Solid Isotropic Material with Penalisation (S.I.M.P.)

$$E_e(x_e) = E_{\min} + x_e{}^p (E_0 - E_{\min})$$

$E_{\min}$ is a small positive number so stiffness matrix doesn’t become singular

For our purpose we will assume

$$E_0 = 1$$
$$E_{\min} = 0.0001$$

Solid Isotropic Material with Penalisation (S.I.M.P.)

$$\min_{x} c(\mathbf{x}) = \mathbf{U}^{\mathrm{T}}\mathbf{K}\mathbf{U} = \sum_{e=1}^{N} E_e(x_e)\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e$$

Subject to

$$\frac{V(\mathbf{x})}{V_0} = f$$

$$\mathbf{K}\mathbf{U} = \mathbf{F}$$

$$\mathbf{0} \leq \boldsymbol{x} \leq \mathbf{1}$$

$$c = E_{\min} + x_e^{p}\,(E_0 - E_{\min})\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e$$

$$\frac{\partial c}{\partial x_e} = -p x_e^{p-1}\,(E_0 - E_{\min})\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e$$

Algorithm pseudocode – Part 1

Input $n_x, n_y, V_f, r_{\min}, p, \upsilon, E\text{max}, E\text{min}$

Initialize $\mathbf{x} = V_f\ \mathbf{1}$,	size $[n_y, n_x]$
Initialize $\mathbf{dc} = \mathbf{0}$, size $[n_y, n_x]$	size $[n_y, n_x]$
Set $n_{\mathrm{DOF}} = 2(n_x + 1)(n_y + 1)$	1
Derive $K_e = \mathrm{lk}(E_{\max}, \upsilon)$	size $[8,8]$
Derive $\mathrm{e}_{\mathrm{DOF,mat}} = \mathrm{edofMatFun}(E_{\max}, \upsilon)$	size $[m, 8]$
Define global Force vector, $\boldsymbol{F} = \mathbf{0}$	Size $[n_{\mathrm{DOF}}, 1]$
Define free and fixed dofs	

Algorithm pseudocode – Part 2

Input $n_x, n_y, V_f, r_{\min}, p, \upsilon, E\max, E\min$

While change > 0.05 and counter < max_iteration	
Derive DOF displacement $\boldsymbol{U} = \mathrm{FE}\left(n_x, n_y, x, p, K_e, e_{dof,Mat}, F, freeDOF, fixedDOF\right)$	Size $[n_{\mathrm{DOF}}, 1]$
Set compliance $c = 0$	1
Loop through n_y	Size [8,8]
Loop through n_x	Size $[m, 8]$
Find row in $\mathrm{e}_{\mathrm{DOF,mat}}$ that correspond to the current element	Size [1,8]
Find U_e from U that correspond to the current element using above	Size [8,1]
Calculate c_e compliance per element $c_e = \mathrm{x}_e^p \mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e$	1
Calculate $-\frac{dc_e}{dx_e} = -p\mathrm{x}_e^{p-1} \mathbf{u}_e^T \mathbf{K}_e \mathbf{u}_e$	size $\left[\mathrm{n}_y, n_x\right]$

