Explicit topology optimization approaches

Internship report

Advisors: Prof. Joseph MORLIER (ISAE SUPAERO, DMSM)

PhD candidate Simone CONIGLIO (Airbus)

School referent: Prof. Miguel CHARLOTTE (ISAE SUPAERO, DMSM)

Mahfoud HERRAZ

ISAE Supaero,Toulouse, France

Academic year: 2017/2018

Table of content

Abstract: 5

Introduction: 5

Moving Morphable Components (MMC) framework: 7

MMC formulation: 7

Enhancement of the MMC code: 11

KKT condition: 11

MMA parameters: 12

KS function to approximate max: 12

Appropriate normalization of gradients: 12

Heaviside parameter alpha: 12

Volume gradient correction: 12

Moving Node Approach (MNA) framework: 13

Method of Moving Asymptotes (MMA) optimizer: 17

Extension to curved components: 20

New equivalent design variables: 22

Benchmarking of methods: 23

1) Cantilever beam 23

Original MMC code results: 24

Enhanced MMC code results: 25

MMC with curvature results: 26

MNA code results: 27

MNA with curvature results: 28

SIMP (top88 with MMA) results: 29

2) MBB beam 30

Original MMC code results: 31

Enhanced MMC code results: 32

MMC with curvature results: 32

MNA code results: 33

MNA with curvature results: 35

SIMP (top88 with MMA) results: 36

3) L-shape beam 37

Original MMC code results: 37

Enhanced MMC code results: 38

MMC with curvature results: 39
MNA code results: 40
SIMP (top88 with MMA) results: 42
4) Self-supported beam 43
Enhanced MMC code results 43
MMC with curvature results 44
MNA code results 45
MNA with curvature results 47
Enhanced MMC code results 48
MMC with curvature results: 49
MNA code results 50
MNA with curvature results 51
5) Mitchell truss 53
Enhanced MMC code results 53
MMC with curvature results 54
MNA code results 55
MNA with curvature results 56
6) Wing rib optimization 57
Enhanced MMC code results: 58
MMC with curvature results 59
MNA code results 60
MNA with curvature results 62
Introducing stress constraint: 63
Example on MMC 66
Example on MNA 68
Airbus project: 69
2D case results 71
MMC results 71
MNA results 73
SIMP results: 74
3D case results 75
MMC results 75
MNA results: 76
SIMP results: 77

Discussion on MNA/MMC differences 77
Conclusion: 79
References: 80

Abstract:

In this report, a new Moving Nodes Approach (MNA), based on Finite Element (FE) analysis and material penalization law, is presented. In addition, a set of modifications has been made to the Moving Morphable Components (MMC) code based on Ersatz material model, and both codes are compared for academic test cases. Moreover, extension to circular curved components had been successfully introduced for both MNA and MMC approaches, and proved to be useful. Stress constraint was also integrated to the topology optimization problem under MMC and MNA frameworks. Finally, in the context of an Airbus project, an industrial case is carried out under MNA and MMC.

Introduction:

Topology optimization deals with finding the optimal material layout of a structure and it is widely used in industry as a numerical tool, especially in the early design phases to reduce the cost and save mass. Besides, topology optimization classic methods have already been implemented in commercial software, such as Altair-OptiStruct, Nastran and Abaqus to solve real engineering design optimization problems. Despite this success, a withstanding challenge in topology optimization is the ability to impose geometric considerations to render designs that are amenable to production. As most industry practitioners of topology optimization can attest to, a design obtained via topology optimization is rarely produced as-is. Consequently, following the topology optimization, design engineers create designs that try to capture as much as possible the material distribution of the optimal topology, but incorporate necessary aspects of the production process. This 'interpreted' design often incurs in a significant detriment of the structural performance. As a result, the engineer must spend a significant amount of time performing design changes by trial and error, which is not only a resource-intensive task, but one that leads to a suboptimal design. This is arguably the primary reason why topology optimization has not yet earned a permanent place in engineering workflows, despite the wide availability of commercial software.

Classic topology optimization approaches are implicit, two mains categories could be mentioned; element density based approach, for instance the well-known SIMP approach, and node based approach such as the Level Set method. In this kind of approaches, design variables are densities in elements of the FEM mesh, or values of a level set function at mesh nodes, and design is determined implicitly by mean of density field and depends on the mesh.

Explicit topology optimization is receiving recently more and more attention. Unlike implicit topology optimization, design variables are often geometrical and the solution (structure's shape) is explicitly determined by design variables. To the author's knowledge, the idea of using an explicit geometric design representation was first explored in the bubble method (ESCHENAUER et al. 1994), where splines are used to represent the structural boundaries, with the control points of the splines being the design parameters.

Several works had been done on the explicit topology optimization: a moving node approach was proposed in (OVERVELDE's thesis 2012), it's a flow-inspired meshless method based on the meshless Element-Free Galerkin (EFG) method, yet it presented many difficulties, and it was suggested that the method might be improved by replacing the EFG method with FEM.

Also, a method for the continuum-based topology optimization of structures made of discrete elements was proposed by (NORATO et al. 2015), where the structure is obtained via the layout of linearly elastic planar structures made of bars of fixed width and semicircular ends. The design space for the optimization consists of the endpoint locations of the bar's medial axes and their out-of-plane thicknesses. Latter, a more general method for the continuum-based topology optimization was developed by (NORATO 2018), whereby the structure is represented by the union of super-shapes. Super-shapes are an extension of super-ellipses that can exhibit variable symmetry as well as asymmetry and that can describe through a single equation, the so-called super-formula, a wide variety of shapes, including geometric primitives.

In the same context, a moving morphable components (MMC) approach was introduced by (GUO et al. 2014) and then improved using the Ersatz material model in (GUO et al. 2015) where a 2D Matlab code was provided, components with curved skeletons (central lines) were introduced in (GUO et al 2016a). The MMC method was extended to 3D case in (GUO et al. 2016b). In addition, a feature-driven topology optimization that deals with engineering features of arbitrary shapes was introduced in (GUO et al 2016c) this was achieved by constructing KS functions (of engineering features) by means of level set functions (of primitive components) and Boolean operations (to perform union an intersection of components). The MMC method was then improved to be more efficient using a multi-resolution approach in (GUO et al 2018), the optimization model and the finite element analysis model are decoupled, super-elements were introduced for structural response and a domain decomposition strategy was adopted to preserve topological complexity of optimized structures. As a result, high-resolution optimization results can be obtained with much less number of degrees of freedoms and design variables.

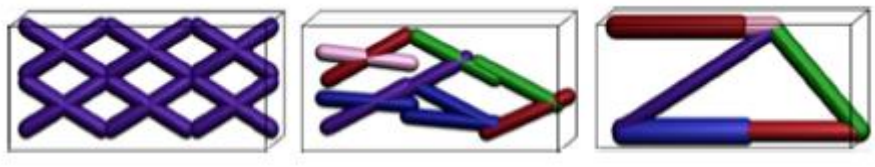
Another explicit topology optimization approach that could be mentioned is the moving morphable void (MMV), it was introduced for 3D problems by (GUO et al. 2017) and appears to reduce the number of design variables and degrees of freedom in finite element analysis. In MMV method, the boundary of the structure is determined by a set of voids controlled through B-splines surfaces control points. Latter stress constraint was integrated in the MMV framework (GUO et al. 2018).

The rest of the report is organized as following: first, we review the MMC method scheme and show the modifications brought to the original code given in (GUO et al. 2015) in order to improve its efficiency. Secondly, we present our new MNA framework based on FE analysis and penalization material law. The optimizer used in both MNA and MMC is the well-known MMA optimizer, a brief review of the method will be presented. Then, extension of MNA and MMC approaches to circular curved components is introduced and a benchmarking with academic test cases is presented. In addition, stress constraint is added to the topology optimization problem, stress tensor computation and constraint formulation are detailed in that section. Besides, in the context of an Airbus project including the PhD thesis of the internship advisor S.CONIGLIO, an industrial test case (originally under SIMP approach) is introduced, and run under MNA and MMC methods. Finally, a discussion about MMC and MNA differences is carried out and conclusions are drawn.

Moving Morphable Components (MMC) framework:

MMC formulation:

The basic idea of MMC method consist on representing any structure by a set of components, in this section we explain the MMC method in 2D case considering rectangular components.



Each component is then characterized by the following design variables:

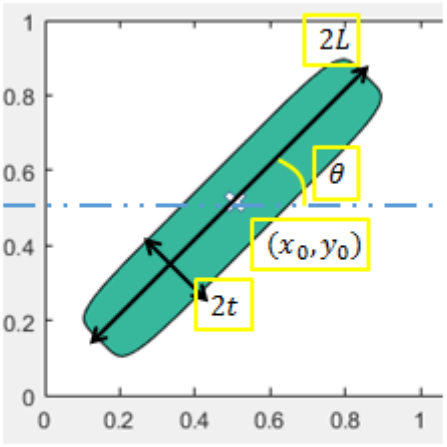
(x_0, y_0) coordinates of the component's center

L half length of the component (along x axis)

t half thickness of the component (along y axis)

θ orientation angle with respect to x axis

Following figure shows a component's geometry and its design variables:



$$X_c = (x_0, y_0, \theta, L, t)$$

Mathematically, the region Ω occupied by one component can be described by the following implicit function:

$$\begin{cases} (x,y) \in \Omega \ \ if \ \ \Phi(x,y) > 0 \\ (x,y) \in \partial\Omega \ \ if \ \ \Phi(x,y) = 0 \\ (x,y) \in D \backslash \Omega \ \ if \ \ \Phi(x,y) < 0 \end{cases}$$

Where: $\partial\Omega$ is the border of Ω, D is the whole design domain and (x, y) are coordinates (in global basis) of a current point. Φ is called topology description function (TDF) and its given by following expression:

$$\Phi(x,y) = 1 - \left(\frac{x_1}{L}\right)^p - \left(\frac{y_1}{t}\right)^p$$

p is a relatively large even integer (here $p = 6$), and (x_1, y_1) are coordinates of the current point in the local basis of the component's center.

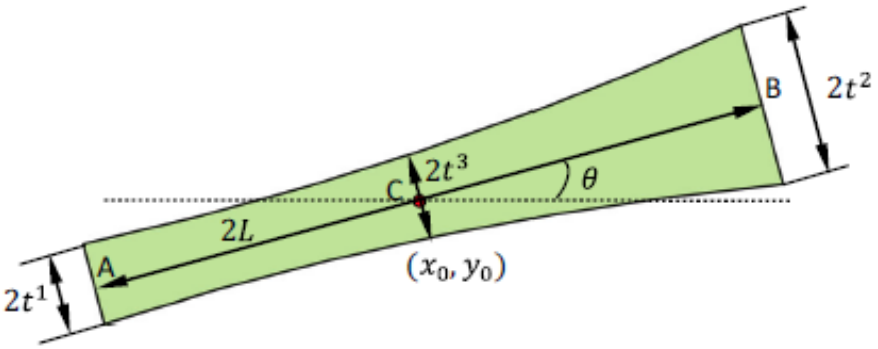
The thickness of component might be variable, in particular, it varies according to a second degree polynomial function in the code given by (GUO et al. 2015) controlled by 3 points: half thickness in the middle and two extremities of the component, thus two additional design variables are used:

$$X_c = (x_0, y_0, \theta, L, t_1, t_2, t_3)^T$$

And TDF is expressed as:

$$\Phi(x,y) = 1 - \left(\frac{x_1}{L}\right)^p - \left(\frac{y_1}{f(x_1)}\right)^p$$

With: $f(x_1) = \frac{t_1+t_2-2t_3}{2L^2}x_1^2 + \frac{t_2-t_1}{2L}x_1 + t_3$, following figures shows a component with quadratically varying thickness:



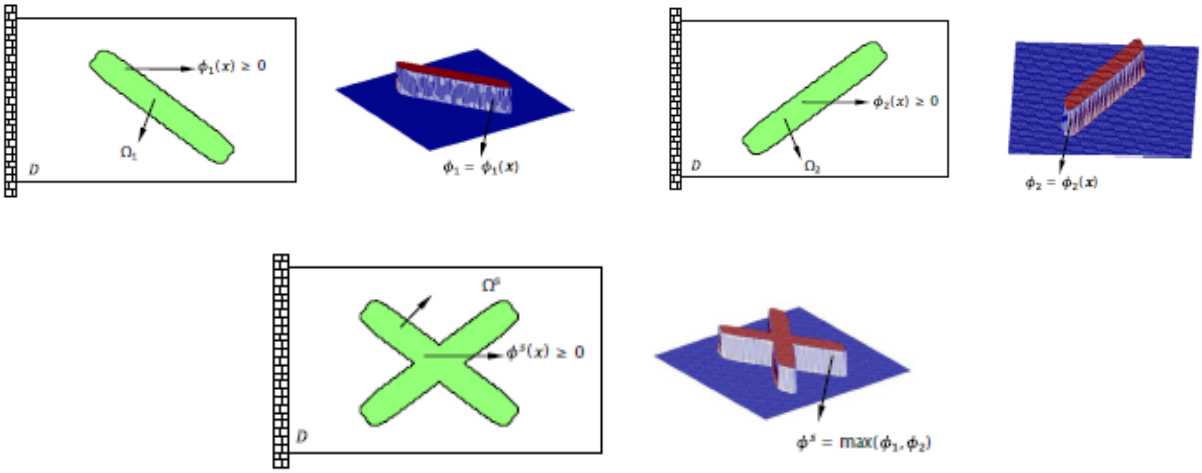
The transformation from global to local coordinates is given by following equation:

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

Note that the value of TDF Φ depends on both current point's coordinates, and design variables through local coordinates.

The structural component can move, dilate/shrink and rotate in the design domain by changing the values of design variables (x_0, y_0, L, t, θ)

The topology of the total structure is achieved via the union of all components consider n components forming the structure, Ω_i is the region occupied by i-th component, then the region occupied by the whole structure would be $\Omega^s = \bigcup_{i=1}^{n} \Omega_i$ figures below illustrates this for 2 components:



The topology is implicitly described as:

$$\begin{cases} (x,y) \in \Omega^s \ \ if \ \ \Phi^s(x,y) > 0 \\ (x,y) \in \partial\Omega^s \ \ if \ \ \Phi^s(x,y) = 0 \\ (x,y) \in D \backslash \Omega^s \ \ if \ \ \Phi^s(x,y) < 0 \end{cases}$$

Where the TDF of the structure Φ^s is expressed as the Boolean union of the TDFs of components using the max operator:

$$\Phi^s(x,y) = \max_{1 \leq i \leq n} \Phi_i(x,y)$$

Optimization problem formulation:

Consider the classic problem of compliance minimization under volume constraint:

$$\begin{cases} \min\limits_X C \\ st\ V(X) \leq V_0 \\ static\ equilibrum \end{cases}$$

Where $X = (X_1, \dots, X_i, \dots, X_n)^T$ is the design vector, and $X_i = (x_{0i}, y_{0i}, L_i, t_i, \theta_i)$ design variables of component i. Eulerian description and fixed finite element mesh on the design domain are used to solve the topology optimization problem. Hence, the problem can be formulated as:

$$Find\ X = (X_1, \dots, X_i, \dots, X_n)^T \quad minimize\ C = F^T U$$

$$st.\ KU = F$$

$$V(X) \leq V_0$$

$$X_{min} \leq X \leq X_{max}$$

Where F is the load vector, U the displacement vector and K the stiffness matrix. V_0 is the volume fraction constraint and V the volume of the structure. X_{min} and X_{max} are vectors containing upper and lower bounds of each design variable.

Four-node quadrilateral elements (with bi-linear shape functions) are used for the mesh, and stiffness elementary matrix is given by:

$$K_{el} = \int_{\Omega_{el}} B^t D B d\Omega$$

Where B is the displacement differentiation matrix, and D is the elasticity matrix, K_{el} is an 8x8 matrix

Since the element density may vary, its stiffness needs to be interpolated. For this purpose an Ersatz material law is used to compute Young modulus of each element:

$$E_e = E \sum_{i=1}^{4} \frac{H(\Phi_i^e)^q}{4}$$

Where Φ_i^e is the value of the structure's TDF Φ on the node i of element e, and H is the regularized Heaviside projection:

$$H(x) = \begin{cases} 1 \ if \ x > \epsilon \\ \frac{3(1-\alpha)}{4}\left(\frac{x}{\epsilon} - \frac{x^3}{3\epsilon}\right) + \frac{1+\alpha}{2} \ if \ -\epsilon \le x \le \epsilon \\ \alpha \ otherwise \end{cases}$$

Here ϵ is a parameter that controls magnitude of regularization and α is a small positive number to ensure the non-singularity of the global stiffness matrix K, in fact the density of void is equal to α^2.

Note that if Φ_i denotes the value of TDF in node i then $H(\Phi_i)$ is the vector of nodal densities.

Finally, the global stiffness matrix is assembled:

$$K = \sum_{e=1}^{n_e} K_e = \sum_{e=1}^{n_e} E_e K_{el,e}$$

Here K_e is the interpolated elementary stiffness matrix, K_e and $K_{el,e}$ are expressed in the global degrees of freedom basis ($n_{dof} \times n_{dof}$ matrices instead of 8×8), that's why $K_{el,e}$ depends on e.

