

## 目次

弓削康平.....	1
コンプライアンス最小問題 .....	1
1 均質化法とトポロジー最適化.....	1
2 密度法 .....	2
3 Lagrange 関数と Kuhn-Tucker 条件.....	2
4 感度解析.....	3
5 最適性基準法.....	4
6 プログラム例.....	4
7 プログラムの実行.....	9
8 チェッカーボード現象と節点密度法.....	11
9 節点密度法のプログラム例 .....	11

## コンプライアンス最小問題

構造最適化の代表的な問題は、

$$\text{Min } \theta = \mathbf{u} \cdot \mathbf{f}$$

$$\text{Subject to } h \leq h_0$$

で定義される。 $\mathbf{f}$  は系に作用する外力であり、 $\mathbf{u}$  は外力の作用点の変位である。また、 $h$  は系に使用する材料の総量を表す。

(Min は「最小化する」、subject to は 「...という制約の下で」の意)

ここで定義した  $\theta$  はコンプライアンスと呼ばれ、剛性最大化の指標となる。

## 1 均質化法とトポロジー最適化

均質化法は下図に示すように微小な周期構造を持つ複合材料の平均化された材料定数を算定する方法である。周期性の最小領域を基本周期領域と呼ぶ。Suzuki と Kikuchi<sup>1)</sup>は、

- 1) 許容設計領域が長方形の空隙を有する基本周期領域からなる複合材料からなるとし、
- 2) 許容設計領域を有限要素に分割して
- 3) 各要素の基本周期領域の長方形空隙の縦横の辺の長さ<sup>2)</sup>と長方形の角度<sup>3)</sup>を設計変数とし、
- 4) コンプライアンス最小を目的関数、
- 5) 材料の総量を制約条件

とする位相最適化手法を提案し、この方向の研究のブームを引き起こした（図2）。

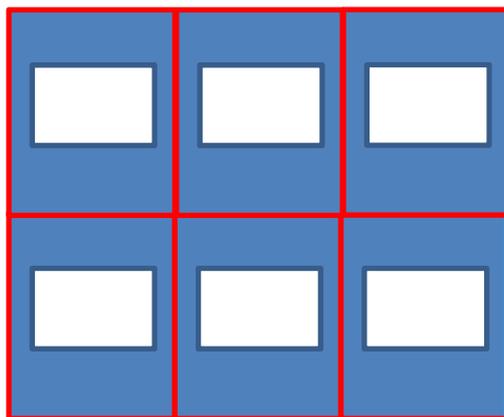


図1 周期性のある微小構造からなる複合材料例

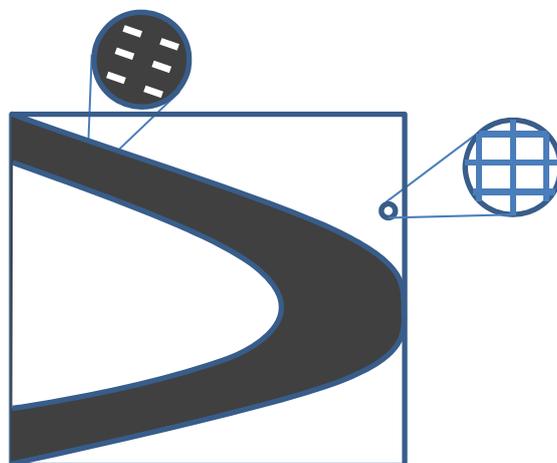


図2 均質化法による最適化

## 2 密度法

均質化法は、実際に空隙を有する複合材料を用いており、これを用いた位相最適化は物理的な意味がはっきりとしているが、一方で、計算アルゴリズムが複雑になる。そこで、材料の密度を変化させることができ、ヤング率が密度のべき乗関数として

$$E = E_0 \rho^k \quad (0 \leq \rho \leq 1) \quad 3$$

で表される仮想的な材料を考え、目的関数が最小となるよう許容設計領域内の密度分布を決定する密度法が提案された。密度法では Bendsøe と Sigmund はこれを SIMP 法<sup>2)</sup>と命名している。密度法は、等方性材料であり、計算が用意である。また、弾塑性材料など非線形材料へも拡張が容易である。一方、仮想的な材料なので中間密度部分の理論的な解釈が困難などの問題点がある。

