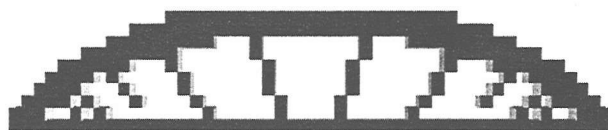
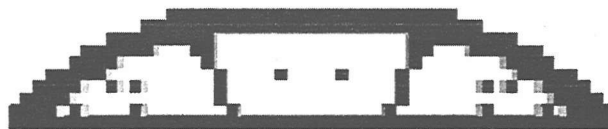


図-5 コンプライアンス値の収束状況

次に、フィルタリングを併用した場合の結果を示す。図-4の場合の重力制御関数値は0.72であったため、まず、重力制御関数値を0.75以上に制約した問題を解く。なお、OC法では、文献6)に示しているように、設計変数を更新する各ステップの内側のループで制約条件をアクティブにするための設計変数の更新を行うが、質量制約に関しては100回、重力制御関数値の制約に関しては5回の更新を行う。これは、重力制御関数の制約が効き過ぎると質量制約が満足されなくなるためである。一方、CONLINでは、フィルタリングを最初のステップからかけると許容解が見つからず、解が発散してしまうため、設計変数の更新回数10ステップ以降からフィルタリングをかけることにする。また、CONLINでは許容解を見つけやすくするために、2.4節に示した制約条件の緩和法を用いる。ただし、(36)式の重み w は5に設定した。なお、この重みは大きくすれば制約条件が厳しくなり、小さくすれば甘くなる。



(a) CONLINを用いた場合の最適位相



(b) OC法を用いた場合の最適位相

図-6 フィルタリングを用いた場合の最適位相

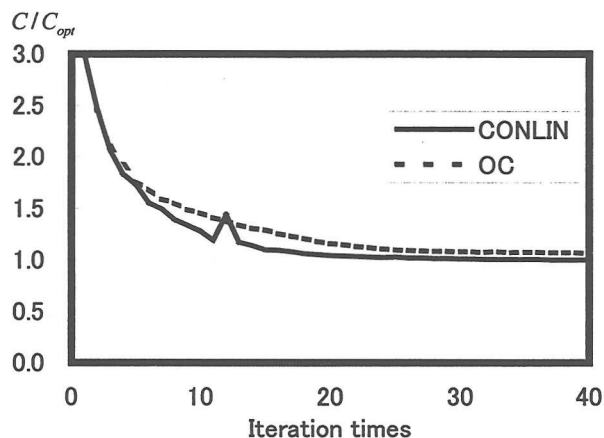


図-7 コンプライアンス値の収束状況

図-6は、それぞれの方法を用いた場合の最適位相を示している。図より、どちらの位相もフィルタリングの効果によりチェッカーボードがほとんど消えていることがわかる。しかし、OC法を用いた場合の結果は、CONLINを用い

た場合の結果に比較して、若干不明瞭な位相となっている。これは、OC法では、重力制御関数の制約条件が厳密に満足されないためである。

図-7は、この時のコンプライアンス値の収束状況を示したものである。図より、CONLINを用いた場合は、重力制御関数の制約が加わる11ステップ以降で若干乱れが見られるが、OC法を用いた場合よりも収束が速く、しかもより低いコンプライアンス値に収束していることがわかる。

4.2 3次元問題による比較

次に、図-3と同様のアーチ橋が求まる例題で、図-8に示す3次元問題¹⁵⁾の解析を行う。図に示すように、荷重は領域下面全体に作用するものとし、道路に相当する部分（下面から高さ $L/10$ の領域）は密度比1で固定する。また、設計領域の下から $2L/3$ の領域（左右 $L/10$ の領域は除く）は、通路として確保するため、密度比0で固定する。それ以外の領域を設計領域とする。要素分割数は、 $50 \times 10 \times 10$ で、総密度比の制約値は、設計領域の全要素の密度比を1とした場合の20%に設定した。また、設計変数のムープリミットは0.1とし、30ステップ以上では、前ステップのムープリミットを1.1で割ることにより絞り込みを行っている。