Once stiffness matrix is assembled, displacement vector U is calculated from static equation $KU = F$, and then sensitivities are calculated:

Let x be a design variable, compliance sensitivity is computed as following:

$$\frac{\partial C}{\partial x} = \frac{\partial F^t U}{\partial x} = \frac{\partial F^t}{\partial x} U + F^t \frac{\partial U}{\partial x}$$

For the sake of simplicity we can make the assumption that the load F is not dependent of the design so that $\frac{\partial F^t}{\partial x} = 0$, this is the case in most of the numerical examples presented here, for the others we will see how to deal with this term. Thus:

$$\frac{\partial C}{\partial x} = U^t K^t \frac{\partial U}{\partial x} = U^t K \frac{\partial U}{\partial x}$$

Deriving the static equation, we obtain: $\frac{\partial K}{\partial x} U + K \frac{\partial U}{\partial x} = \frac{\partial F}{\partial x} = 0$, and:

$$\frac{\partial C}{\partial x} = -U^t \frac{\partial K}{\partial x} U = -U^t \sum_{e=1}^{n_e} \frac{\partial E_e}{\partial x} K_{el,e}\, U$$

$$\frac{\partial C}{\partial x} = -U^t \left(\sum_{e=1}^{n_e} \sum_{i=1}^{4} q \frac{\partial H(\Phi_i^e)}{\partial x} \frac{H(\Phi_i^e)^{q-1}}{4} \times E \times K_{el,e} \right) U$$

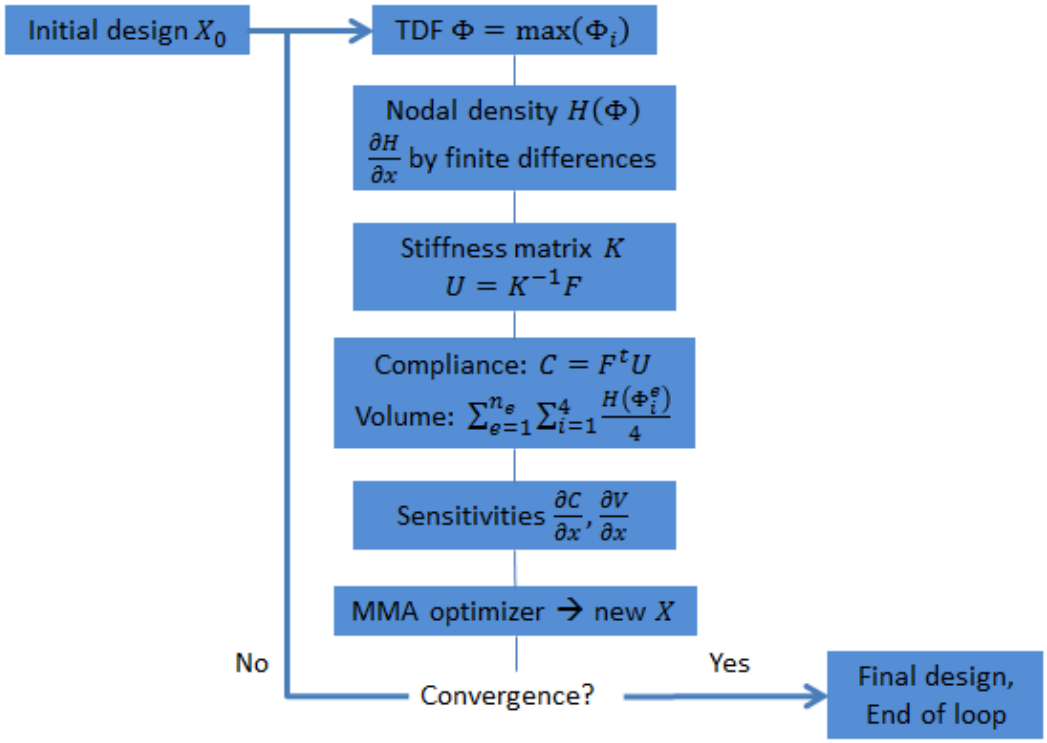
As for the volume sensitivity:

Volume is computed as: $V = \sum_{e=1}^{n_e} \sum_{i=1}^{4} H(\Phi_i^e)$, so:

$$\frac{\partial V}{\partial x} = \sum_{e=1}^{n_e} \sum_{i=1}^{4} \frac{1}{4} \frac{\partial H(\Phi_i^e)}{\partial x}$$

In MMC codes the term $\frac{\partial H}{\partial x}$ is computed using finite differences. Once sensitivities are calculated, they are used as inputs for MMA optimizer in order to update design vector, then stopping criteria are checked, and optimization process ends if satisfied, otherwise a new loop starts.

The following chart shows MMC scheme:



Enhancement of the MMC code:

Many changes have been made on the original code, yet, here we focus on most important ones:

KKT condition:

First, we introduced the KKT condition as a stopping criterion, in fact, the criterion on the change of design vector is not enough since it's a stationarity of the solution while KKT is an optimality condition.

MMA parameters:

In the original code, the MMA parameter *asyincr* was equal to 0.8, this parameter is a factor that determines lower and upper bounds of each design variable in the current MMA iteration, and thus determines the step size from iteration to the next one, a value smaller than 1 means that the step is geometrically decreasing. This is equivalent to enforcing the convergence of the optimization process, and the final solution is not necessary optimal, it just means that the step became so small that the change allowed get smaller than the tolerance. In our code we changed this value to be greater than one (often 1.2).

KS function to approximate max:

In the original code, the TDF of the whole structure is formed using the max function, this function is non-smooth so we used instead lower bound KS function controlled with a parameter P, the smaller is P the smoother is the function, and the bigger is P the more the KS function get close to max. Moreover, the introduction of KS function allows the calculation (if wanted) of analytic sensitivities.

Appropriate normalization of gradients:

In the original code, only gradients of compliance and volume were normalized -divided by their max- while in order to keep the consistency between functions (objective and constraint) and their gradients, functions should be divided by the same value. However for better convergence, volume constraint could be kept non consistent.

Heaviside parameter alpha:

The Heaviside parameter alpha determines the density of void, in fact it's necessary to give a small but non zero value to void density in order to avoid singularity of stiffness matrix. In the original code $alpha = 10^{-3}$ which leads to a density of void equal to $alpha^2 = 10^{-6}$. We noticed that in some cases this value may present difficulties for the code in the beginning, especially if the initial structure is disconnected (doesn't link the load degrees of freedom to boundary conditions). As a result, we change this value to be equal to 10^{-2} for the first iterations and 10^{-3} for the rest of optimization process.

Volume gradient correction:

In the original code the volume constraint was implemented (line 136) as:

```
dfdx(Var_num*k-Var_num+1:Var_num*k,1)=sum(diffH{k})/4;
```

This corresponds to:

$$\frac{\partial V}{\partial x} = \sum_{nodes} \frac{1}{4} \frac{\partial H_n}{\partial x}$$

While the correct expression would be:

$$\frac{\partial V}{\partial x} = \sum_{elements} \frac{\partial \rho_e}{\partial x} = \sum_{elements} \sum_{i=1}^{4} \frac{1}{4} \frac{\partial H_i^e}{\partial x}$$

For confirmation, both gradients were tested by finite differences for several design points and the error of our gradient was less than 30% while the original one was superior to 500%.

Moving Node Approach (MNA) framework:

The moving nodes approach is originally inspired from the work of (Overvelde 2012), where a meshless and flow-inspired method for topology optimization was developed, using the Element-Free Galerkin method (EFG) instead of finite elements method (FEM). The approach we propose here consist on representing the structure with a (finite) number of components which centers are traditionally called mass nodes, each mass node has an influence region, it is the region occupied by the component. Density is equal to one in the mass nodes location, and decreases to zero at the borders of the component. Here rectangular components are used and each component is described by following geometric design variables (same as in MMC method):

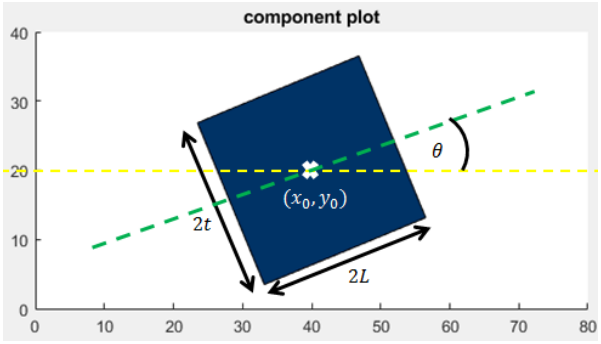
(x_0, y_0) coordinates of the component's center (mass node)

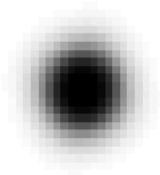
L half length of the component (along x axis)

t half thickness of the component (along y axis)

θ orientation angle with respect to x axis

Left figure shows a component's geometry and its design variables, and right one shows the corresponding density field:





For the sake of simplicity, the MNA formulation is explained in the 2D case, extension to 3D case is easy and natural. A test case in 3D will be presented latter.

The density field generated by one component at a current point of coordinates (x, y) of the design domain is expressed as:

$$\rho_c(x, y) = \omega(\xi) \times \omega(\eta)$$

Where ξ and η are local coordinates of the current point (x, y) in the component's center basis, normalized by thickness and length.

The following equation shows the global to local coordinate's transformation:

$$\begin{pmatrix} \xi \times L \\ \eta \times t \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

ω is the weight function, more the current point is far from the mass node, less is the density value. Here a third order polynomial function is used:

$$\omega(x) = \begin{cases} 1 - 6x^2 + 6x^3 \; if \; x < \frac{1}{2} \\ 2 - 6x + 6x^2 - 2x^3 \; if \; x \geq \frac{1}{2} \end{cases}$$

The density of the structure is obtained after summing the contributions of all the components:

$$\rho_s(x, y) = \sum_{c=1}^{n_c} \rho_c(x, y)$$

With $\rho_s(x, y)$ the density of the structure at current point (x, y) and n_c is the number of components used.

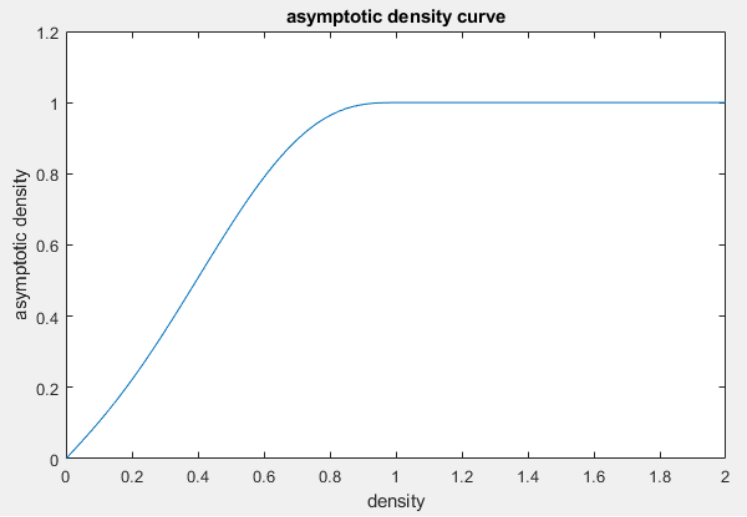
Here we must mention that if two mass nodes are close enough to each other, region with density superior to 1 appears (since we are summing densities). In order to cope with this problem, asymptotic density ρ is introduced; first a polynomial transformation of the density field ρ_s is carried out:

$$\rho = \rho_s + 4\rho_s^3 - 7\rho_s^4 + 3\rho_s^5$$

The purpose of this transformation is that asymptotic density ρ will present a zero asymptote when ρ_s tend to 1, then densities superior to 1 are simply fixed to 1:

$$if \; \rho(x, y) > 1 \; then \;\; \rho(x, y) = 1$$

Following figure shows the variations of asymptotic density ρ with respect to structure's density ρ_s:



The gradient of density needs to be transformed also:

$$\begin{cases} \dfrac{\partial \rho}{\partial x} = (1 + 12{\rho_s}^2 - 28{\rho_s}^3 + 15{\rho_s}^4) \times \dfrac{\partial \rho_s}{\partial x} \\ if \; \rho > 1 \; ; \quad \dfrac{\partial \rho_s}{\partial x} = 0 \end{cases}$$

Note that $\frac{\partial \rho}{\partial x}(\rho_s = 1) = \frac{\partial \rho}{\partial \rho_s}(\rho_s = 1) = 0$

Optimization problem formulation:

The problem is the same as in MMC formulation:

$$\begin{cases} \min\limits_{X} C \\ st\ V(X) \leq V_0 \\ static\ equilibrum \end{cases}$$

Where $X = (X_1, \dots, X_i, \dots, X_n)^T$ is the design vector, and $X_i = (x_{0i}, y_{0i}, L_i, t_i, \theta_i)$ design variables of component i. Here again, Eulerian description and fixed finite element mesh on the design domain are used to solve the topology optimization problem. Hence, the problem can be formulated as:

$$Find\ X = (X_1, \dots, X_i, \dots, X_n)^T \quad minimize\ C = F^T U$$

$$st.\ KU = F$$

$$V(X) \leq V_0$$

$$X_{min} \leq X \leq X_{max}$$

Where F is the load vector, U the displacement vector and K the stiffness matrix. V_0 is the volume fraction constraint and V the volume of the structure. X_{min} and X_{max} are vectors containing upper and lower bounds of each design variable.

Four-node quadrilateral elements (with bi-linear shape functions) are used for the mesh, and stiffness elementary matrix is given by:

$$K_{el} = \int_{\Omega_{el}} B^t D B d\Omega$$

Where B is the displacement differentiation matrix, and D is the elasticity matrix, K_{el} is an 8x8 matrix

For FE analysis, a penalization law is used to calculate the Young modulus of each element:

$$E_e = E_{min} + (E_0 - E_{min})\rho^{pen}$$

With $E_{min} = 10^{-9}$ defines the density of void, it is not equal to zero so that stiffness matrix K is not singular. E_0 is the Young modulus of material (here $E_0 = 1$). $pen = 3$ is the penalization power, this is the same law used in the SIMP approach.

The stiffness of each element is $K_e = E_e K_{el,e}$, and the global stiffness matrix is assembled:

$$K = \sum_{e=1}^{n_e} K_e = \sum_{e=1}^{n_e} E_e K_{el,e}$$

It is worth noting that K_e and $K_{el,e}$ are expressed in the global degrees of freedom basis ($n_{dof} \times n_{dof}$ matrices instead of 8×8), that's why $K_{el,e}$ depends on e.

Once stiffness matrix is assembled, displacement vector U is calculated from static equation $KU = F$, and sensitivities are calculated:

Let x be a design variable, compliance sensitivity is computed as following:

$$\frac{\partial C}{\partial x} = \frac{\partial F^t U}{\partial x} = \frac{\partial F^t}{\partial x} U + F^t \frac{\partial U}{\partial x}$$

For the sake of simplicity we consider the load F independent from the design so that $\frac{\partial F^t}{\partial x} = 0$, thus:

$$\frac{\partial C}{\partial x} = U^t K^t \frac{\partial U}{\partial x} = U^t K \frac{\partial U}{\partial x}$$

Deriving the static equation, we obtain: $\frac{\partial K}{\partial x} U + K \frac{\partial U}{\partial x} = \frac{\partial F}{\partial x} = 0$, and:

$$\frac{\partial C}{\partial x} = -U^t \frac{\partial K}{\partial x} U = -U^t \sum_{e=1}^{n_e} \frac{\partial E_e}{\partial x} K_{el,e} U$$

Deriving the penalization law:

$$\frac{\partial E_e}{\partial x} = pen(E_0 - E_{min}) \rho^{pen-1} \frac{\partial \rho}{\partial x}$$

Then the term $\frac{\partial \rho}{\partial x}$ is expressed using asymptotic density derivation:

$$\begin{cases} \frac{\partial \rho}{\partial x} = (1 + 12{\rho_s}^2 - 28{\rho_s}^3 + 15{\rho_s}^4) \times \frac{\partial \rho_s}{\partial x} \\ if\ \rho > 1\ ; \quad \frac{\partial \rho_s}{\partial x} = 0 \end{cases}$$

$$\text{and } \frac{\partial \rho_s}{\partial x} = \sum_{c=1}^{n_c} \frac{\partial \rho_c}{\partial x} \text{ with } \rho_c = \omega(\xi)\omega(\eta)$$

$$\text{weight functions } \frac{\partial \rho_c}{\partial x} = \frac{\partial \omega(\xi)}{\partial x} \omega(\eta) + \frac{\partial \omega(\eta)}{\partial x} \omega(\xi)$$

$$\text{using chain rule: } \frac{\partial \rho_c}{\partial x} = \frac{\partial \omega}{\partial \xi} \frac{\partial \xi}{\partial x} \omega(\eta) + \frac{\partial \omega}{\partial \eta} \frac{\partial \eta}{\partial x} \omega(\xi)$$

$$\text{here } \frac{\partial \omega}{\partial \xi} = \frac{\partial \omega}{\partial \eta} = \omega'(x) = \begin{cases} -12x + 18x^2\ if\ x < \frac{1}{2} \\ -6 + 12x - 6x^2\ if\ x \geq \frac{1}{2} \end{cases}$$

The remaining terms $\frac{\partial \xi}{\partial x_i}$ and $\frac{\partial \eta}{\partial x_i}$ are calculated using the transformation from global to local coordinates:

$$\begin{pmatrix} \xi \times L \\ \eta \times t \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

This is a trivial derivation.

The volume of the structure is $V(X) = \int_\Omega \rho d\Omega$, so $\frac{\partial V(X)}{\partial x} = \int_\Omega \frac{\partial \rho}{\partial x} d\Omega$

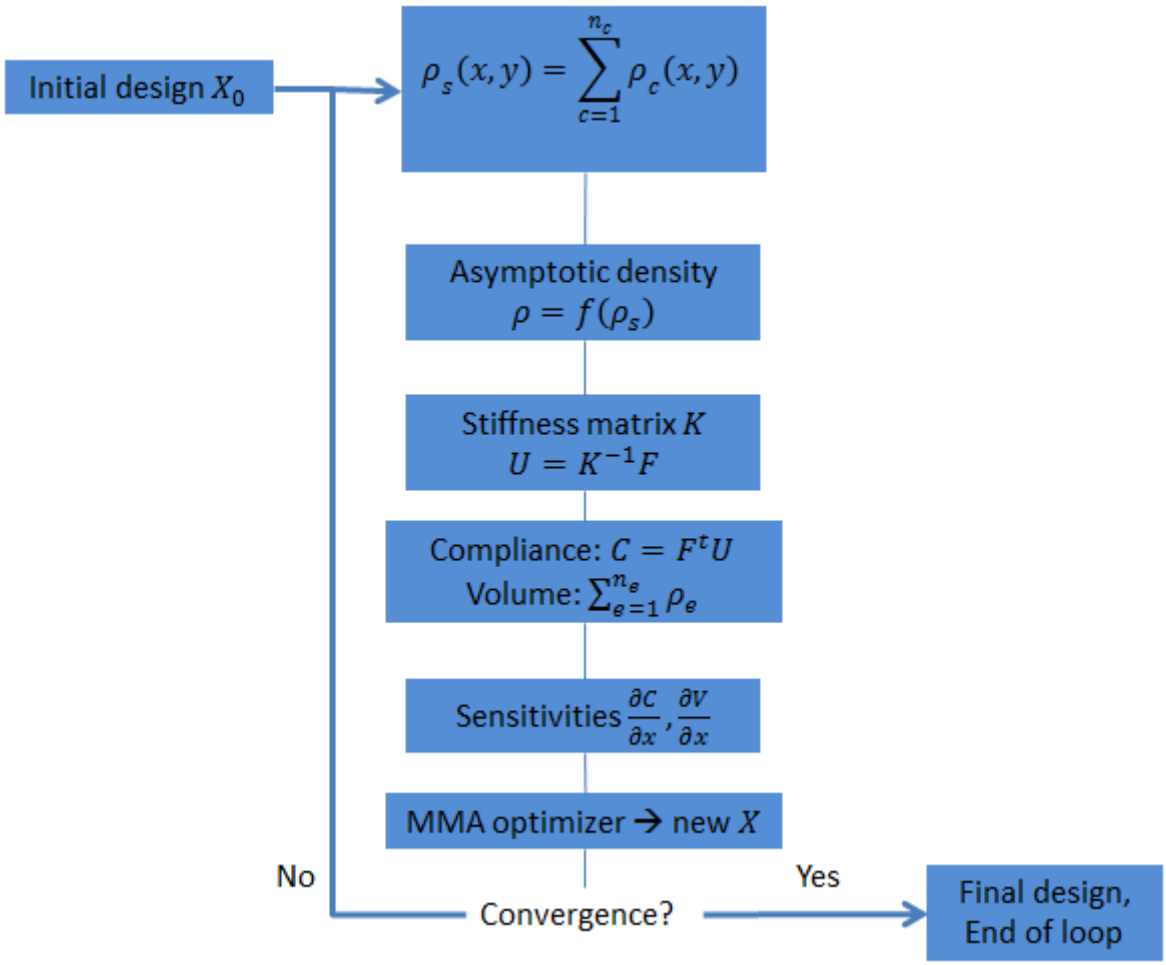
In the discretized form:

$$V(X) = \sum_{e=1}^{n_e} \rho_e \ \text{, where } \rho_e \text{ is the density of element } e$$

$$and \quad \frac{\partial V(X)}{\partial x} = \sum_{e=1}^{n_e} \frac{\partial \rho_e}{\partial x}$$

It is worth noting that sensitivities in MNA are calculated analytically, while in MMC approach a semi-analytical formula was used.