これらの点をふまえ、近年では、中間部分が発生しないレベルセット法が注目されている。一方、レベルセット法は線形問題でも多大な反復計算を要するなどが短所としてある。

## 3 Lagrange 関数と Kuhn-Tucker 条件

目的関数  $\theta$  と制約条件  $h \leq h_0$  をひとつの関数にまとめた次の関数を定義し、これを Lagrange 関数と呼ぶ。

$$L = \theta + \lambda(h - h_0) \quad 4$$

ここに添字  $\rho_i$  は  $i$  番目の設計変数を、 $\lambda (\geq 0)$  は Lagrange 乗数を表す。

最適条件 (Kuhn-Tucker 条件) は上式を用いて次式で与えられる。

$$\frac{\partial L}{\partial \rho_i} = \frac{\partial \theta}{\partial \rho_i} + \lambda \frac{\partial h}{\partial \rho_i} = 0 \quad (i = 1, 2, \dots, N) \quad 5$$

$$\lambda(h - h_0) = 0 \quad \lambda \geq 0$$

#### 4 感度解析

いま、有限要素法で離散化した系の剛性方程式を

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad 6$$

とし、目的関数を

$$\theta = \mathbf{u}^T \mathbf{f} \quad 7$$

とおく。

ここで  $\mathbf{f}$  を既知とし、6を設計変数で偏微分すると

$$\frac{\partial \mathbf{K}}{\partial \rho_i} \mathbf{u} + \mathbf{K} \frac{\partial \mathbf{u}}{\partial \rho_i} = \mathbf{0}$$

となる。目的関数の感度は上式より

$$\frac{\partial \theta}{\partial \rho_i} = \frac{\partial \mathbf{u}^T}{\partial \rho_i} \mathbf{f} = - \left( \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \rho_i} \mathbf{u} \right)^T \mathbf{f} = - \mathbf{u}^T \left( \frac{\partial \mathbf{K}}{\partial \rho_i} \right)^T (\mathbf{K}^{-T}) \mathbf{f} = - \mathbf{u}^T \left( \frac{\partial \mathbf{K}}{\partial \rho_i} \right) (\mathbf{K}^{-1} \mathbf{f}) = - \mathbf{u}^T \frac{\partial \mathbf{K}}{\partial \rho_i} \mathbf{u} \quad 8$$

となる。特に密度法において要素の密度を設計変数に取るならば、全体剛性に及ぼす要素密度の影響は要素剛性のみ表れるから、 $i$  番目の要素剛性を

$$\mathbf{k}_i = \int_{\Omega} [\mathbf{B}][\mathbf{C}][\mathbf{B}] d\Omega \quad 9$$

$[\mathbf{B}]$  は変位ひずみ関係行列、 $[\mathbf{C}]$  は応力ひずみ関係

とすると8は

$$\frac{\partial \theta}{\partial \rho_i} = \mathbf{u}_i^T \left( \int_{\Omega} [\mathbf{B}] \left[ \frac{\partial \mathbf{C}}{\partial \rho_i} \right] [\mathbf{B}] d\Omega \right) \mathbf{u}_i \quad 10$$

となる。ここに  $\mathbf{u}_i$  は要素  $i$  の節点変位である。特に  $\mathbf{C}$  が弾性の応力ひずみ関係式であれば3より

$$\left[ \frac{\partial \mathbf{C}}{\partial \rho_i} \right] = k \rho_i^{k-1} [\mathbf{C}_0] \quad 11$$

である。

また、材料の総使用量は次式で表される。

$$h = \sum_{i=1}^{NE} \rho_i A_i \quad 12$$

ここに  $A_i$  は要素面積である。よって、その感度は

$$\frac{\partial h}{\partial \rho_i} = A_i \quad 13$$

となる。

## 5 最適性基準法

最適解を求める方法として、Suzuki と Kikuchi 位相最適化では最適性規準法がよく用いられる。最適性基準法は設計変数が多く、制約条件が少ない場合に良いと言われる。