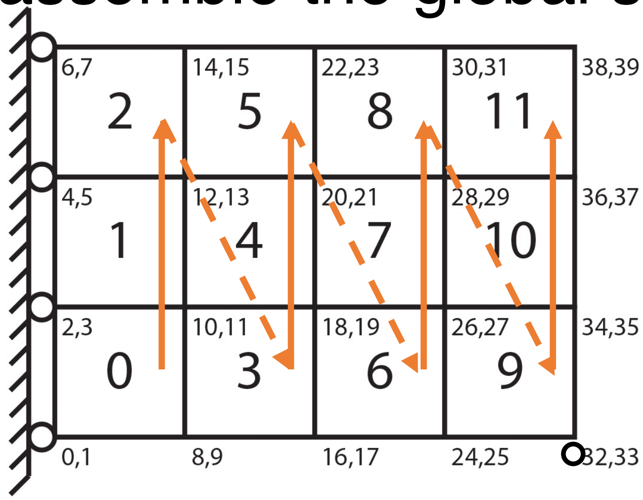
Algorithm pseudocode – Part 3

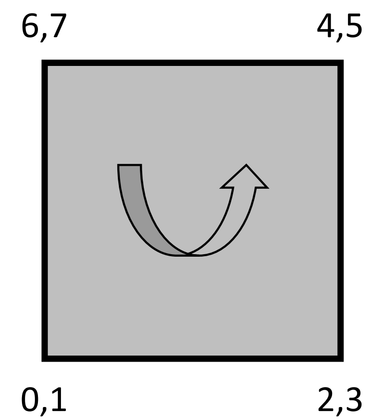
Exit the element loop but not the iteration loop	
Apply filter function $\frac{dc_e}{dx_e} = \mathrm{sFilter}\left(n_x, n_y, x, \frac{dc_e}{dx_e}, r_{\min}\right)$	size $[\mathrm{n}_y, n_x]$
Calculate new x by $\mathrm{x}_{\mathrm{new}} = \mathrm{oc}\left(n_x, n_y, x, \frac{dc_e}{dx_e}, V_f, E_{\min}, E_{\max}\right)$	size $[\mathrm{n}_y, n_x]$
Calculate $Change = \max(\mathbf{x} - \mathbf{x}_{\mathrm{old}})$	1

K matrix for E = 1, v = 0.3

0.49450549	0.17857143	-0.3021978	-0.0137363	-0.2472528	-0.1785714	0.05494505	0.01373626
0.17857143	0.49450549	0.01373626	0.05494505	-0.1785714	-0.2472528	-0.0137363	-0.3021978
-0.3021978	0.01373626	0.49450549	-0.1785714	0.05494505	-0.0137363	-0.2472528	0.17857143
-0.0137363	0.05494505	-0.1785714	0.49450549	0.01373626	-0.3021978	0.17857143	-0.2472528
-0.2472528	-0.1785714	0.05494505	0.01373626	0.49450549	0.17857143	-0.3021978	-0.0137363
-0.1785714	-0.2472528	-0.0137363	-0.3021978	0.17857143	0.49450549	0.01373626	0.05494505
0.055	-0.0137363	-0.2472528	0.17857143	-0.3021978	0.01373626	0.49450549	-0.1785714
0.014	-0.3021978	0.17857143	-0.2472528	-0.0137363	0.05494505	-0.1785714	0.49450549

assemble the global stiffness matrix (edofMatFun)





0	1	2	3	4	5	6	7	Local	
0	1	8	9	10	11	2	3	Elem	0
2	3	10	11	12	13	4	5	Elem	1
4	5	12	13	14	15	6	7	Elem	2
8	9	16	17	18	19	10	11	Elem	3
10	11	18	19	20	21	12	13	Elem	4
12	13	20	21	22	23	14	15	Elem	5
16	17	24	25	26	27	18	19	Elem	6
18	19	26	27	28	29	20	21	Elem	7
20	21	28	29	30	31	22	23	Elem	8
24	25	32	33	34	35	26	27	Elem	9
26	27	34	35	36	37	28	29	Elem	10
28	29	36	37	38	39	30	31	Elem	11

Finite element analysis FE()

Input $n_x, n_y, x, p, K_e, e_{dof,mat}, F, freeDOF, fixedDOF$

Initialize $\mathbf{K} = \mathbf{0}$	size $[n_{DOF}, n_{DOF}]$
Initialize $\boldsymbol{U} = \mathbf{0}$	size $[n_{DOF}, 1]$
Loop through each element (ny , and within each ny, loop through nx)	
Assemble $\mathbf{K}$ by getting the row in $e_{dof,mat}$ corresponding to the element, called id	size $[8,1]$
$K[id, id] = K[id, id] + x_e^p (E_{max} - E_{min}) K_e$	
Solve for $\mathbf{U}_{free} = K_{free,free} \backslash \mathrm{F}_{free}$	
Return $\mathbf{U}$	

Optimality Criteria

$c = x_e^p\ (E_0 - E_{\min})\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e$