After sensitivities being computed, the next step is to use them as inputs for MMA optimizer in order to update design vector X, then stopping criteria are checked, and optimization process ends if satisfied, otherwise a new loop starts. The following chart summarizes the MNA scheme:



Method of Moving Asymptotes (MMA) optimizer:

The MMA optimizer is used to solve the topology optimization problem, as many non-linear problems, the topology optimization problem can have the following "standard" form:

$$minimize\ f_0(\mathbf{x})$$

$$subject\ to\ f_i(\mathbf{x}) \leq 0, \quad 1 \leq i \leq m$$

$$x_j^{min} \leq x_j \leq x_j^{max}, \quad 1 \leq j \leq n$$

Yet, the problem is not solved under this form in MMA, because it may not have any feasible solution. Indeed the form solved by MMA optimizer is written as following:

$$minimize \quad f_0(\mathbf{x}) + a_0 z + \sum_{i=1}^{m} c_i y_i + \frac{1}{2} d_i y_i^2$$

$$subject\ to \quad f_i(\mathbf{x}) - a_i z - y_i \leq 0, \quad 1 \leq i \leq m$$

$$x_j^{min} \leq x_j \leq x_j^{max}, \quad 1 \leq j \leq n$$

$$y_i \geq 0 \qquad 1 \leq i \leq m$$

$$z \geq 0$$

Where $\mathbf{x} = (x_1, \dots, x_n)^T$, $\mathbf{y} = (y_1, \dots, y_m)^T$ and z are design variables, $x_1, \dots, x_n$ are the "true" optimization variables (since they are the only ones from the "standard" form), while $y_1, \dots, y_m$ and z are "artificial" optimization variables (doesn't exist in the "standard" form). $f_0, f_1, \dots, f_m$ are given, continuously differentiable, real-valued functions. x_j^{min} and x_j^{max} are given real numbers which satisfy $x_j^{min} < x_j^{max}$. a_0 and a_i are given real numbers which satisfy $a_0 > 0$ and $a_i \geq 0$. c_i and d_i are given real numbers which satisfy $c_i \geq 0, d_i \geq 0$ and $c_i + d_i > 0$.

This form could be seen as a generalization of the first form where $a_0 = 1$ and $\forall\ 1 \leq i \leq m\ ,\ a_i = 0$. Then $z = 0$ in any optimal solution. Further, for each i, let $d_i = 0$ and c_i equal to a "large" number, so that the variables y_i become "expensive". Then typically $\mathbf{y} = 0$ in any optimal solution. And the corresponding x is solution of the first (standard) form.

It is worth noting that the form solved in MMA always has feasible solutions, and in fact at least one optimal solution. This holds even if the standard form doesn't have any feasible solution, in which case some $y_i > 0$ in the optimal solution of MMA form.

Once the optimization problem to be solved is made under the previous form, MMA use the following approach to solve it:

In each iteration, the current iteration design vector is given $\left(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}, z^{(k)}\right)$. Then an approximating explicit subproblem is generated. In this subproblem, the functions $f_i(x)$ are replaced by approximating convex functions $\tilde{f}_i^{(k)}(\mathbf{x})$. These approximations are based mainly on gradient information at the current iteration point, but also (implicitly) on information from previous iteration points. The subproblem is solved and the unique optimal solution becomes the next iteration point $\left(\mathbf{x}^{(k+1)}, \mathbf{y}^{(k+1)}, z^{(k+1)}\right)$. Then, a new subproblem is generated, etc. the subproblem mentioned above looks as follows:

$$minimize \quad \tilde{f}_0^{(k)}(\mathbf{x}) + a_0 z + \sum_{i=1}^{m} c_i y_i + \frac{1}{2} d_i y_i^2$$

$$subject\ to \quad \tilde{f}_i^{(k)}(\mathbf{x}) - a_i z - y_i \le 0, \quad 1 \le i \le m$$

$$\alpha_j^{(k)} \le x_j \le \beta_j^{(k)}, \quad 1 \le j \le n$$

$$y_i \ge 0 \qquad 1 \le i \le m$$

$$z \ge 0$$

The approximating functions $\tilde{f}_i^{(k)}(\mathbf{x})$ are chosen as:

$$\forall\ 0 \le i \le m\,, \qquad \tilde{f}_i^{(k)}(\mathbf{x}) = \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{u_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - l_j^{(k)}} \right) + r_i^{(k)}$$

Where:

$$p_{ij}^{(k)} = \left(u_j^{(k)} - x_j^{(k)}\right)^2 \left(\left(\frac{\partial f_i}{\partial x_j}\left(\mathbf{x}^{(k)}\right) \right)^{+} + \kappa_{ij}^{(k)} \right)$$

$$q_{ij}^{(k)} = \left(x_j^{(k)} - l_j^{(k)}\right)^2 \left(\left(\frac{\partial f_i}{\partial x_j}\left(\mathbf{x}^{(k)}\right) \right)^{-} + \kappa_{ij}^{(k)} \right)$$

$$r_i^{(k)} = f_i\left(\mathbf{x}^{(k)}\right) - \sum_{j=1}^{n} \left(\frac{p_{ij}^{(k)}}{u_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - l_j^{(k)}} \right)$$

$$\alpha_j^{(k)} = \max\left\{x_j^{min}, 0.9 l_j^{(k)} + 0.1 x_j^{(k)}\right\}$$

$$\beta_j^{(k)} = \min\left\{x_j^{max}, 0.9 u_j^{(k)} + 0.1 x_j^{(k)}\right\}$$

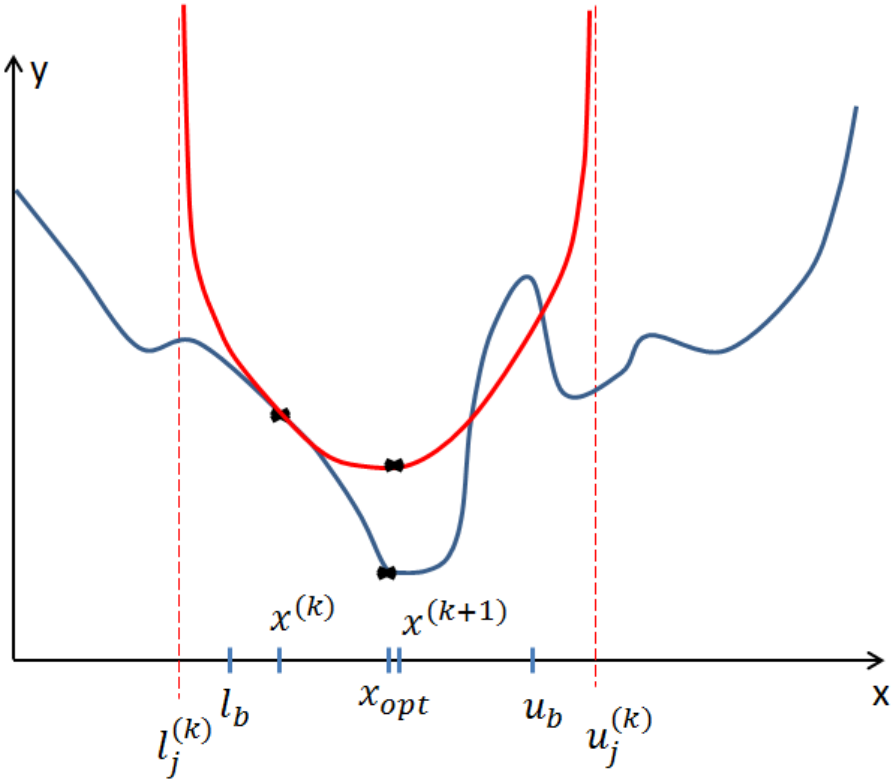
$$\left(\frac{\partial f_i}{\partial x_j}\left(\mathbf{x}^{(k)}\right) \right)^{+} = \max\left\{0, \frac{\partial f_i}{\partial x_j}\left(\boldsymbol{x}^{(k)}\right)\right\} \ and \ \left(\frac{\partial f_i}{\partial x_j}\left(\mathbf{x}^{(k)}\right) \right)^{-} = \max\left\{0, -\frac{\partial f_i}{\partial x_j}\left(\boldsymbol{x}^{(k)}\right)\right\}$$

This implies that all the approximating functions $\tilde{f}_i^{(k)}$ are strictly convex, which in turn implies that there is always a unique optimal solution of the MMA subproblem.

Regardless of the values of the parameters $\kappa_{ij}^{(k)}$, the functions $\tilde{f}_i^{(k)}$ are always first order approximations of the original functions f_i at the current iteration design point:

$$\tilde{f}_i^{(k)}\left(\mathbf{x}^{(k)}\right) = f_i\left(\mathbf{x}^{(k)}\right) \ \ and \ \ \frac{\partial \tilde{f}_i^{(k)}}{\partial x_j}\left(\mathbf{x}^{(k)}\right) = \frac{\partial f_i}{\partial x_j}\left(\mathbf{x}^{(k)}\right)$$

Following figure shows how a nonlinear objective function in 1D is approximated by convex function and next iteration point is the solution of the subproblem:



Curve in blue corresponds to the nonlinear function to be approximated, while the red curve corresponds to the convex MMA approximation function.

$l_j^{(k)}$ and $u_j^{(k)}$ are the lower and upper asymptotes respectively

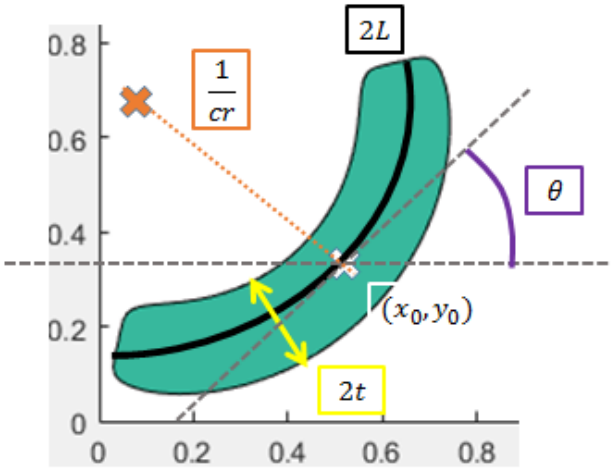
$x^{(k)}$ and $x^{(k+1)}$ are design points of iteration k and $k+1$ respectively.

l_b and u_b are the lower and upper bounds of next iteration point: $l_b \leq x^{(k+1)} \leq u_b$

x_{opt} is the true optimum (of the objective function)

Extension to curved components:

Here we propose the curvature as new feature for components, in order to obtain such components, a new design variable had been introduced (for each component): the curvature cr, it is the inverse of the curvature radius of the component's mid-line. The choice of curvature rather than radius can be justified by following points: First, the radius of a straight component is infinite, as a result a straight component cannot be described by such a design variable, while we want to keep a general description; curved components should generalize straight ones. Second, the curvature variable cr allows a continuous transition from curved to straight components ($cr = 0$). The following picture shows a curved component design variables:



Hence, the topology description function (TDF) needs to be adapted for curved components; the idea is that first, the global to local (center of component) basis is carried out (same as for straight components):

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x - x_0 \\ y - y_0 \end{pmatrix}$$

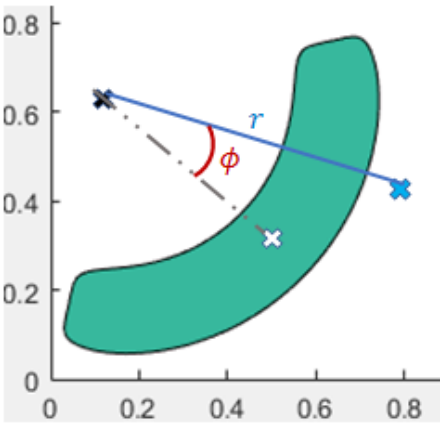
Then, if the component is curved $cr \neq 0$ a translation from the center of component to the center of curvature basis is carried out:

$$x_2 = sign(cr)x_1 \quad and \quad y_2 = sign(cr)\left(y_1 + \frac{1}{cr}\right)$$

And Cartesian coordinates (x_2, y_2) are transformed to polar coordinates (r, ϕ) via following relations:

$$r = \sqrt{x_2^2 + y_2^2} \quad and \quad \phi = atan2(y_2, x_2)$$

Polar coordinates of a current point are shown in the following figure:



The TDF is now written as:

$$\Phi(x, y) = 1 - \left(\frac{\phi}{cr \times L}\right)^p - \left(\frac{r - \frac{1}{|cr|}}{t}\right)^p$$

For sensitivity analysis, the only change is that a new term $\frac{\partial H}{\partial cr}$ needs to be evaluated by finite differences in order to obtain $\frac{\partial C}{\partial cr}$ and $\frac{\partial V}{\partial cr}$.

That was for MMC method. Curved components can be introduced for MNA method also.

In MNA framework, the local coordinates are now written as:

$$\xi = \frac{r - \frac{1}{|cr|}}{t} \quad and \quad \eta = \frac{\phi}{cr \times L}$$

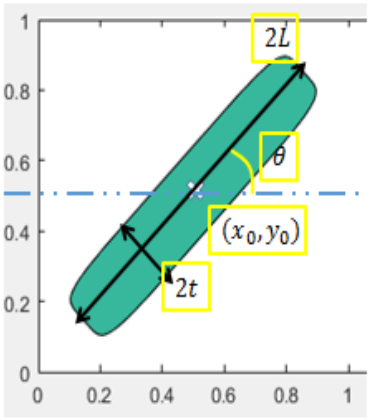
However calculating sensitivities analytically (with curved components) is problematic since the polar coordinates (r, ϕ) transformation is defined by pieces (depending if $cr = 0$ or not), moreover the function $(x, y) \longrightarrow atan2$ is differentiable only if $x > 0$ or $y \neq 0$. As a result the gradient of density was calculated using finite differences.

Curvature is sought to be a generalization of straight components, which means that if the optimal design is obtained for straight components (no need for curvature) then the code with curvature is able to reach the same optimal design. However it can improve designs especially when few components are used.

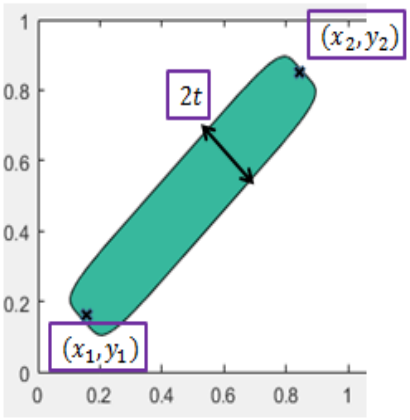
New equivalent design variables:

We implemented new design variables on MMC framework, instead of describing a component's topology by the coordinates of its center, orientation angle, length and width; we describe it by coordinates of the two extremities and its width. The interest of new variables was to obtain the same range (Xmin and Xmax) for all design variables. The following figure shows the old and new design variables:

Old variables



New variables



$$X_{old} = (x_0, y_0, \theta, L, t)$$

$$X_{new} = (x_1, y_1, x_2, y_2, t)$$

However, even if new variables are used, it's still necessary to transform them to old formulation before computing the TDF of each component. Following equations show the transformation from new to old and old to new variables:

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ t \end{bmatrix} = \begin{bmatrix} x_0 - \frac{L\cos(\theta)}{2} \\ y_0 - \frac{L\sin(\theta)}{2} \\ x_0 + \frac{L\cos(\theta)}{2} \\ y_0 - \frac{L\sin(\theta)}{2} \\ t \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_0 \\ y_0 \\ \theta \\ L \\ t \end{bmatrix} = \begin{bmatrix} \frac{x_1 + x_2}{2} \\ \frac{y_1 + y_2}{2} \\ \text{atan2}(y_2 - y_1, x_2 - x_1) \\ \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\ t \end{bmatrix}$$

The merit of the new design variables is that all design variables vary in the same range, which is better for the optimizer, although it is also possible by normalizing each variable of the design vector:

$$\forall\ 1 \le i \le n_{var} \quad \mathbf{x}(i) = \frac{\boldsymbol{X}(i) - \boldsymbol{X}_{min}(i)}{\boldsymbol{X}_{max}(i) - \boldsymbol{X}_{min}(i)}$$

Where $\boldsymbol{X}$ is the design vector and $\mathbf{x}$ is the normalized design vector

Also, it avoids the issue of periodicity of the angle θ. However, old and new design variables were tested for several examples and appear to give same results. Therefore old design variables are used in what follows.