目的関数と制約条件の感度  $\frac{\partial F}{\partial \rho_i}$ 、 $\frac{\partial h}{\partial \rho_i}$  が計算できる場合、最適解は、最適性基準法で求めることができる。最

適性基準法では

- (a)  $\rho_i$  ( $i=1,2,\dots,N$ ) に初期値を与える。
- (b)  $m=0$
- (c) 構造解析を実施する
- (d) 目的関数  $F^{(m)}$  を計算する
- (e)  $F^{(m)} \approx F^{(m-1)}$  なら収束とし計算終了
- (f)  $\frac{\partial F}{\partial \rho_i}$  および  $\frac{\partial h}{\partial \rho_i}$  を計算する
- (g)  $\lambda$  を求めるための反復計算の回数  $k=0$  とし、 $\lambda^k=1$  (適当な値) とおく
- (h)  $\lambda$  が収束するまで以下の計算を反復する
  - 1) 設計変数  $\rho_i^k$  を次のように更新する

$$\rho_i^{k+1} = \begin{cases} \max \left[ (1-\xi)\rho_i^k, \underline{\rho} \right] & \rho_i^k D_i^k \leq \max \left[ (1-\xi)\rho_i^k, \underline{\rho} \right] \\ \rho_i^k D_i^k & \max \left[ (1-\xi)\rho_i^k, \underline{\rho} \right] \leq \rho_i^k D_i^k \leq \min \left[ (1+\xi)\rho_i^k, \bar{\rho} \right] \\ \min \left[ (1+\xi)\rho_i^k, \bar{\rho} \right] & \rho_i^k D_i^k \geq \min \left[ (1+\xi)\rho_i^k, \bar{\rho} \right] \end{cases}$$

ここに

$$\rho_i^k D_i^k = \left[ -\frac{\frac{\partial F}{\partial \rho_i}}{\left( \frac{\partial h}{\partial \rho_i} \lambda^k \right)} \right]^\eta \rho_i^k$$

$\xi$  はムーブリミット (設計変数の変化量の上限)、 $\eta$  は重み係数である。

- 2)  $h = \sum_{i=1}^N A_i \rho_i^k$  を計算する
- 3)  $\lambda^{k+1} = \left[ \frac{h}{h_0} \right]^\eta \lambda^k$
- 4)  $h - h_0 \leq \varepsilon$  なら終了
- 5)  $k=k+1$  として 1) へ
- (i)  $m=m+1$  として (d) へ

なお、3)の $\lambda$ の探索は2分法の方がはるかに収束が早い。

## 6 プログラム例

矩形の許容設計領域を設定し、左端は固定端として、右端中央部に上向き集中荷重をかける場合の位相最適化プログラム例を示す。

プログラムはメインプログラムと有限要素メッシュを作成する関数 mesh.m、最適性基準法に基づいて設計変数を更

新する関数optimal\_criteria.mからなる。

```
clear all
k=4 % E=E*r(ie)^k
%矩形領域と分割数の定義
lx=100:nx=40; % x方向長さと分割数
ly=100:ny=40; % y方向長さと分割数
area=lx*ly/(nx*ny);
%有限要素メッシュの作成
[nnode, nelm, ne, x, y, mprop, free, f, ke]=mesh(lx, ly, nx, ny);

% 設計変数の初期値設定
h0=0.5; % 全体の面積に占める許容材料使用面積
for ie=1:nelm
    r(ie)=h0;
end
convergence=0;old_object=0;itry=0;
comp=zeros(1,10); % コンプライアンスの履歴プロット用配列
for itry=1:40
    % 剛性行列の作成
    K=zeros(2*nnode);
    for ie=1:nelm
        mapn=ne(ie,1:4);
        map= [2*mapn-1 2*mapn]; %要素自由度の並び
        K(map,map)=K(map,map)*(ke*r(ie)^k); % 全体剛性行列に要素剛性行列を加える。
    end

% 変位を求める
u=zeros(2*nnode,1);
u(free)=K(free,free)\f(free); % 連立方程式を解く
object=u(free)'*f(free);
c=abs((object-old_object)/object);
if c<0.01
    convergence=1;
end

% 感度計算
for ie=1:nelm
    mapn=ne(ie,1:4);
    map= [2*mapn-1 2*mapn];
    ue=u(map);
```

```

sensitivity(ie)=-ue'*(k*r(ie)^(k-1)*ke)*ue;
end

%最適性基準法による設計変数の更新
[r]= optimal_criteria(sensitivity,k,r,nelm,area,h0);

comp(itry)=object
U(itry)=max(abs(u));
r
end
plot(comp)
figure;
%
for ie=1:nelm
    xl=x(ne(ie,1:4));
    yl=y(ne(ie,1:4));
    c=[1-r(ie) 1-r(ie) 1-r(ie)];
    fill(xl,yl,c); ;hold on;
end