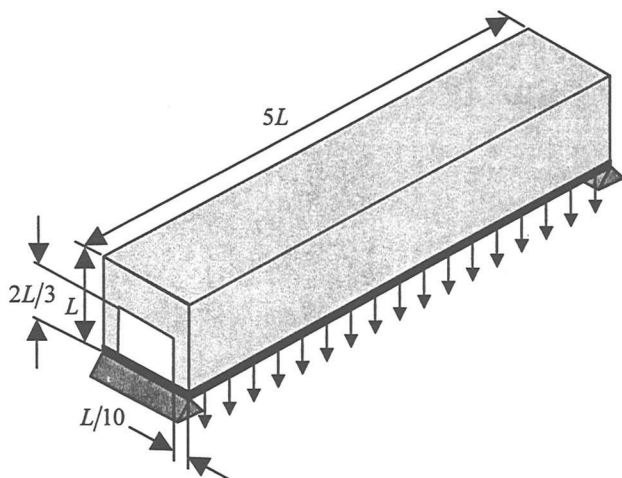


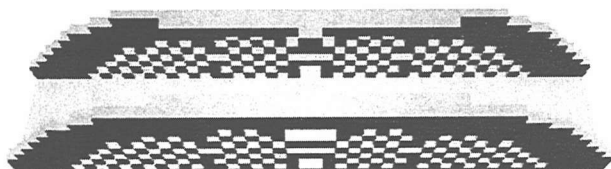
図-8 設計領域と境界条件および荷重条件

図-9は、CONLINを用いた場合の解析結果を示している。フィルタリングを用いる場合では、設計変数の更新回数が10ステップ以降から重力制御関数の制約を加えている。また、(36)式の重み w は5に設定した。重力制御関数の制約値は、制約をかけない場合の重力制御関数値が0.88であったため、0.9と0.91に設定した。なお、図においては、密度比0.3以下の要素は除いている。図より、CONLINを用いた場合は、重力制御関数値を制約することにより、チェッカーボードが消え、よりシンプルな位相が得られることがわかる。

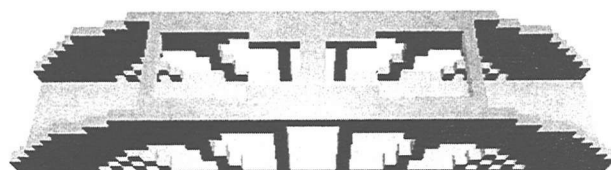
図-10は、OC法を用いた場合の解析結果を示している。OC法を用いた場合、この問題ではフィルタリングの効果

が非常に鈍感であったため、 $\bar{G}=1.0$ に設定したが、図に示すようにチェッカーボードを消すことはできなかった。

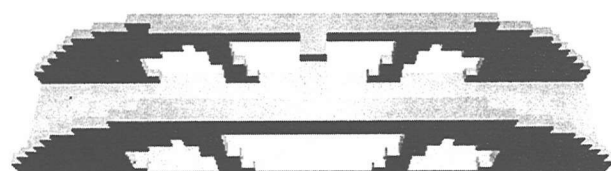
図-11は、図9の(a)と(c)の場合、および図-10の(a)の場合について、コンプライアンス値の収束状況を示したものである。図からわかるように、フィルタリングを用いない場合は、CONLINを用いた場合の収束が、OC法を用いた場合に比較して速く、また、より低い値に収束している。また、フィルタリングを用いた場合は、重力制御関数の制約が加わる10ステップ以降で少し高い値となるが、最終的には、OC法の収束値より若干低い値に収束している。