Benchmarking of methods:

In this section, numerical examples are given under different codes (MNA, MMC, SIMP …), so that our framework under MNA and MMC is compared to existing methods, mainly:

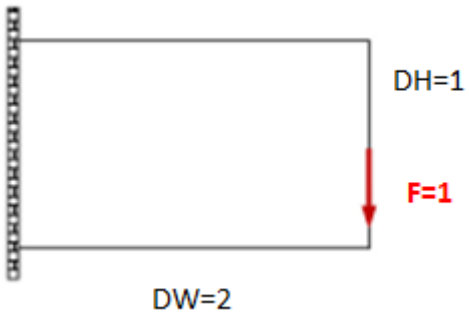
- original MMC code
- our enhanced MMC code
- MMC code with curvature
- MNA code with straight components
- MNA code with curvature
- SIMP code (top88 with MMA optimizer)

1) Cantilever beam

The design domain is a rectangle of length 2 and height 1 discretized by a 80x40 mesh, the topology optimization problem is formulated as:

$$\begin{cases} \min\limits_{X} C \\ st\ V(X) < volfrac \end{cases}$$

Geometry, load and boundary conditions are shown in the following figure:

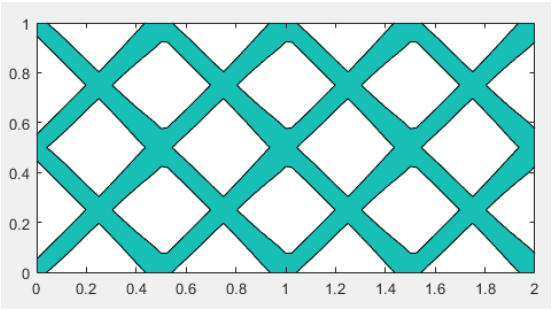


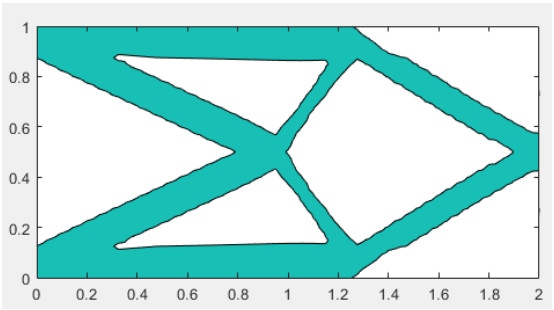
Following tables shows some parameters for the simulation:

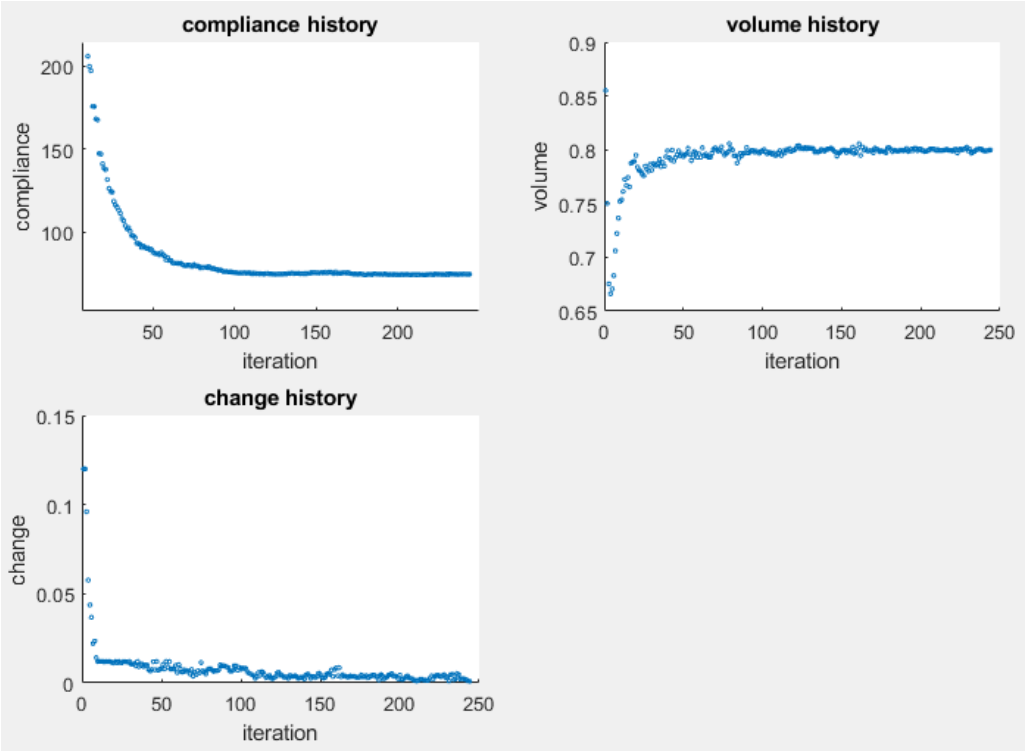
Mesh	80x40
Volfrac	0.4
Change tolerance	0.001
KKT tolerance	0.01
Maximum number of iterations	600

Original MMC code results:

Initial configuration, final design and convergence history are shows in following figures:





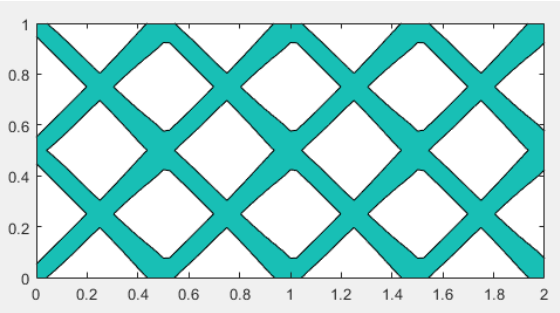


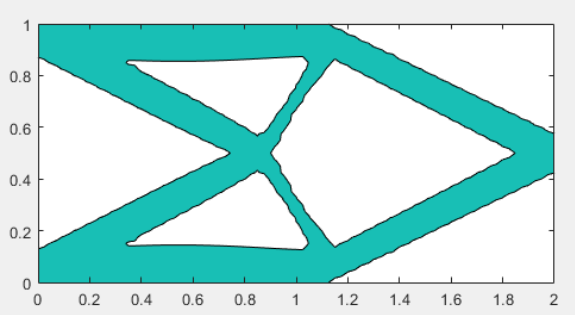
Following table summarizes the simulation results:

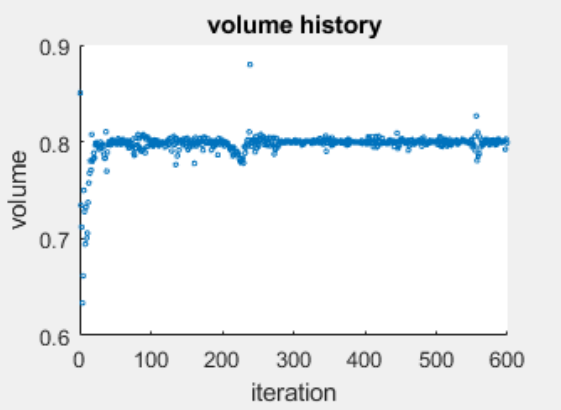
Final compliance	74.68
Number of components	8*4=32
Number of design variables	32*7=224
Stopping criterion	Change satisfied after 244 iteration
Total CPU time	54.62s
Average CPU time (per iteration)	0.224s

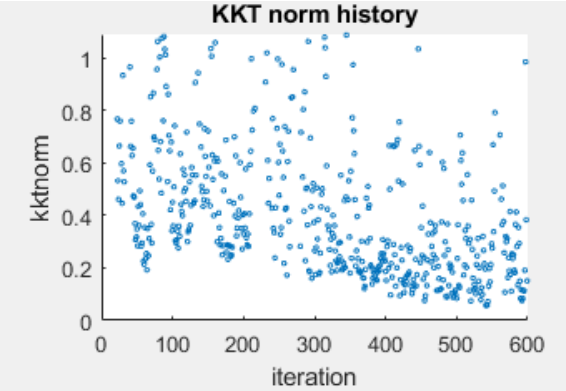
Enhanced MMC code results:

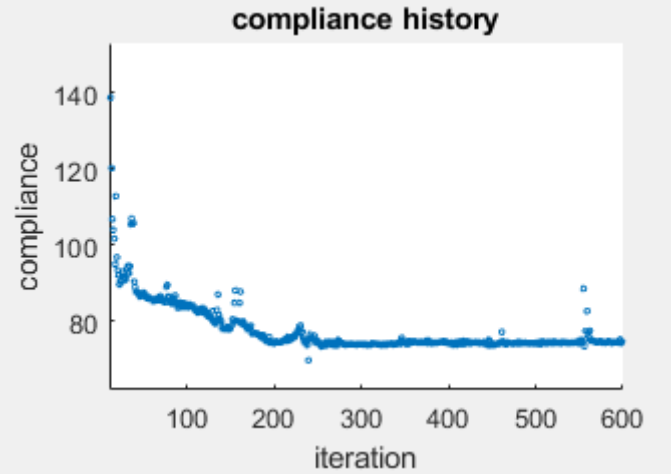
Initial configuration, final design and convergence history are shows in following figures:











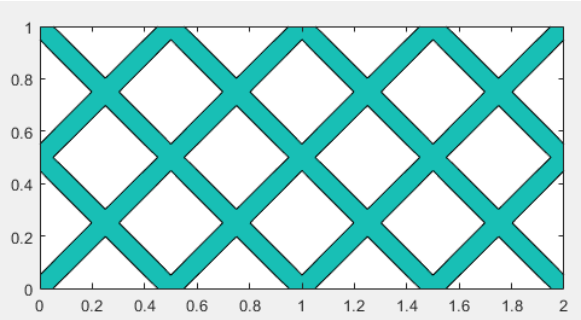
Following table summarizes the simulation results:

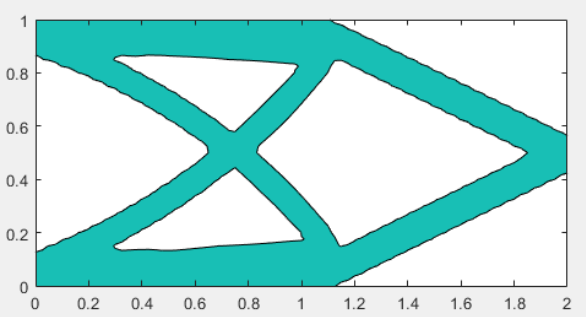
Final compliance	73.9

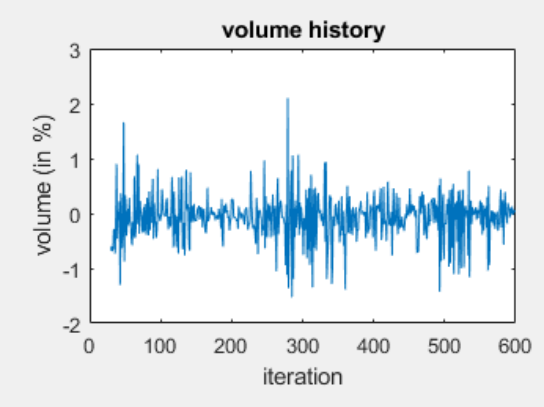
Number of components	8*4=32
Number of design variables	32*7=224
Stopping criterion	Max number of iteration reached
Total CPU time	175.7s
Average CPU time (per iteration)	0.29s

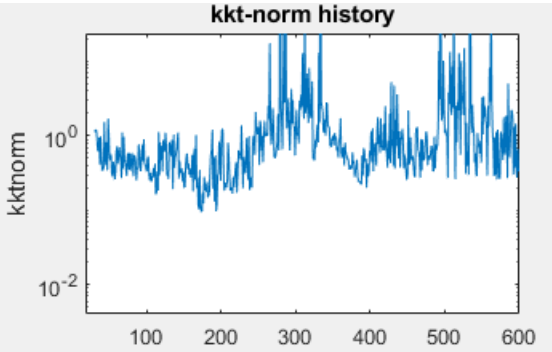
MMC with curvature results:

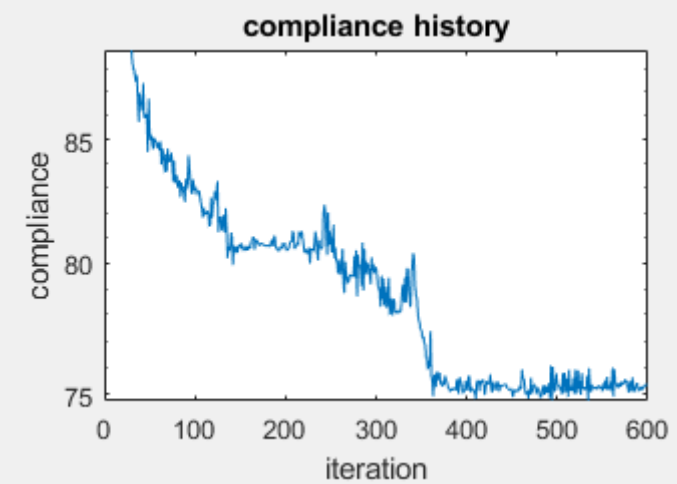
Initial configuration, final design and convergence history are shows in following figures:











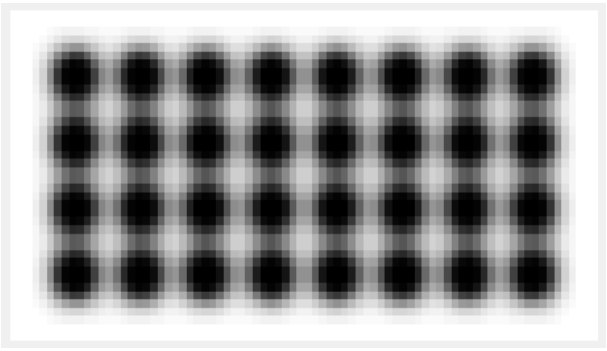
Following table summarizes the simulation results:

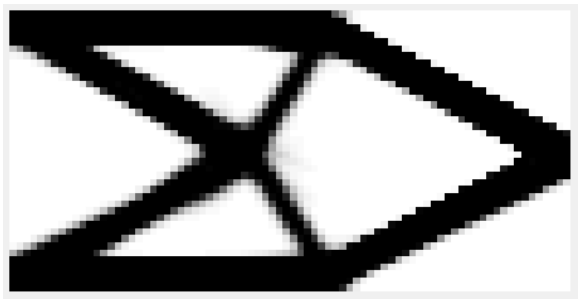
Final compliance	75.08
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	Max number of iteration reached

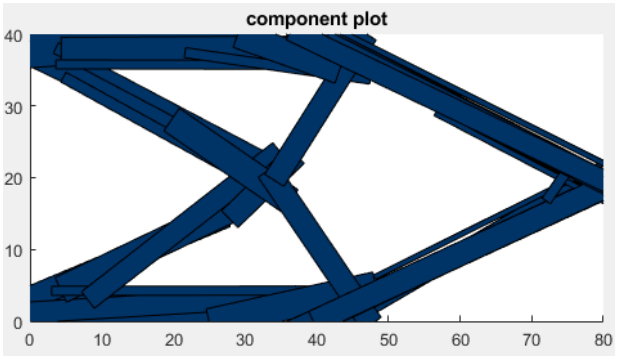
Total CPU time	136.9s
Average CPU time (per iteration)	0.228s

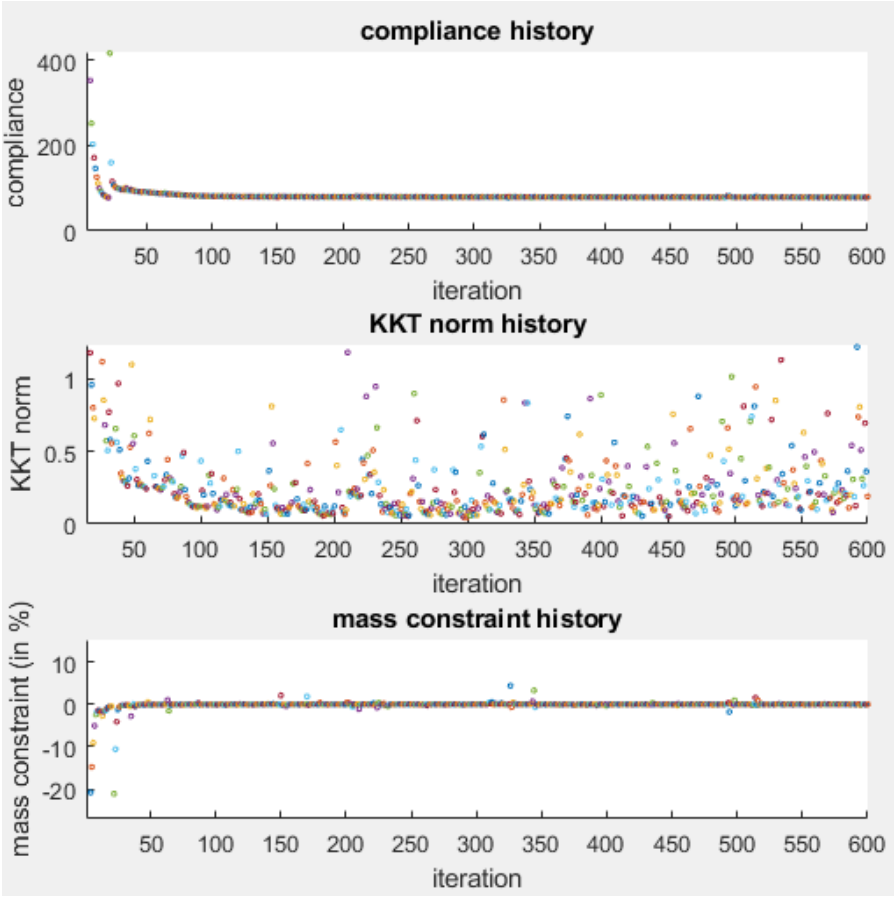
MNA code results:

Initial configuration, final design and convergence history are shows in following figures:







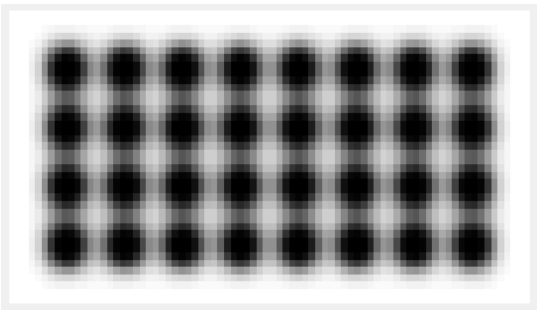


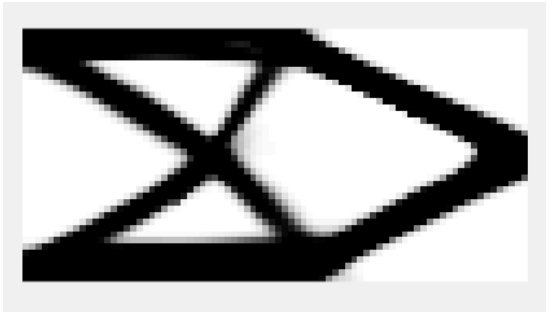
Following table summarizes the simulation results:

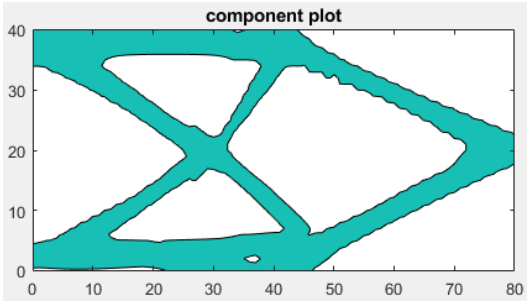
Final compliance	78
Number of components	8*4=32
Number of design variables	32*5=160
Stopping criterion	Max number of iteration reached
Total CPU time	183.4s
Average CPU time (per iteration)	0.3s

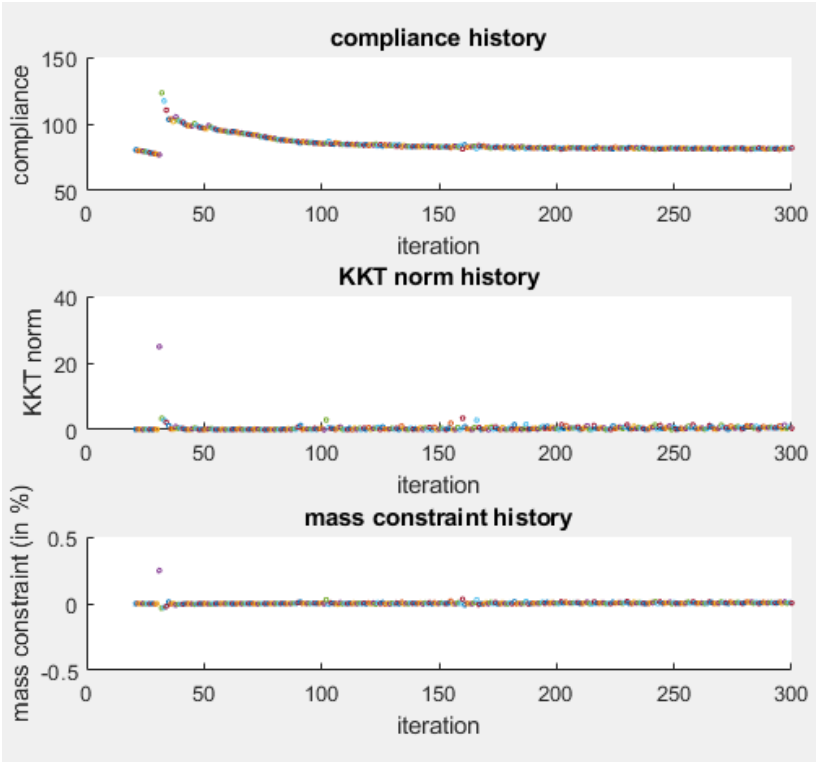
MNA with curvature results:

Initial configuration, final design and convergence history are shows in following figures:







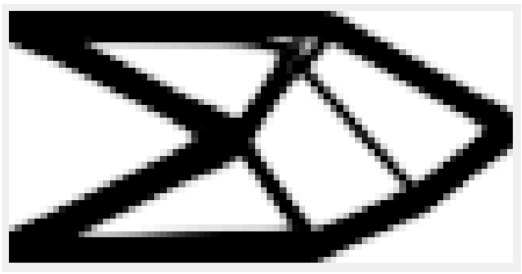


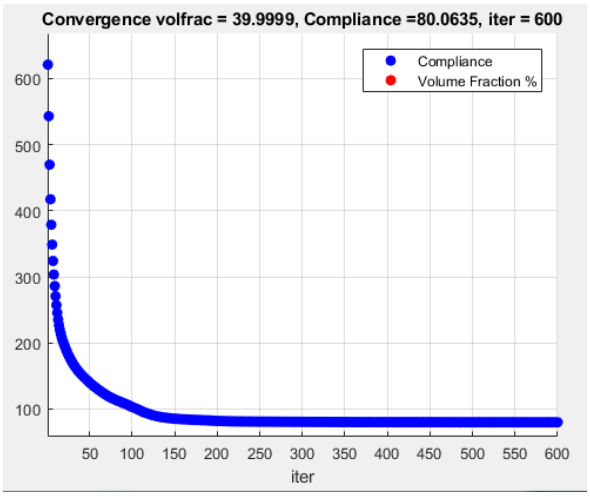
Following table summarizes the simulation results:

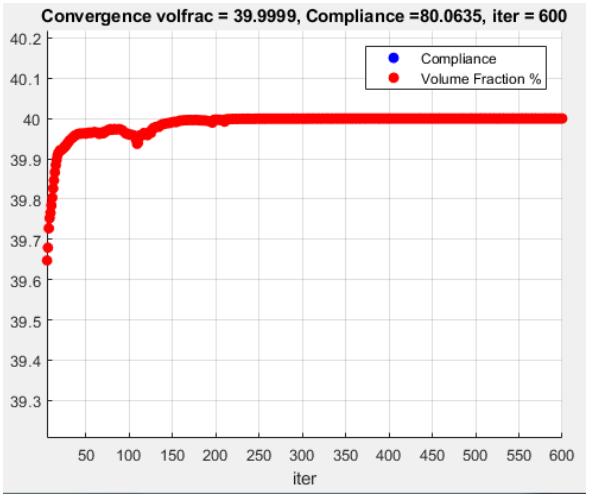
Final compliance	81.9
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	Max number of iteration reached (300)
Total CPU time	737.12s
Average CPU time (per iteration)	2.45s

SIMP (top88 with MMA) results:

Initial configuration, final design and convergence history are shows in following figures:







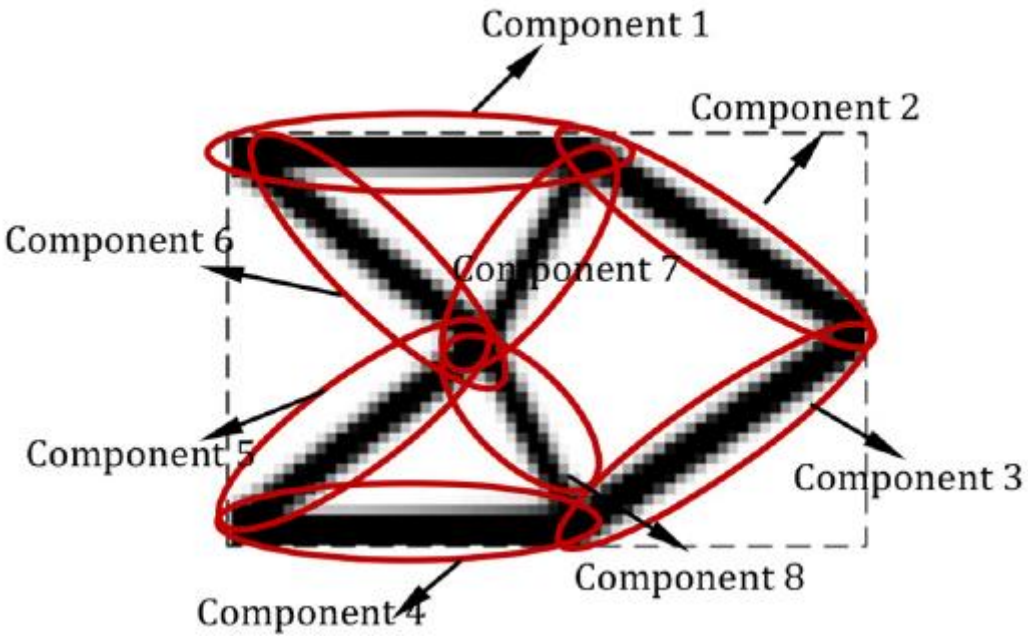
Following table summarizes the simulation results:

Final compliance	80.6
Number of components	
Number of design variables	80*40=3200
Stopping criterion	Max number of iteration reached
Total CPU time	223.85s
Average CPU time (per iteration)	0.37s

The best compliance was obtained by the enhanced MMC code, both MMC and MNA codes manage to find a design that corresponds to the well know global optimum of Cantilever beam test case, even though compliance was a little bit higher for MNA. The slightly higher compliance by MNA (although geometry of the design is the same) is due to the presence of gray regions (intermediate densities). However, as will be seen latter, these gray regions have the merit of facilitating the

convergence, especially for more complex problems, when a black and white formulation (with almost no gray regions) such as MMC is presenting hard convergence.

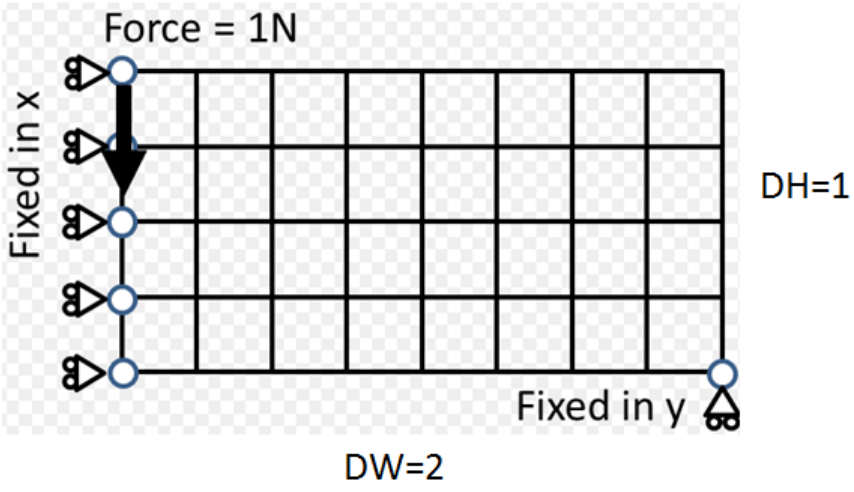
It is worth noting that the optimal solution can be considered as the union of 8 components (beams) as shown in figure below:



However, using curved components allows reducing this number to only 6 components, with 4 straight and 2 curved as shown in the design of MMC code with curvature.

2) MBB beam

Design domain is a rectangle of length 4 and height 1, simply supported from its sides and load is applied in the middle of the upper edge, because of the symmetry of the problem, only half domain is discretized by a 80x40 mesh, and symmetry conditions are applied on the left edge, following figure shows geometry, load and boundary conditions:

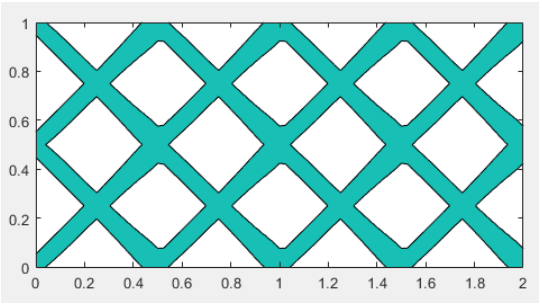


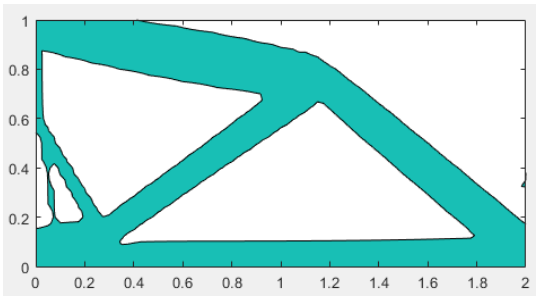
Following tables shows some parameters for the simulation:

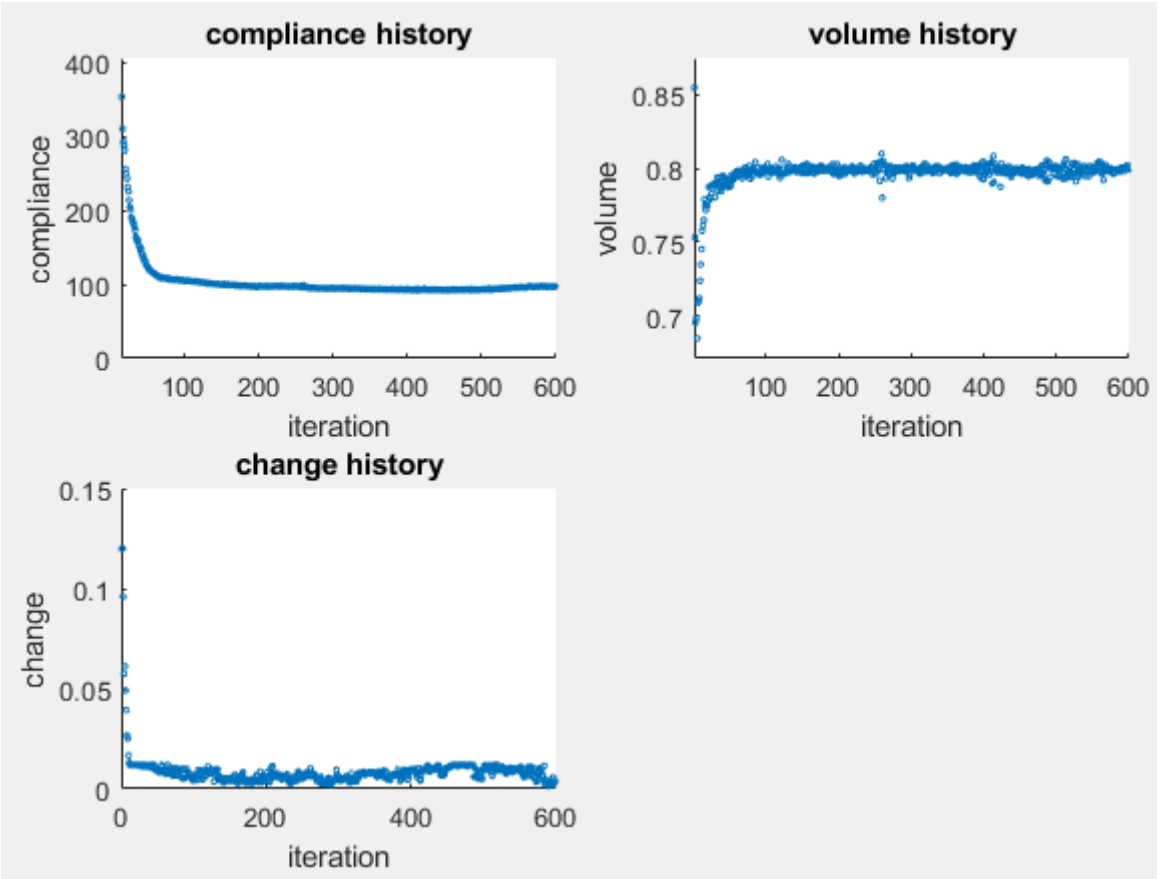
Mesh	80x40
Volfrac	0.4
Change tolerance	0.001
KKT tolerance	0.01
Maximum number of iterations	600

Original MMC code results:

Initial configuration, final design and convergence history are shows in following figures:







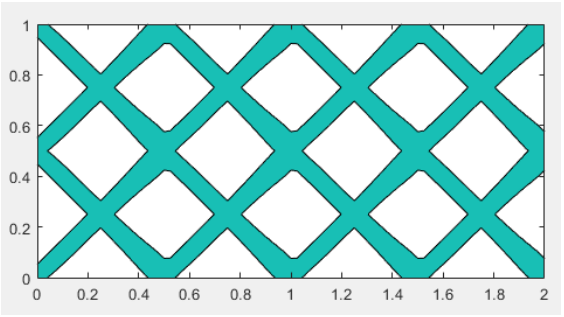
Following table summarizes the simulation results:

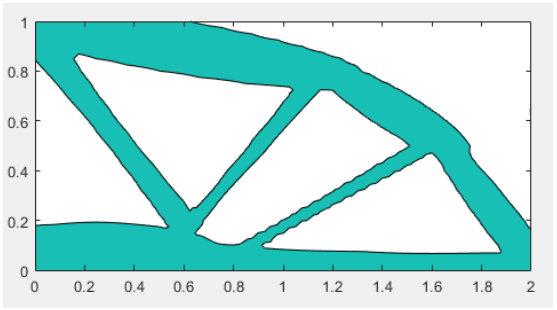
Final compliance	97
Number of components	8*4=32
Number of design variables	32*7=224
Stopping criterion	Max number of iterations reached
Total CPU time	145.17s

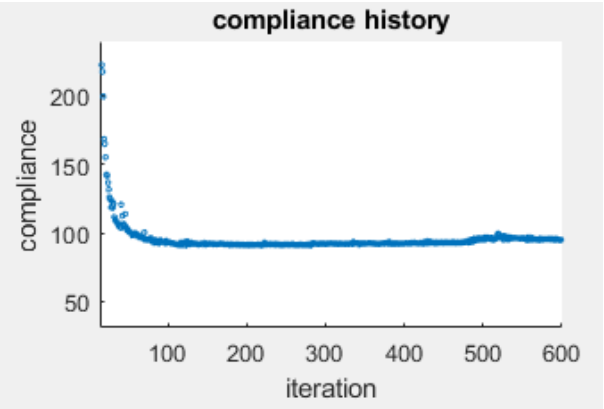
Average CPU time (per iteration)	0.242s

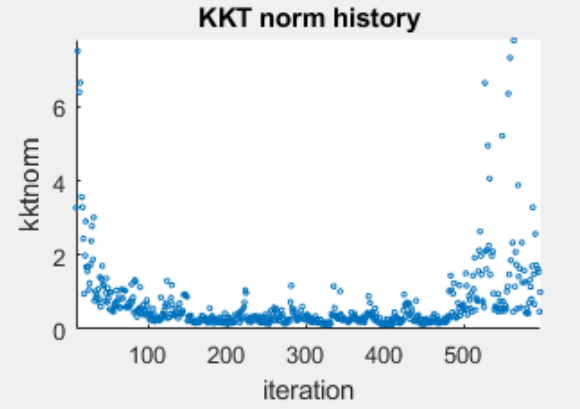
Enhanced MMC code results:

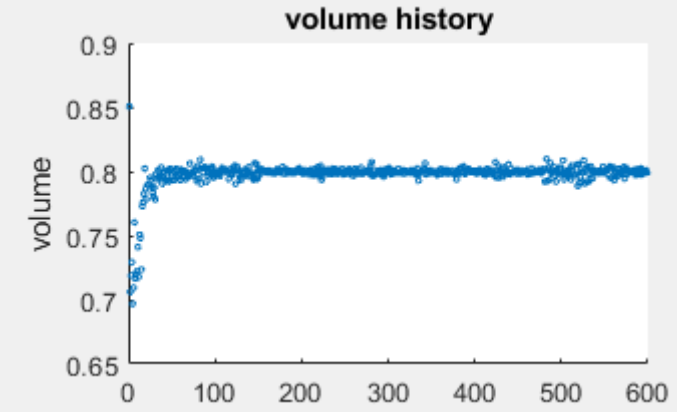
Initial configuration, final design and convergence history are shows in following figures:









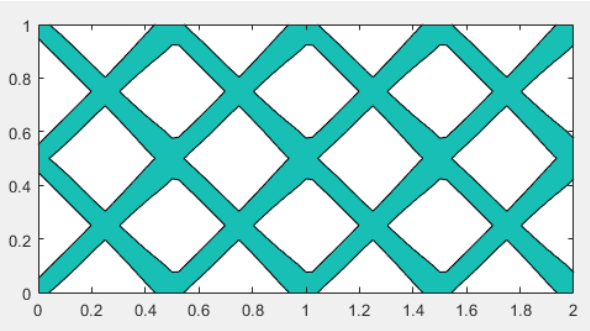


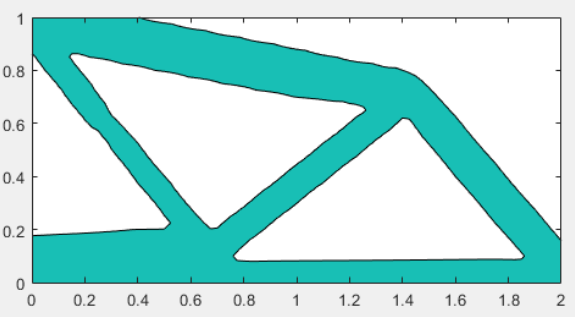
Following table summarizes the simulation results:

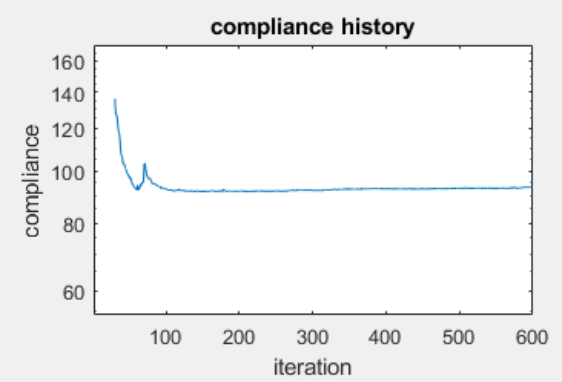
Final compliance	91.19
Number of components	8*4=32
Number of design variables	32*7=224
Stopping criterion	Max number of iteration reached
Total CPU time	174.43s
Average CPU time (per iteration)	0.29s