```

```

%最適性基準法による設計変数の更新
[r]= optimal_criteria(sensitivity,k,r,nelm,area,h0);

comp(itry)=object
U(itry)=max(abs(u));
r
end
plot(comp) %目的関数の履歴をプロット
% 得られた最適形状の表示
for ie=1:nelm
    xl=x(ne(ie,1:4));
    yl=y(ne(ie,1:4));
    c=[1-r(ie) 1-r(ie) 1-r(ie)]; % 塗りつぶしの色 1:白 :黒
    fill(xl,yl,c); ;hold on; % 要素を塗りつぶす
end

```

---

```

function [nnode, nelm, ne, x, y, mprop, free, f, ke] = mesh(lx, ly, nx, ny)
%UNTITLED この関数の概要をここに記述
% 詳細説明をここに記述

```

```

young=200000.0; pois=0.27; % 材料定数
mprop=[young pois] ;
% 節点座標
dx=lx/nx;
dy=ly/ny;
x1=0:dx:lx;
y1=0:dy:ly;
x=[];y=[];
for i=1:ny+1; x=[x x1]; end;
for i=1:nx+1; y=[y; y1:]; end;
y=reshape(y, 1, (nx+1)*(ny+1));

% 要素コネクティビティ
for i=1:nx
    for j=1:ny
        ie=nx*(j-1)+i;
        ne(ie,1)=(nx+1)*(j-1)+i;
        ne(ie,2)=ne(ie,1)+1;
        ne(ie,3)=ne(ie,2)+nx+1;
        ne(ie,4)=ne(ie,3)-1;
    end
end
%
nnode=(nx+1)*(ny+1);
nelm=nx*ny;

% 拘束条件 (左一列の節点は全拘束)
k=0;
for i=1:ny+1
    in=(nx+1)*(i-1)+1;
    k=k+1;fix(k)=2*in-1;
    k=k+1;fix(k)=2*in;
end
% 荷重条件 (1点に対する集中荷重)
f=zeros(2*nnode,1);
in=(nx+1)*(ny/2+1);
f(2*in)=1;
% 拘束されていない自由度のリスト作成
free=setdiff([1:2*nnode],fix) ;

```

```

% 要素剛性行列の作成
ke=zeros(8) ;
xl=[0 dx dx 0];
yl=[0 0 dy dy];
p=1/3^0.5 ; s=[-p p p -p]; t=[-p -p p p]; % 積分点パターンメータ座標
for i=1:4 % 積分点のループ

% 4角形要素の変位ひずみ行列
H2=0.25*[-1+t(i) 1-t(i) 1+t(i) -1-t(i); -1+s(i) -1-s(i) 1+s(i) 1-s(i)];
ZE=[0 0 0 0;0 0 0 0];
H3=[H2 ZE; ZE H2];
dx=H2*xl' ; dy=H2*yl' ;
J=[dx dy];
ds=det(J);
G=inv(J);
sx=G(1,1); tx=G(1,2);sy=G(2,1);ty=G(2,2);
H1=[sx tx 0 0; 0 0 sy ty;sy ty sx tx] ;
b=H1*H3;
% 応力ひずみ関係式 [C]
young=mprop(1); pois=mprop(2); % 材料定数
c=zeros(3);
c1=young/(1-pois*pois); c2=pois*c1;c3=young/(2*(1+pois));
c(1,1)=c1; c(2,2)=c1;c(1,2)=c2; c(2,1)=c2;c(3,3)=c3;

ke=ke+ds*b'*c*b; % 剛性行列の計算
end
end