$\frac{\partial c}{\partial x_e} = -p x_e^{p-1}\ (E_0 - E_{\min})\mathbf{u}_e^{\mathrm{T}}\mathbf{k}_0\mathbf{u}_e$

$$x_e^{new} = \begin{cases} \max(0, x_e - m) & \text{if } x_e B_e^\eta \leq \max(0, x_e - m) \\ \min(0, x_e - m) & \text{if } x_e B_e^\eta \geq \min(0, x_e - m) \\ x_e B_e^\eta & \text{otherwise} \end{cases}$$

$m > 0$ (e.g. 0.2)

$$B_e = -\frac{\frac{\partial c}{\partial x_e}}{\lambda \frac{\partial V}{\partial x_e}}$$

$$\frac{\partial V}{\partial x_e} = 1$$

$$B_e = -\frac{1}{\lambda}\frac{\partial c}{\partial x_e}$$

Optimality Criteria

Input $n_x, n_y, x, p, K_e, e_{dof,mat}, F, freeDOF, fixedDOF$

Initialize $\mathbf{K} = \mathbf{0}$	size $[n_{DOF}, n_{DOF}]$
Initialize $\boldsymbol{U} = \mathbf{0}$	size $[n_{DOF}, 1]$
Loop through each element (ny , and within each ny, loop through nx)	
Assemble $\mathbf{K}$ by getting the row in $e_{dof,mat}$ corresponding to the element, called id	size $[8,1]$
$K[id, id] = K[id, id] + x_e^p(E_{max} - E_{min})K_e$	
Solve for $\mathbf{U}_{free} = K_{free,free} \backslash \mathrm{F}_{free}$	
Return $\mathbf{U}$	

Optimality Criteria, pseudo code 1

Input $n_x, n_y, \frac{\partial c}{\partial x_e}, x_e$

set $\lambda_L = \mathbf{0}, \lambda_H = 10^9, \mathrm{m} = 0.2$

While $\lambda_H - \lambda_L \geq 10^{-4}$

$\lambda_M = \frac{1}{2}(\lambda_H + \lambda_L)$

calculate $B_e^\eta = \frac{1}{\lambda_M}\left(-\frac{\partial c}{\partial x_e}\right)^{\frac{1}{2}}$

for i_x in n_x:

for i_y in n_y:

if $x_e[i_y, i_x]B_e^\eta[i_y, i_x] \leq \max(E_{min}, x_e[i_y, i_x] - m)$

$x_e^{new}[i_y, i_x]$=$\max(E_{min}, x_e[i_y, i_x] - m)$

elif $x_e[i_y, i_x]B_e^\eta[i_x, i_y] \geq \min(E_{max}, x_e[i_y, i_x] + m)$

$x_e^{new}[i_y, i_x]$=$\min(E_{max}, x_e[i_y, i_x] + m)$

else

$x_e^{new}[i_y, i_x]$=$x_e[i_y, i_x]B_e^\eta[i_y, i_x]$

Optimality Criteria, pseudo code 2

Input $n_x, n_y, \frac{\partial c}{\partial x_e}, x_e$

While $\lambda_H - \lambda_L \geq 10^{-4}$

... (see page 1)