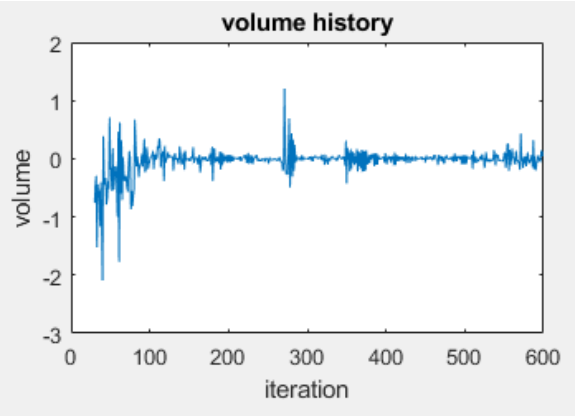
MMC with curvature results:

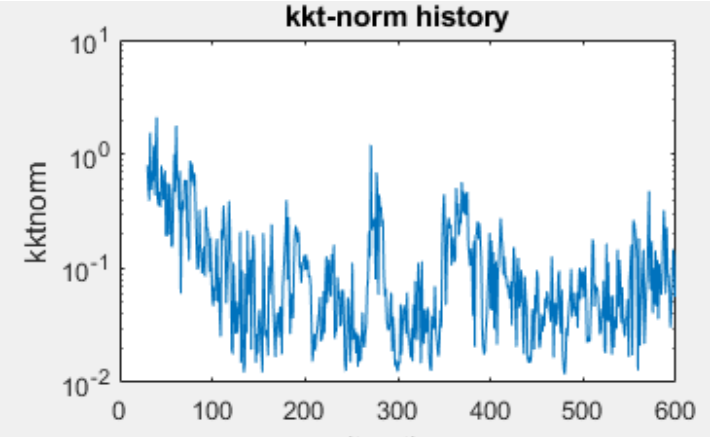
Initial configuration, final design and convergence history are shows in following figures:









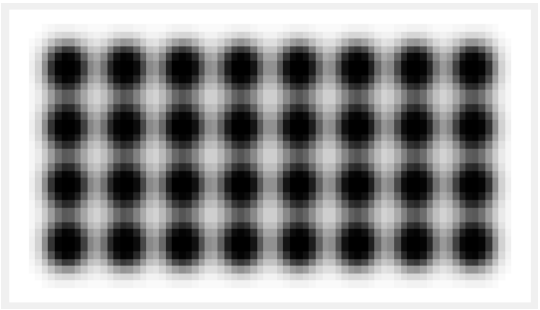


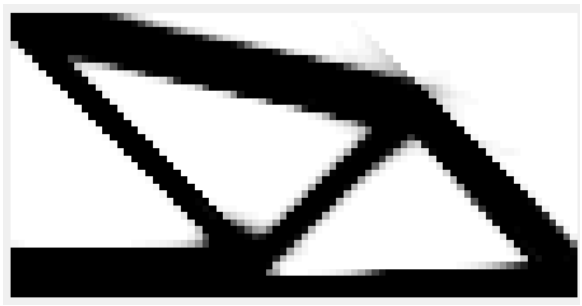
Following table summarizes the simulation results:

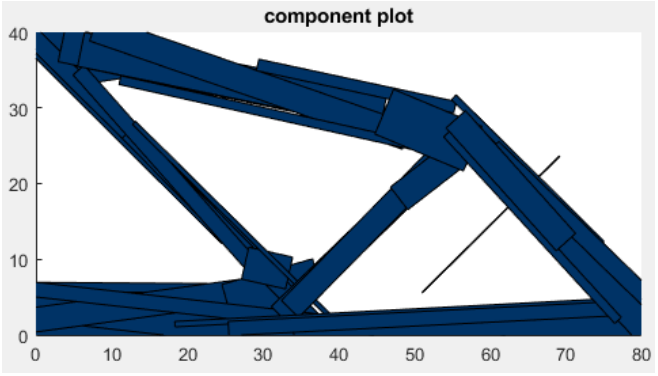
Final compliance	91.6
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	Max number of iteration reached
Total CPU time	135.6s
Average CPU time (per iteration)	0.226s

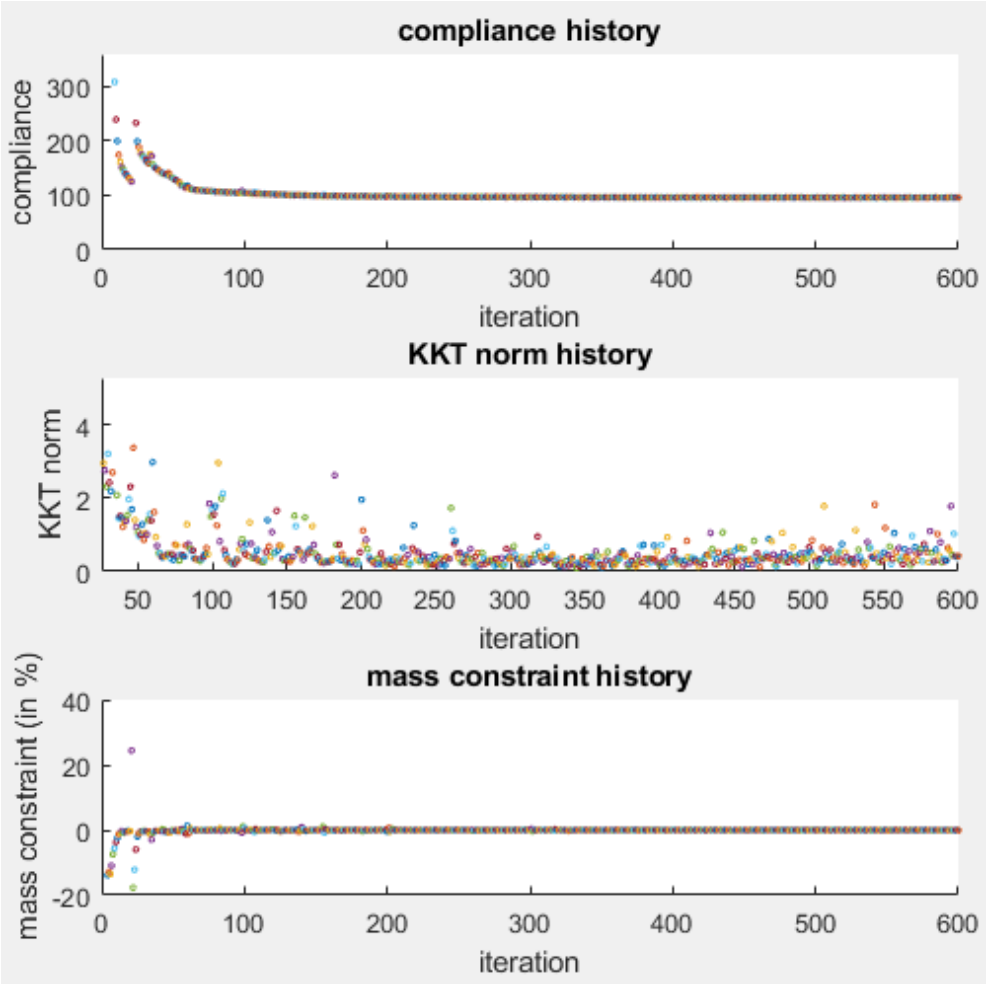
MNA code results:

Initial configuration, final design and convergence history are shows in following figures:









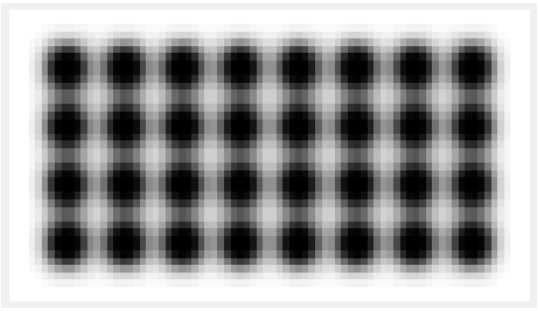
Following table summarizes the simulation results:

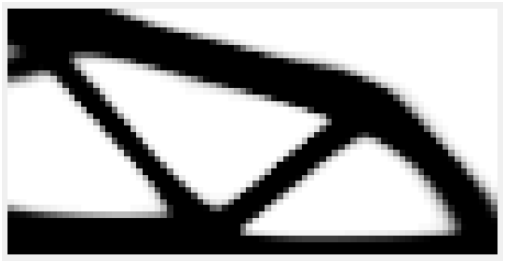
Final compliance	95.8

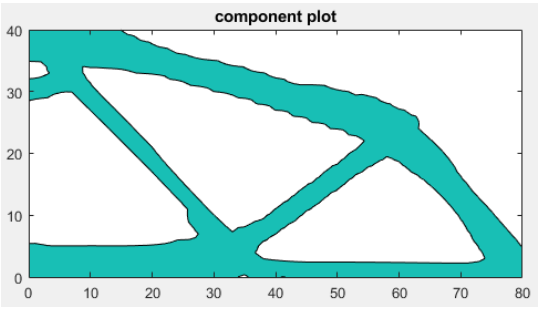
Number of components	8*4=32
Number of design variables	32*5=160
Stopping criterion	Max number of iteration reached
Total CPU time	199.9s
Average CPU time (per iteration)	0.33s

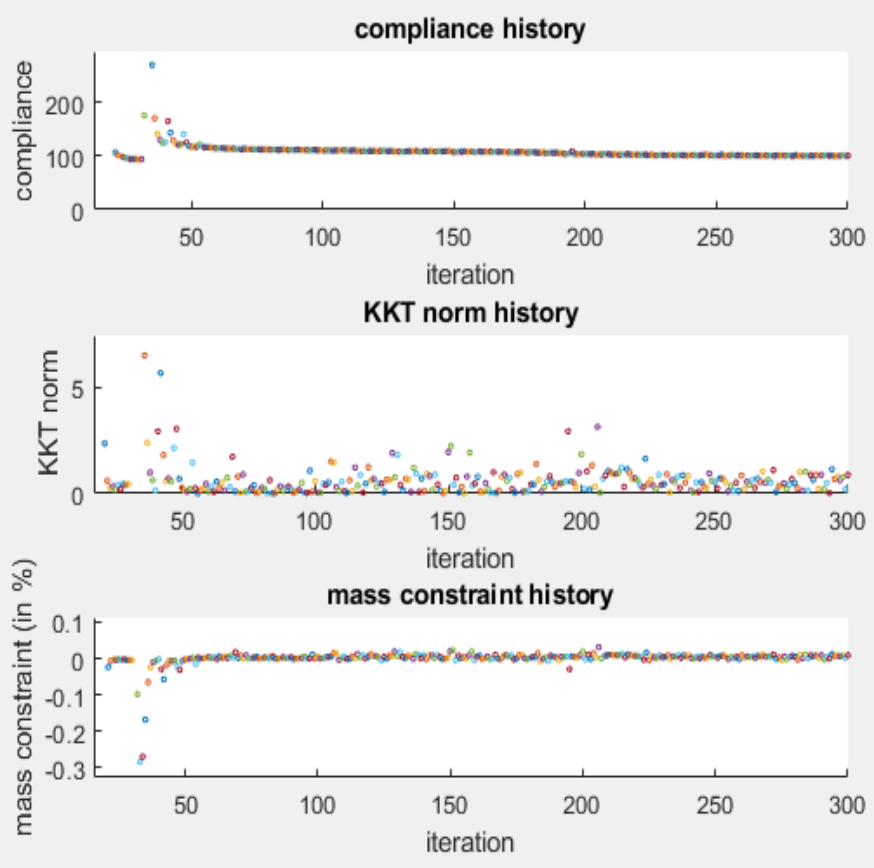
MNA with curvature results:

Initial configuration, final design and convergence history are shows in following figures:







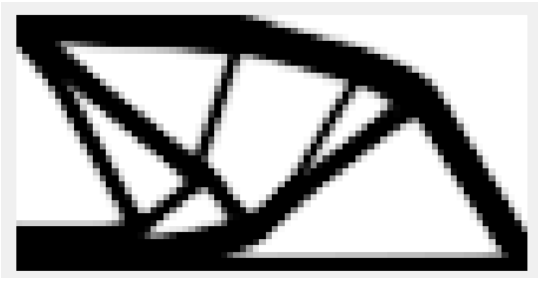


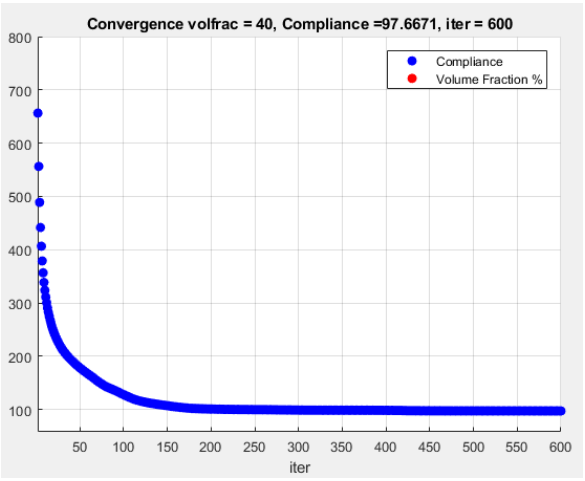
Following table summarizes the simulation results:

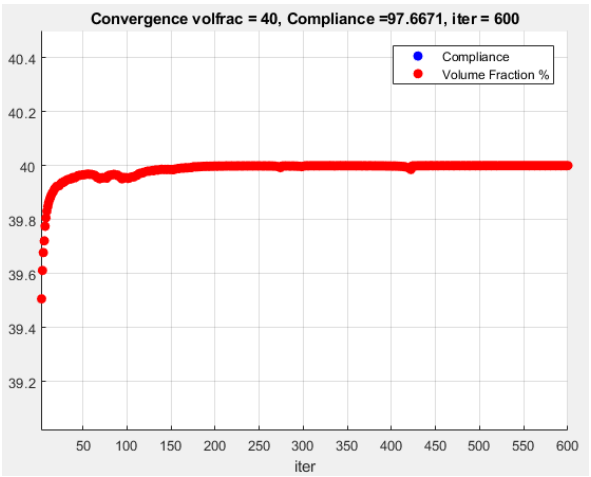
Final compliance	100
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	Max number of iteration reached (300)
Total CPU time	831s
Average CPU time (per iteration)	2.77s

SIMP (top88 with MMA) results:

Initial configuration, final design and convergence history are shows in following figures:







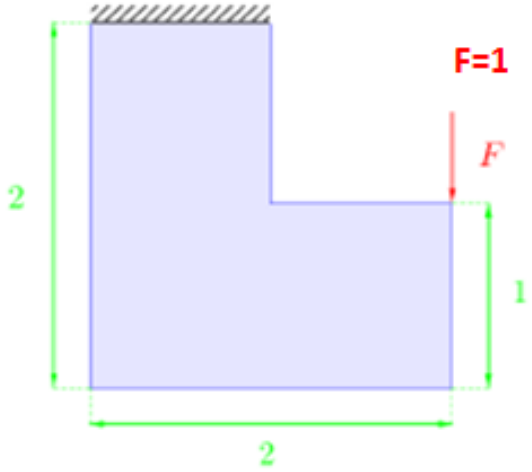
Following table summarizes the simulation results:

Final compliance	97.66
Number of components	
Number of design variables	80*40=3200
Stopping criterion	Max number of iteration reached
Total CPU time	203.8
Average CPU time (per iteration)	0.34s

Here, same remarks as in previous test case could be stated. Also, the MNA code with curvature takes a lot of time compared to other codes (that's why we set the maximum number of iterations to 300), this is because density field gradient is computed using finite differences, in addition the finite differences step may be not be optimal, leading to slower convergence. Another important point is that MNA uses only 5 design variables per components while MMC uses 7, and total number of design variables is lower in MNA than in MMC (although same number of components are used). Therefore, results should be interpreted considering this last point.

3) L-shape beam

In this example, design domain is a 2 by 2 square, where the up-right quarter of square is removed (in practice it is discretized by passive elements, where density is always equal to zero). Geometry, load and boundary conditions are shown in figure below:



The topology optimization formulation is the same, minimizing compliance under volume constraint:

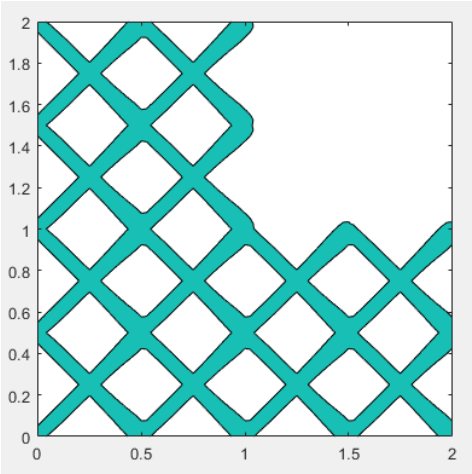
$$\begin{cases} \min\limits_{X} C \\ st\ V(X) < volfrac \end{cases}$$

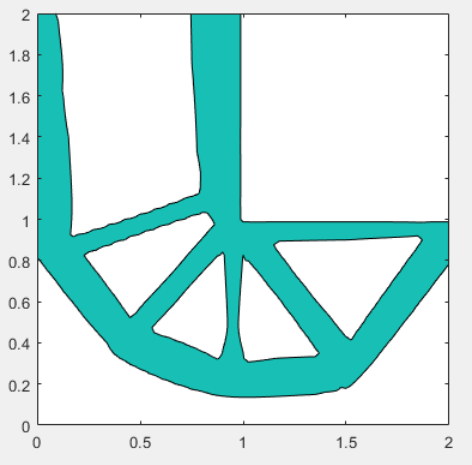
Following tables shows some parameters for the simulation:

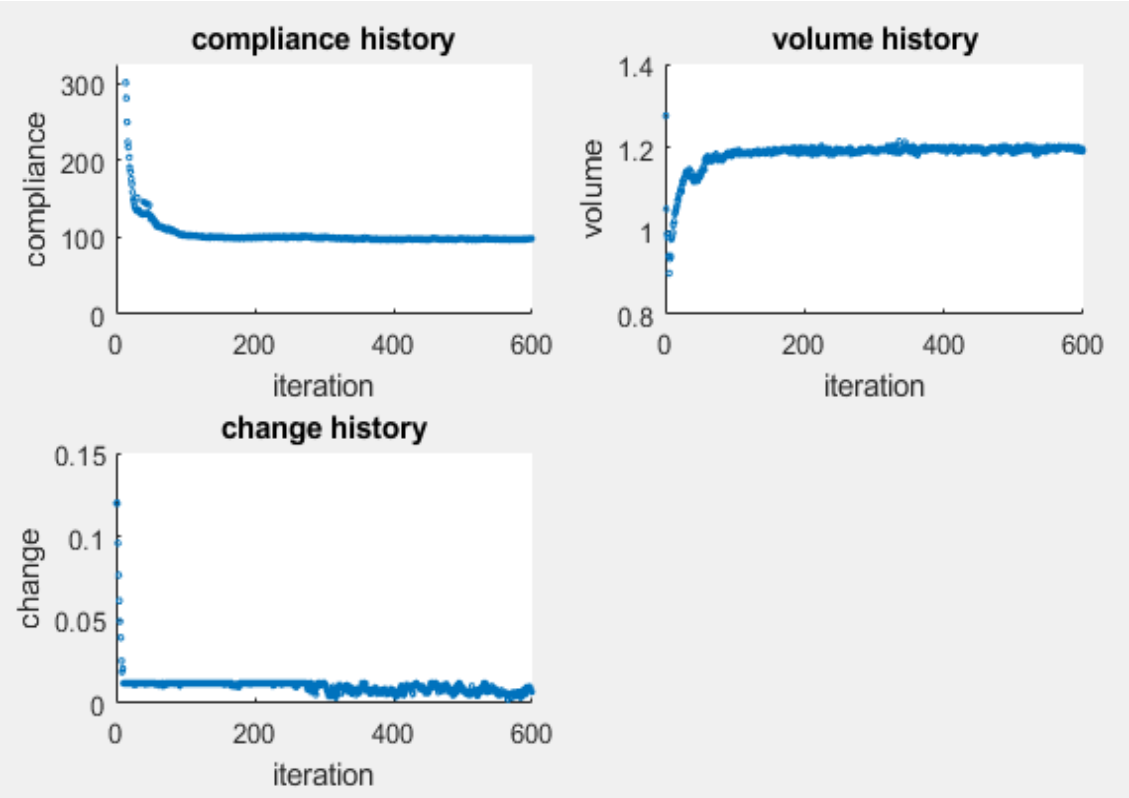
Mesh	80x80
Volfrac	0.4*0.75
Change tolerance	0.001
KKT tolerance	0.01(MMC) 0.002(MNA)
Maximum number of iterations	600