```

---

```

function [r]= optimal_criteria(sensitivity, k, r, nelm, area, h0)
ramda=1; % 初期値
eta=1;
mlimit=0.01;
kk=0;
rmin=0;rmax=1;
ra=r;
while(kk<=10000)
    if kk>10000
error('ERROR: Failed in getting ramda');

```

```

end
cvol=0;tvol=0;
for ie=1:nelm
    Dr=(-sensitivity(ie)/(area*ramda))^eta*r(ie);
    cmin=max((1-mlimit)*r(ie),rmin);
    cmax=min((1+mlimit)*r(ie),rmax);
    if Dr<=cmin
        ra(ie)=cmin;
    elseif Dr< cmax
        ra(ie)=Dr;
    else
        ra(ie)=cmax;
    end
    cvol=cvol+area*ra(ie);
    tvol=tvol+area;
end
h=cvol/tvol;
ramda=(h/h0)^eta*ramda; % 2分法でやる方が早いがここでは簡単にこの式でやる
    if abs((h-h0)/h0)<0.001
        break; % 収束 変数更新終了
    end
kk=kk+1;
end
r=ra; % 更新された設計変数

```

---

## 7 プログラムの実行

プログラムを実行すると目的関数の履歴と得られた最適設計の密度分布が下図のように得られる。

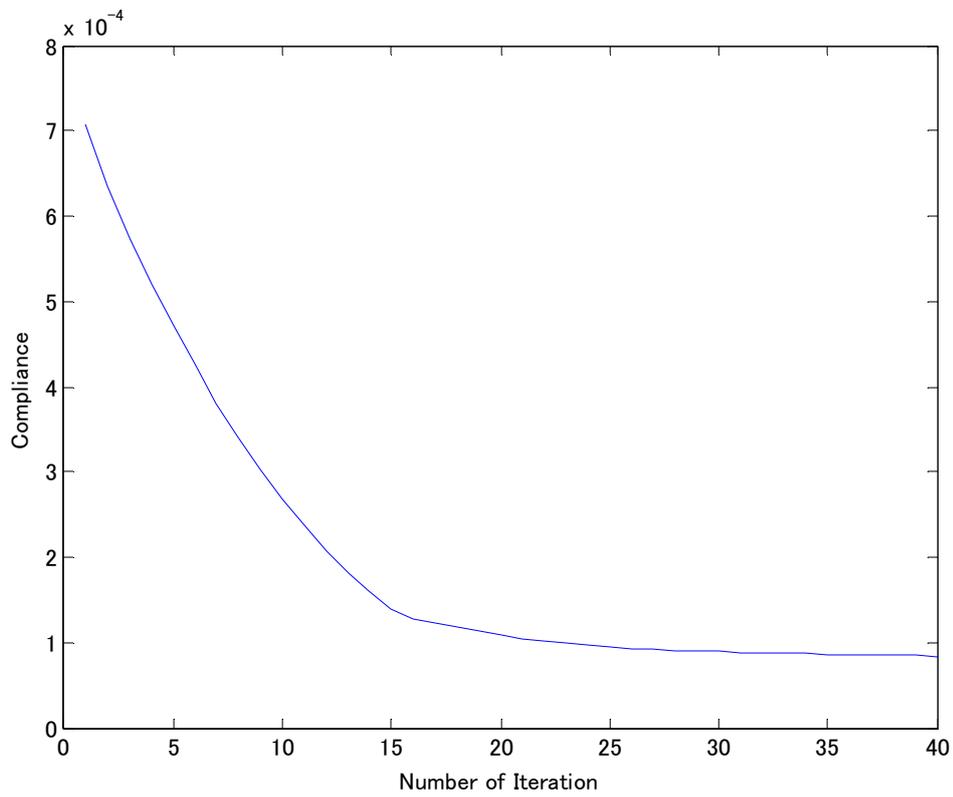


図 3 目的関数の履歴

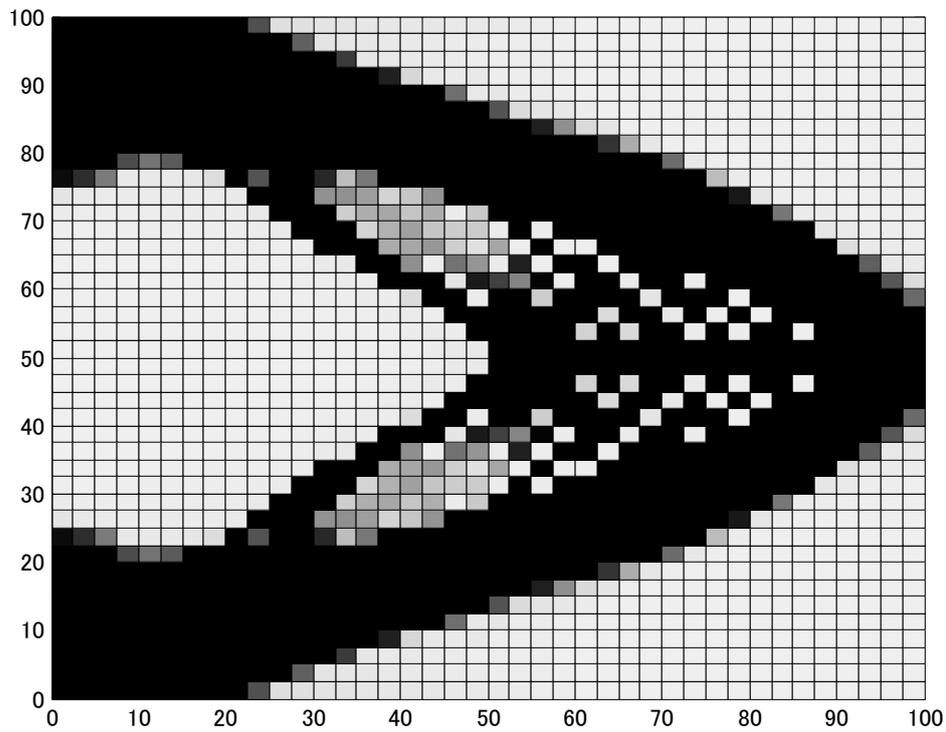


図 4 最適設計 (要素密度)

## 8 チェッカーボード現象と節点密度法

上の最適設計として得られた材料の分布をみると中央部がチェッカーボード状となっている。これに対する一番簡単な対策は、設計変数を要素密度から節点密度に変更することである。

この場合目的関数および体積制約条件に関する設計変数の感度は

$$\frac{\partial \theta}{\partial \rho_i} = -\mathbf{u}^T \frac{\partial \mathbf{K}}{\partial \rho_i} \mathbf{u} = \sum_{J=1}^{JE} \left( -\mathbf{u}_J^T \frac{\partial \mathbf{k}_J}{\partial \rho_i} \mathbf{u}_J \right) = \sum_{J=1}^{JE} \left( -\mathbf{u}_J^T \frac{\partial \mathbf{k}_J}{\partial \rho_i} \mathbf{u}_J \right) = \sum_{J=1}^{JE} \left( -\frac{k \rho_i^{(k-1)}}{4} \mathbf{u}_J^T \mathbf{k}_J^0 \mathbf{u}_J \right) \quad 14$$