(a) フィルタリングを用いない場合

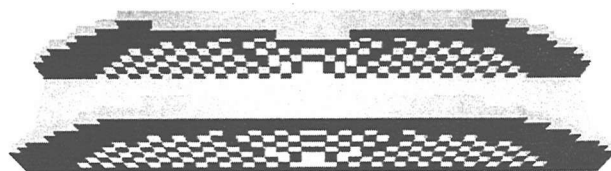


(b) フィルタリングを用いた場合 ($\bar{G}=0.9$)

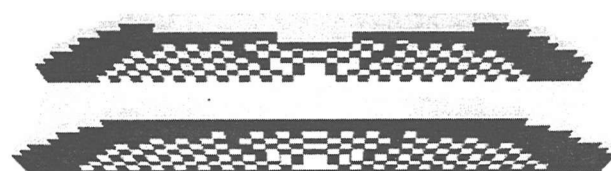


(c) フィルタリングを用いた場合 ($\bar{G}=0.91$)

図-9 CONLINを用いた場合の最適位相



(a) フィルタリングを用いない場合



(b) フィルタリングを用いた場合 ($\bar{G}=1.0$)

図-10 OC法を用いた場合の最適位相

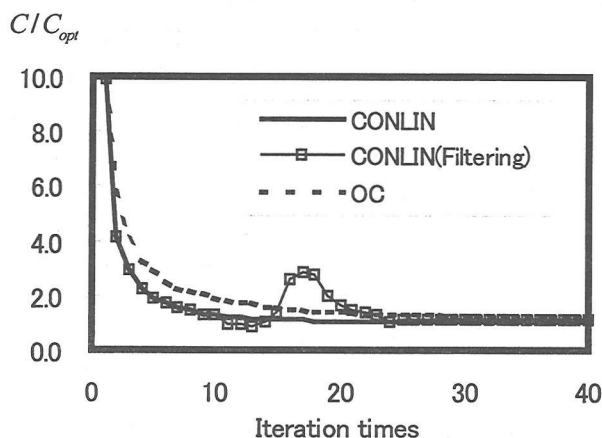
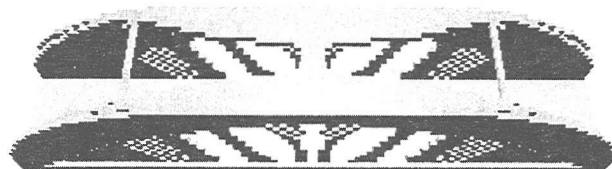


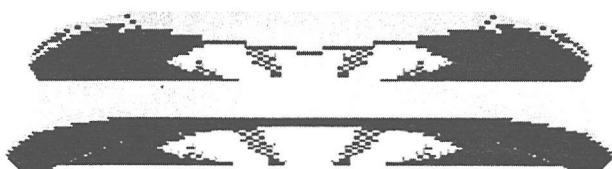
図-11 コンプライアンス値の収束状況



(a) CONLINを用いた場合 ($\bar{G}=0.93$)



(b) CONLINを用いた場合 ($\bar{G}=0.94$)



(c) OCを用いた場合 ($\bar{G}=1.2$)

図-12 分割数を増やした場合の最適位相

最後に、図-8の例題で、要素分割数を $100 \times 20 \times 20$ にした場合の解析を行う。

図-12は、図-9,10と同じ解析条件により解析を行った場合の結果を示している。ただし、フィルタリングを用いない場合の重力制御関数値は0.91であった。図からわかるように、CONLINを用いた場合は、この場合も、フィルタリングが効果的に働いていることがわかる。一方、OCを用いた場合も、 $\bar{G}=1.2$ とすることで、チェッカーボードを消去することは可能であるが、CONLINを用いた場合に比較して若干不鮮明な位相となっている。