Original MMC code results:

Initial configuration, final design and convergence history are shows in following figures:





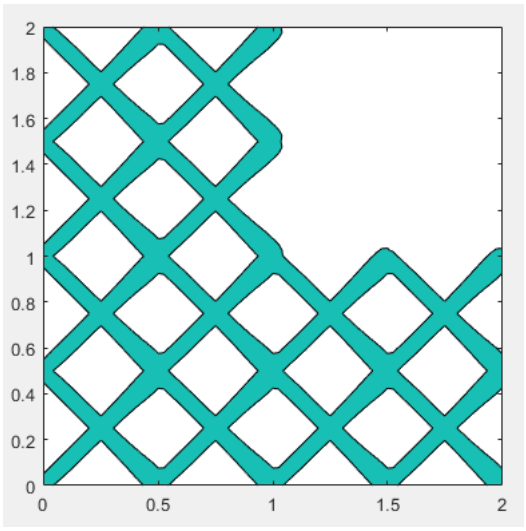


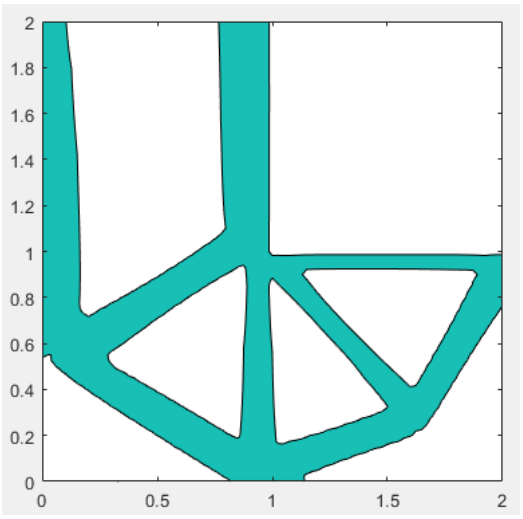
Following table summarizes the simulation results:

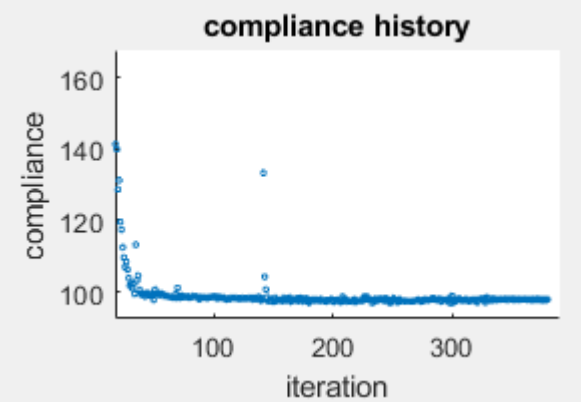
Final compliance	98.16
Number of components	8*8*0.75=48
Number of design variables	48*7=336
Stopping criterion	Max number of iterations reached
Total CPU time	342.13s
Average CPU time (per iteration)	0.57s

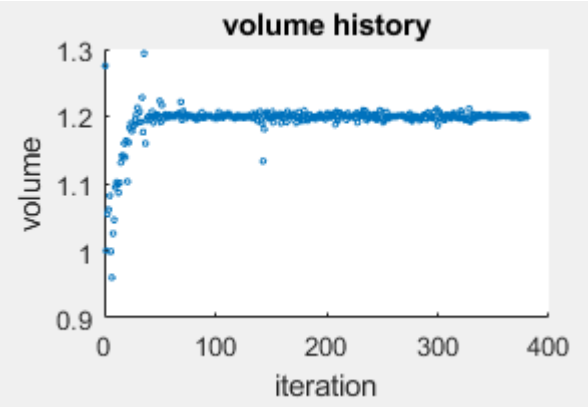
Enhanced MMC code results:

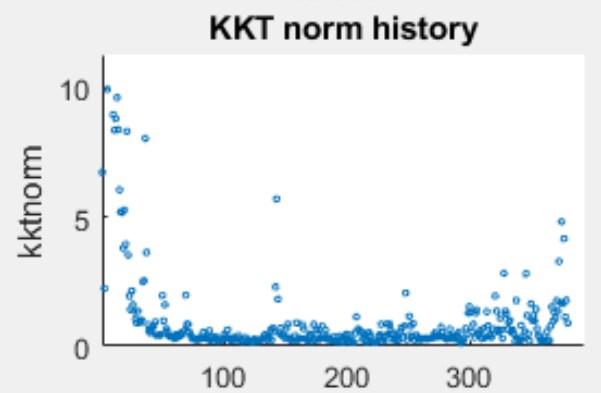
Initial configuration, final design and convergence history are shows in following figures:









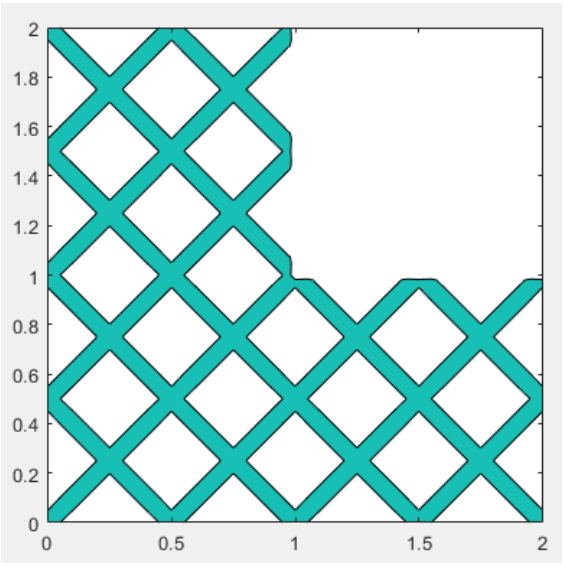


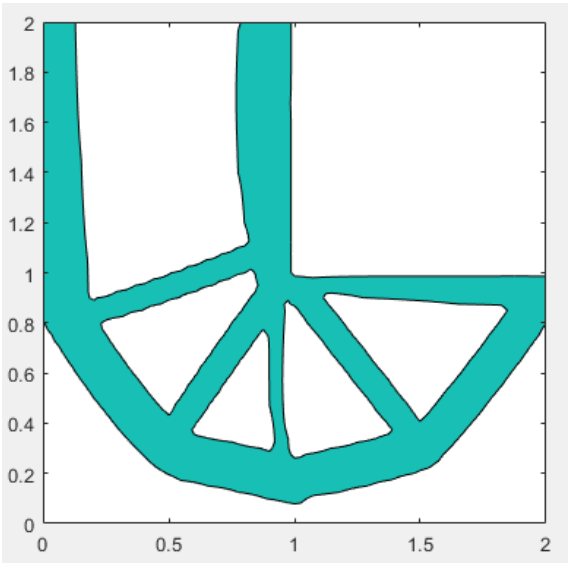
Following table summarizes the simulation results:

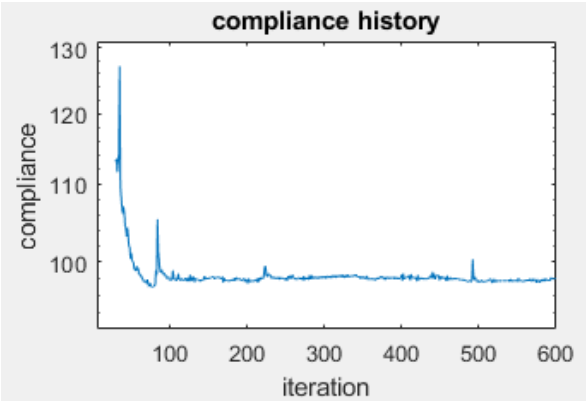
Final compliance	97.42
Number of components	8*8*0.75=48
Number of design variables	48*7=336
Stopping criterion	change satisfied after 380 iterations
Total CPU time	285s
Average CPU time (per iteration)	0.75s

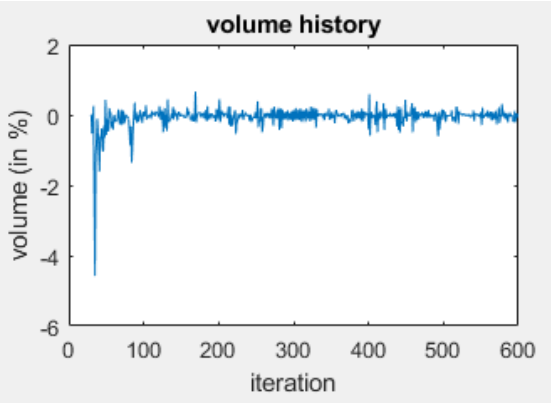
MMC with curvature results:

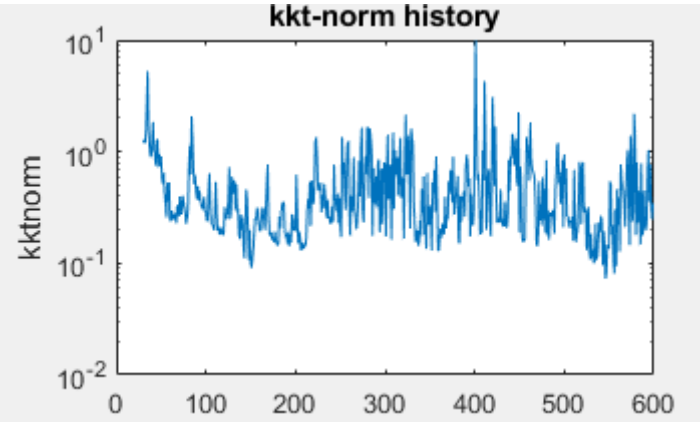
Initial configuration, final design and convergence history are shows in following figures:









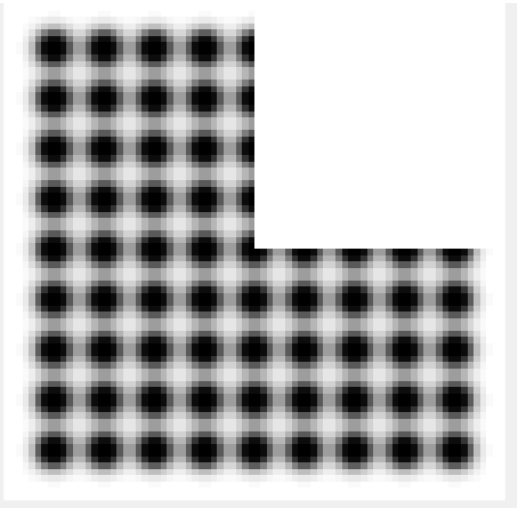


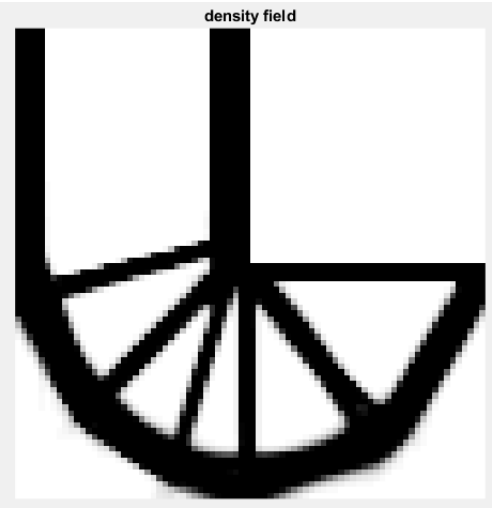
Following table summarizes the simulation results:

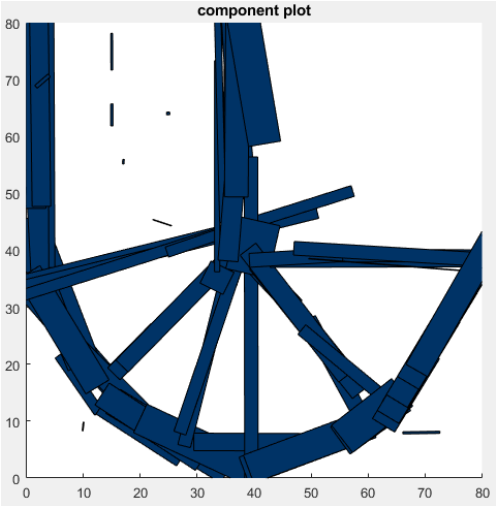
Final compliance	96.9
Number of components	8*8*0.75=48
Number of design variables	48*6=288
Stopping criterion	Max number of iterations reached
Total CPU time	354s
Average CPU time (per iteration)	0.59s

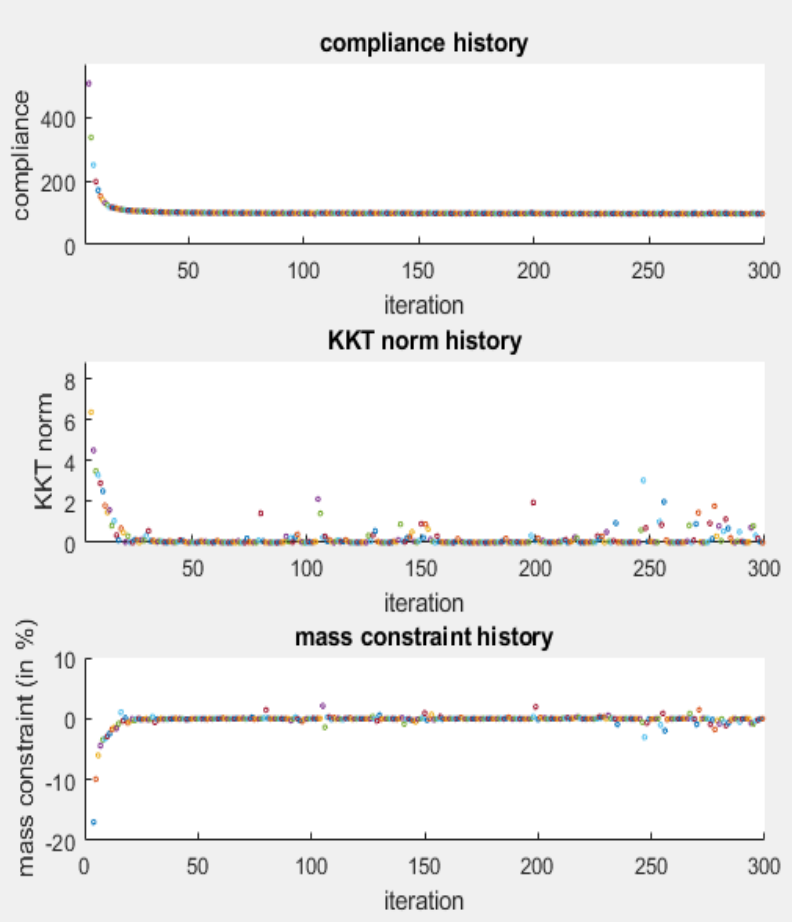
MNA code results:

Initial configuration, final design and convergence history are shows in following figures:







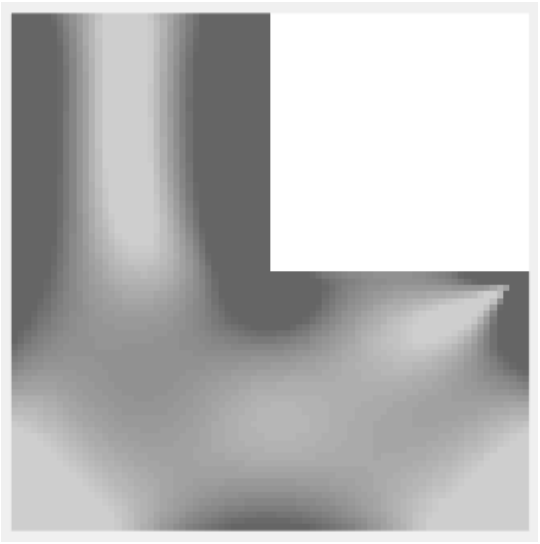


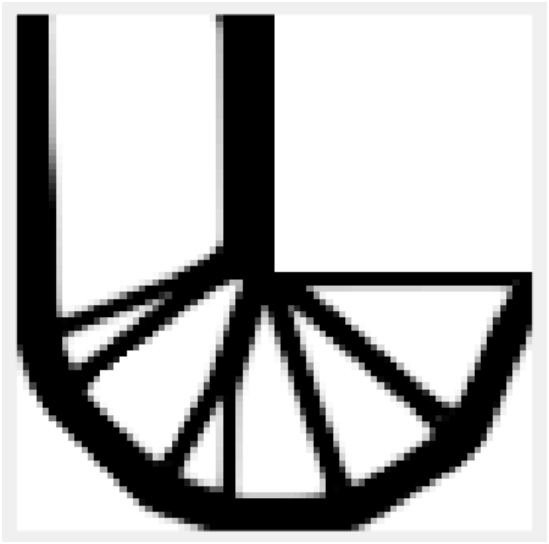
Following table summarizes the simulation results:

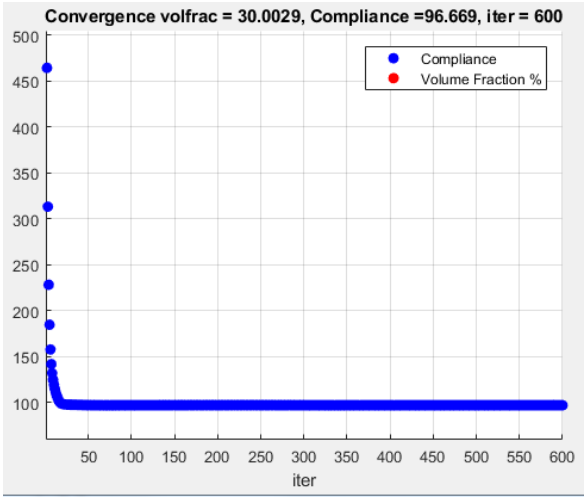
Final compliance	97.15
Number of components	9*9*0.75=65
Number of design variables	65*5=325
Stopping criterion	KKT norm satisfied after 299 iterations
Total CPU time	160.6s
Average CPU time (per iteration)	0.53s

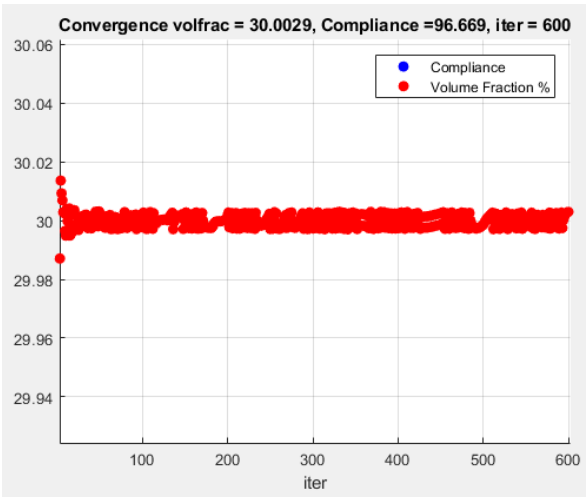
SIMP (top88 with MMA) results:

Initial configuration, final design and convergence history are shows in following figures:









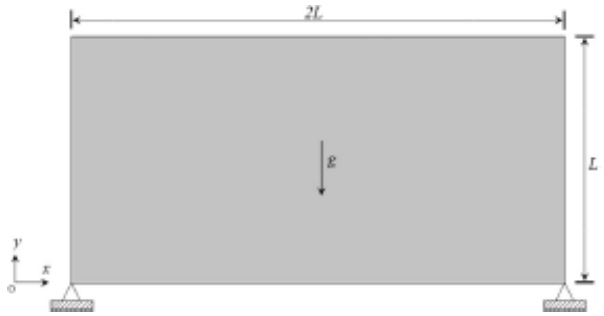
Following table summarizes the simulation results:

Final compliance	96.66
Number of components	
Number of design variables	80*80*0.75=4800
Stopping criterion	Max number of iterations reached
Total CPU time	231.6s
Average CPU time (per iteration)	0.386s

Here, instead of starting with same number of components for MMC and MNA, as in previous test cases (which will leads to different design variable numbers, as mentioned before). We start with close design variable numbers: 336 for MMC and 325 for MNA. Better results were obtained for MNA, compared to MMC (straight) codes, this proves the potential and efficiency of MNA. Moreover, curved features could be seen in the design of MNA code with straight components, indeed, since more intermediate density is involved in MNA, better control over the structure's shape is possible and curved features and also variable sections can be reproduced using only straight components.