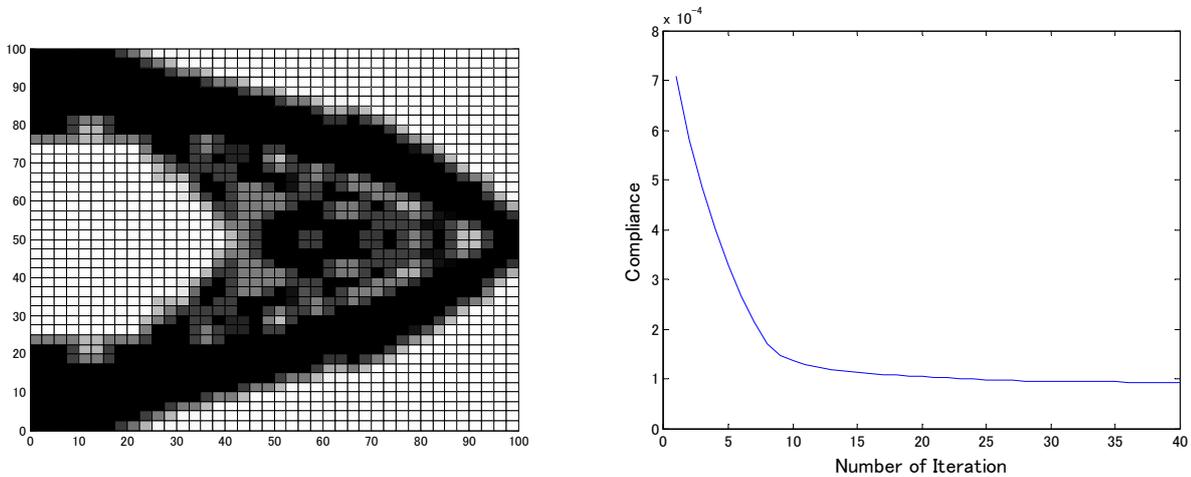
なお上式においては以下の関係を利用した。

$$\frac{\partial \mathbf{k}_J}{\partial \rho_i} = \frac{1}{4} \int_{\Omega} [\mathbf{B}]^T \left[ \frac{\partial \mathbf{C}}{\partial \rho_i} \right] [\mathbf{B}] d\Omega = \frac{k \rho_i^{(k-1)}}{4} \mathbf{k}_J^0$$

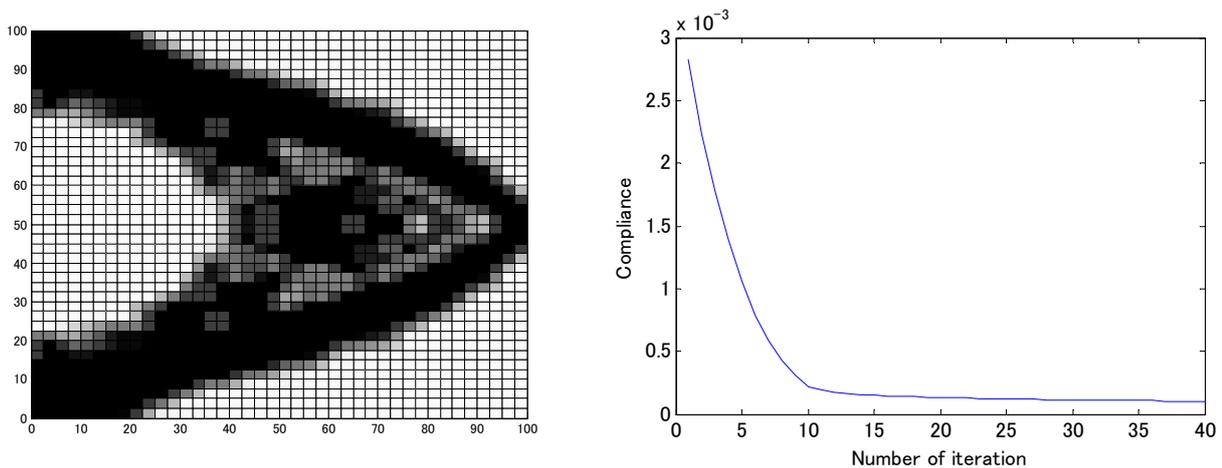
ここに  $\mathbf{k}_J^0$  は要素  $J$  が  $\rho = 1$  の場合の要素剛性行列である。