5. まとめ

本論文では、密度法による3次元構造物の位相最適化問題の解法に、CONLINを用いた場合の有効性について検討を行った。その結果、以下のような知見が得られた。

- (1) フィルタリング法を用いない場合、すなわち制約条件が1つの場合は、OC法に比較してCONLINの収束が速く、3次元解析ではより低いコンプライアンス値に収束する。
- (2) 重力制御関数値を制約する場合は、制約条件を緩和する必要がある、緩和に用いる重みは5程度が良い。
- (3) 重力制御関数値を制約する場合、CONLINでは、OC法に比較して、重力制御関数値の制約を加えた時点で若干高い値となるが、最終的にはOC法に比較してより低いコンプライアンス値に収束する。なお、収束値に関しては、両手法の結果の差は小さいが、最終ステップでの位相から判断してもCONLINを用いた場合の解の方がより最適解に近いと言える。
- (4) CONLINでは、与えた制約条件をほぼ満足する解が得られるため、3次元解析においてもチェッカーボードを確実に消すことができるが、OC法の場合、3次元解析では、チェッカーボードをうまく消去できない場合がある。

以上の結果から、密度法とフィルタリング法を併用する位相最適化手法では、CONLINを用いる方がより有効であると結論づけられる。

謝辞：本研究を遂行するにあたり、平成12年度科学研究費（基盤研究(c)(2)）の補助を受けた。また、憐くいんと石井恵三氏より貴重な助言を頂いた。ここに、記して感謝の意を表します。

参考文献

- 1) C. Fleury and V. Braibant, Structural Optimization: A new dual method using mixed variables, International Journal for Numerical Methods in Engineering, Vol.23, pp.409-428, 1986
- 2) C. Fleury, CONLIN: an efficient dual optimizer based on convex approximation concepts, Structural Optimization, Vol.1, pp.81-89, 1989
- 3) M. Beekers and C. Fleury, A Primal-Dual Approach in Truss Topology Optimization, Computers & Structures, Vol.64, No.1-4, pp.77-88, 1997
- 4) M. Beekers, Dual methods for discrete structural optimization problems, International Journal for Numerical Methods in Engineering, Vol.48, pp.1761-1784, 2000
- 5) O. Sigmund and J. Petersson, Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima, Structural Optimization, Vol.16, pp.68-75, 1998
- 6) 藤井大地, 鈴木克幸, 大坪英臣, 最適性規準法を用い

た位相最適化におけるフィルタリング法, 日本建築学会構造系論文集, No.543, 2001.5

- 7) 藤井大地, 鈴木克幸, 大坪英臣, ボクセル有限要素法を用いた構造物の位相最適化, 日本計算工学会論文集, Vol.2, pp.87-94, 2000
- 8) K. Svanberg, The method of moving asymptotes – a new method for structural optimization, International Journal for Numerical Methods in Engineering, Vol.24(2), pp.359-373, 1987
- 9) K. Svanberg, A globally convergent version of MMA without linesearch, In Proceedings of the first World Congress of Structural and Multidisciplinary Optimization, Goslar, Germany, Olhoff N, Rozvany GIN (eds), pp.9-16, 1995
- 10) 山川宏, 最適化デザイン, 培風館, pp.167-170, 1993
- 11) 関口美奈子, 菊池昇, 混合的な有限要素剛性マトリックスの導き方に関する一考察—Clough 1960年の論文を

中心として—, 計算工学講演会論文集, Vol.4, No.1, pp.131-134, 1999

- 12) Yang, R.J. and Chuang, C.H., Optimal topology design using linear programming, Computers & Structures, Vol.52(2), 265-275, 1994
- 13) Fernandes, P., Guedes, J.M. and Rodrigues, H. : Topology optimization of three-dimensional linear elastic structures with a constraint on “perimeter”, Computers and Structures, Vol.73, pp.583-594, 1999
- 14) D. Fujii, N. Kikuchi, Improvement of numerical instabilities in topology optimization using the SLP method, Structural Optimization, Vol.19, pp.113-121, 2000
- 15) M. Beckers, Topology optimization using a dual method with discrete variables, Structural Optimization, Vol.17(1), pp.14-24, 1999

(2001 年 4 月 20 日受付)