if $\mathrm{sum}(x_e^{new}) - frac \cdot n_x \cdot n_y > 0$

$\lambda_L = \lambda_M$

else

$\lambda_H = \lambda_M$

Sensitivity filter

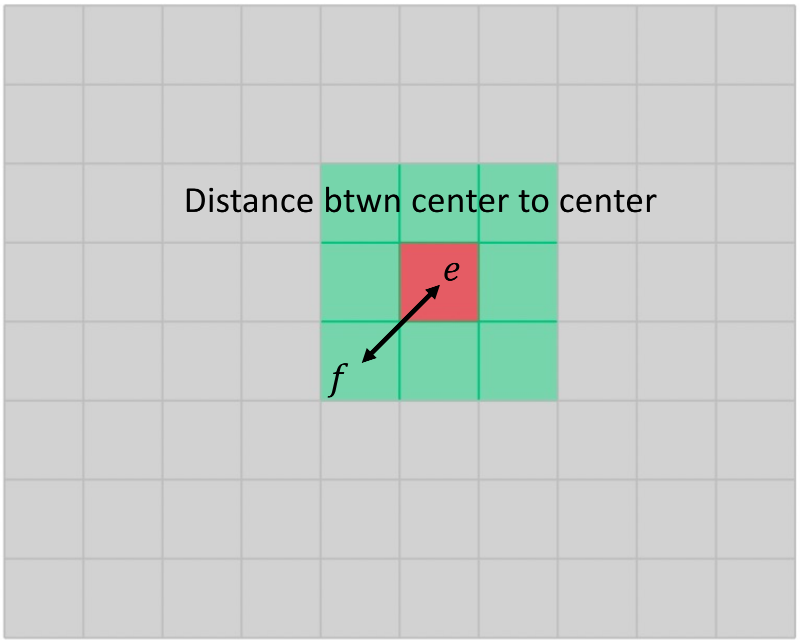
function sFilter(n_x, n_y, x_e, $\frac{dc_e}{dx_e}$, $r_{\min}$)	
Set $\frac{dc_e}{dx_e}^{\text{new}} = 0$	$[n_y, n_x]$
for i_x from 0 to n_x	
for i_y from 0 to n_y	
$\mathrm{H_f} = 0$	
$\mathrm{x_{min}} = \max(i_x - \mathrm{floor}(r_{\min}), 0)$	Integer 1
$\mathrm{x_{max}} = \min(i_x + \mathrm{floor}(r_{\min}) + 1, n_x)$	Integer 1
$y_{\min} = \max(i_y - \mathrm{floor}(r_{\min}), 0)$	Integer 1
$\mathrm{y}_{max} = \min(i_y + \mathrm{floor}(r_{\min}) + 1, n_y)$	Integer 1
for k from $\mathrm{x_{min}}$ to $\mathrm{x_{max}}$	
for l from $\mathrm{y_{min}}$ to $\mathrm{y_{max}}$	
$\mathrm{dist} = r_{\min} - \sqrt{(i_x - k)^2 + (i_y - l)^2}$	
$\mathrm{H_f} \mathrel{+}= \max(0, \mathrm{dist})$	

$$\sum_{f=1}^{N}\left(H_f x_f \frac{\partial c}{\partial x_f}\right)$$

$\frac{dc_e}{dx_e}^{\text{new}}[i_y, i_x] \mathrel{+}= \max(0, \mathrm{dist}) \cdot x_e[i_y, i_x] \cdot \frac{dc_e}{dx_e}[i_y, i_x]$
$\frac{dc_e}{dx_e}^{\text{new}}[i_y, i_x] = \frac{1}{x_e[i_y, i_x] \cdot \mathrm{H_f}} \cdot \frac{dc_e}{dx_e}^{\text{new}}[i_y, i_x]$
return $\frac{dc_e}{dx_e}^{\text{new}}$

$$\frac{\partial c}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^{N} H_f} \sum_{f=1}^{N}\left(H_f x_f \frac{\partial c}{\partial x_f}\right)$$

Sensitivity filter



$$\frac{\partial c}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^{N} H_f} \sum_{f=1}^{N} \left(H_f x_f \frac{\partial c}{\partial x_f} \right)$$

The convolution operator (weight factor)

$$H_f = \mathrm{r_{min}} - \mathrm{dist}(e, f)$$

For f

$$\{f \in N \mid \mathrm{dist}(e, f) \leq \mathrm{r_{min}}\}$$

In all $e = 1, \ldots, N$

Sensitivity filter

Input n_x, n_y, x, dc, r_{min}	
Initialize $\mathbf{dc}_{\text{new}} = \mathbf{0}$	size $[n_y, n_x]$
Loop through each element (n_y , and within each n_y, loop thru. n_x)	
Loop through each elem f within r_{min} around current elem e	
$H_f \mathrel{+}= \max\big(0, r_{min} - dist(e,f)\big)$ $\mathbf{dc}_{\text{new,e}} = \max\big(0, r_{min} - dist(e,f)\big)\, x_f\, dc$	$\frac{\partial c}{\partial x_e} = \frac{1}{x_e \sum_{f=1}^{N} H_f} \sum_{f=1}^{N} \left(H_f x_f \frac{\partial c}{\partial x_f} \right)$
Now calculate $\mathbf{dc}_{\text{new,e}} = \frac{\mathbf{dc}_{\text{new,e}}}{x_e H_f}$	
Return $\mathbf{dc}_{\text{new}}$	