設計変数を節点密度に変更したときの計算結果を以下に示す。

### 1) Move limit = 1.0



### 2) move limit=0.8



節点密度法のプログラム例を以下に示す。なお、mesh.m は上と同じなので再表記しない。

## 9 節点密度法のプログラム例

```
clear all
k=6 % E=E*r (ie)^k
```

```

%矩形領域と分割数の定義
lx=100;nx=40; % x方向長さ と 分割数
ly=100;ny=40; % y方向長さ と 分割数
%有限要素メッシュの作成
[nnode, nelm, ne, x, y, mprop, free, f, ke, area]=mesh(lx, ly, nx, ny);

% 設計変数の初期値設定
h0=0.5 ; % 全体の面積に占める許容材料使用面積
for ie=1:nelm; r_elm(ie)=h0; end;
for in=1:nnode; r_node(in)=h0; end;

itry=0;
comp=zeros(1,10); % plotデータ用メモリ

for itry=1:40

% 剛性行列の作成
K=zeros(2*nnode);
for ie=1:nelm
    mapn=ne(ie,1:4);
    map= [2*mapn-1 2*mapn]; %要素自由度の並び
    K(map, map)=K(map, map)+(ke*r_elm(ie)^k); %
end

% 変位を求める
u=zeros(2*nnode,1);
u(free)=K(free, free)\f(free); % 連立方程式を解く
object=u(free)'*f(free);

% 感度計算
sensitivity=zeros(1, nnode);
for ie=1:nelm
    mapn=ne(ie,1:4);
    map= [2*mapn-1 2*mapn];
    ue=u(map);
    ee=-ue'*(ke)*ue/4;
    sensitivity(mapn)=sensitivity(mapn)+k*r_node(mapn).^(k-1)*ee;
end

% 最適性基準法による設計変数の更新

```

```

[r_node]= optimal_criteria(sensitivity, k, r_node, nnode, area, h0);
% 節点密度→要素密度
for ie=1:nelm
    mapn=ne(ie, 1:4);
    r_elm(ie)=sum(r_node(mapn))/4;
end

comp(itry)=object; % plotデータの記録
U(itry)=max(abs(u));% plotデータの記録

end
figure;
plot(comp)
figure;
%
for ie=1:nelm
    xl=x(ne(ie, 1:4));
    yl=y(ne(ie, 1:4));
    c=[1-r_elm(ie) 1-r_elm(ie) 1-r_elm(ie)];
    fill(xl, yl, c); ;hold on;
end
function [r_node]= optimal_criteria(sensitivity, k, r_node, nnode, area, h0)
ramda=1; % 初期値
eta=1;
mlimit=0.05;
kk=0;
rmin=0;rmax=1;
ra=r_node;
while(kk<=10000)
    if kk>10000
        error('ERROR: Failed in getting ramda');
    end
    cvol=0;tvol=0;
    for in=1:nnode
        Dr=(-sensitivity(in)/(area(in)*ramda))^eta*r_node(in);
        cmin=max((1-mlimit)*r_node(in), rmin);
        cmax=min((1+mlimit)*r_node(in), rmax);
        if Dr<=cmin
            ra(in)=cmin;
        elseif Dr< cmax

```

```

        ra(in)=Dr;
    else
        ra(in)=cmax;
    end
    cvol=cvol+area(in)*ra(in);
    tvol=tvol+area(in);
end
h=cvol/tvol;
ramda=(h/h0)^eta*ramda; % 2文法でやる方が早いここでは簡単にこの式でやる
    if abs((h-h0)/h0)<0.001
        break; % 収束 変数更新終了
    end
kk=kk+1;
end

r_node=ra;

```

#### 参考文献

- 1) Katsuyuki Suzuki, Noboru Kikuchi, "A homogenization method for shape and topology optimization", Computer Methods in Applied Mechanics and Engineering, Volume 93, Issue 3, December 1991, Pages 291-318
- 2) M. P. Bendsøe and O. Sigmund., Topology optimization : theory, methods and applications. Springer, , 2003.