4) Self-supported beam

In this example we use a 2x1 beam simply supported from both lower corners, the load is equal to the structures weight and applied at the center of the beam. Design domain is discretized by a 160x80 mesh. The following figure shows loads and boundary conditions:



This example is different from other ones, since the load depends on the design (density field) which changes at each iteration, thus we can no more use: $\frac{\partial F}{\partial x_i} = 0$ where F is the load and x_i is a design variable. The gradient of compliance needs to be calculated again:

Static equilibrium: $KU = F$,

After derivation: $\frac{\partial K}{\partial x}U + K\frac{\partial U}{\partial x} = \frac{\partial F}{\partial x}$ $(\neq 0)$, where x is a design variable.

Then $\frac{\partial U}{\partial x} = K^{-1}\left(\frac{\partial F}{\partial x} - \frac{\partial K}{\partial x}U\right)$,

Compliance can be written as: $C = F^t U$,

This gives after derivation: $\frac{\partial C}{\partial x} = \frac{\partial F^t}{\partial x}U + F^t\frac{\partial U}{\partial x}$

The term $\frac{\partial U}{\partial x}$ can be substituted by its expression to obtain: $\frac{\partial C}{\partial x} = \frac{\partial F^t}{\partial x}U + F^t K^{-1}\left(\frac{\partial F}{\partial x} - \frac{\partial K}{\partial x}U\right) = \frac{\partial F^t}{\partial x}U + U^t K^t K^{-1}\left(\frac{\partial F}{\partial x} - \frac{\partial K}{\partial x}U\right)$

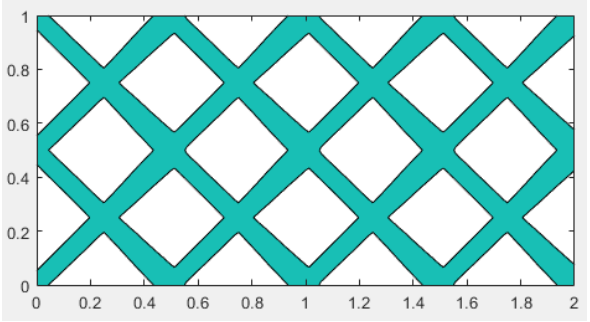
Using symmetry of K and $\frac{\partial F^t}{\partial x}U = U^t\frac{\partial F}{\partial x}$ we obtain $\frac{\partial C}{\partial x} = -U^t\frac{\partial K}{\partial x}U + 2\frac{\partial F^t}{\partial x}U$

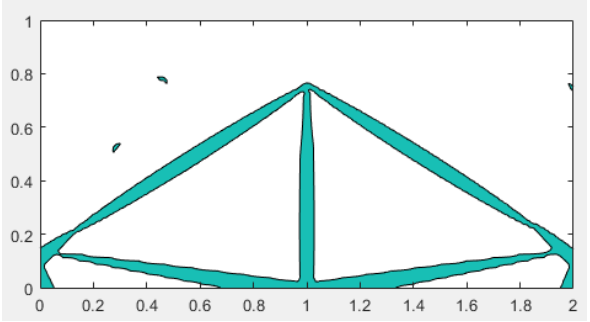
A 160x80 mesh is used and volume fraction constraint is set to 0.5, since load depends on density, a full design would not necessary gives the 'absolute' minimal compliance. In fact stiffness is increasing but load also, so we can't say whether the compliance will decrease or not. Thus minimizing compliance with volume superior to volume fraction is not a trivial formulation.

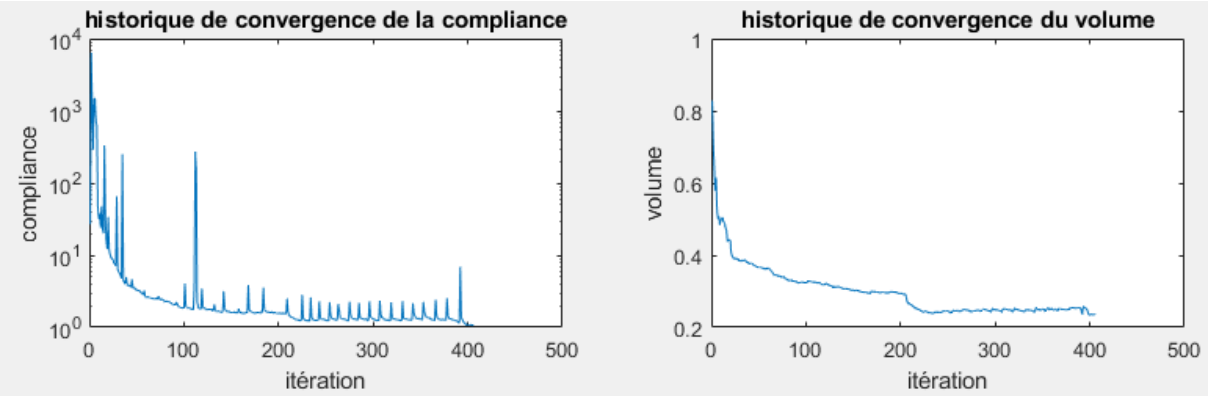
In following we present results of the first formulation: $\begin{cases} find\ x,\ \min C \\ st\ V(x) \leq 0.5 \end{cases}$

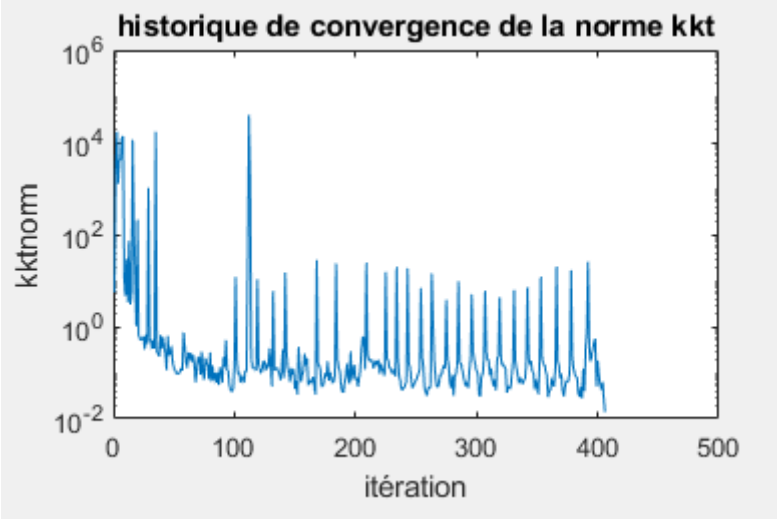
Enhanced MMC code results

Initial configuration, final design and convergence history are shows in following figures:







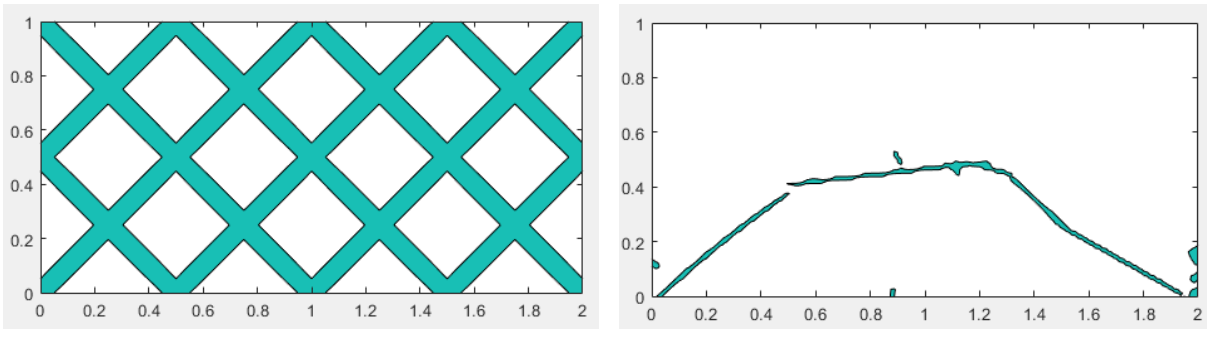


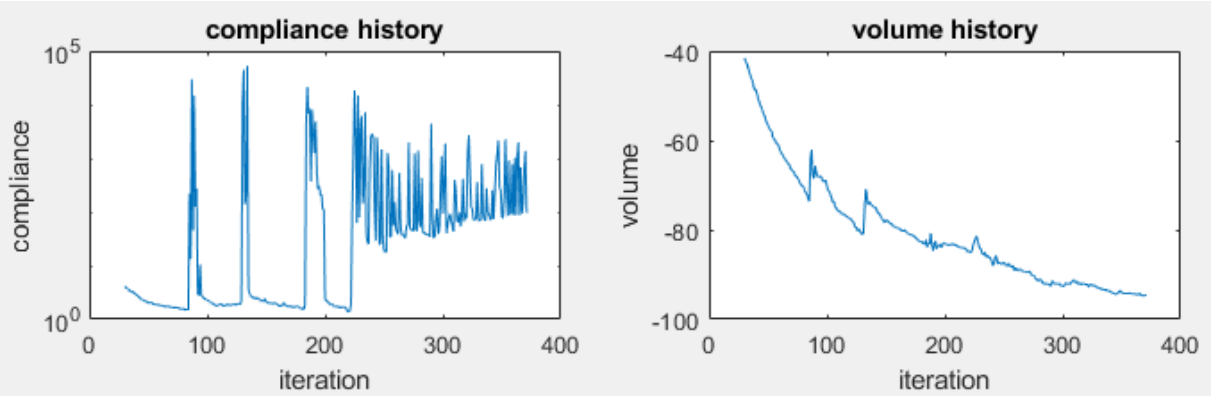
Following table summarizes the simulation results:

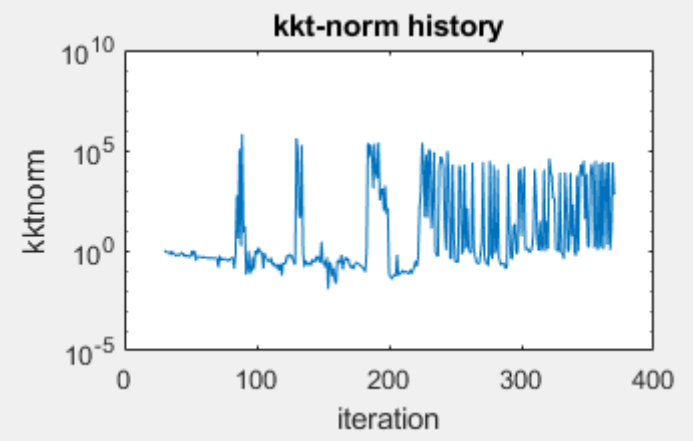
Final compliance	1.07
Number of components	8*4=32
Number of design variables	32*7=224
Stopping criterion	KKT tolerance (0.02) satisfied in 406 iterations
Total CPU time	372s
Average CPU time (per iteration)	0.916s
Final volume	24%

MMC with curvature results

Initial configuration, final design and convergence history are shows in following figures:





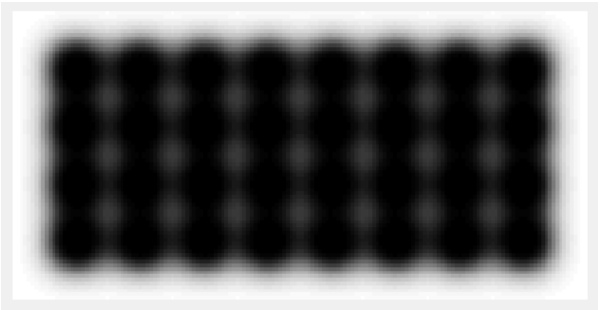


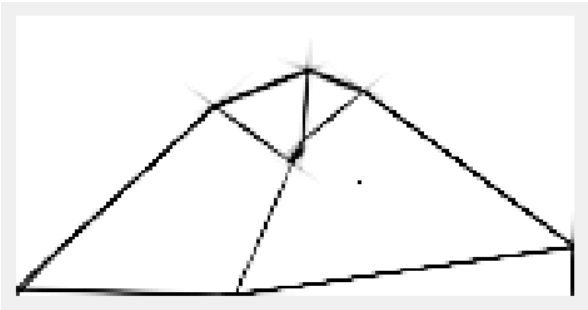
Following table summarizes the simulation results:

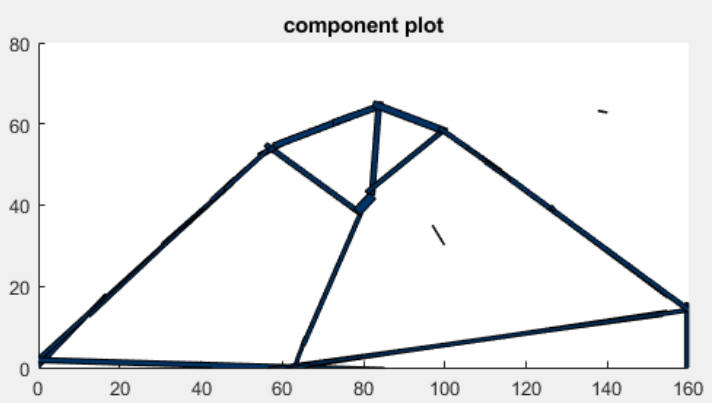
Final compliance	96.2
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	KKT=NaN at iteration 371
Total CPU time	271.8
Average CPU time (per iteration)	0.73s
Final volume	6%

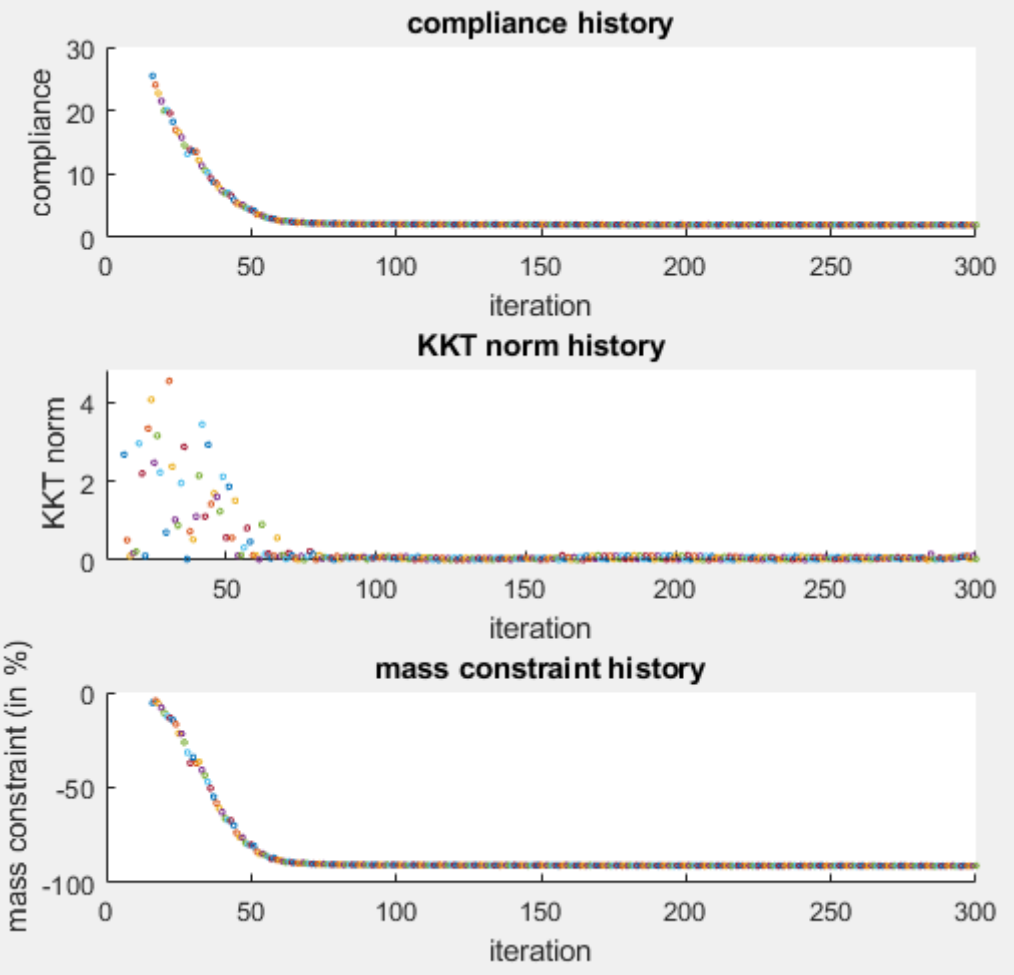
MNA code results

Initial configuration, final design and convergence history are shows in following figures:







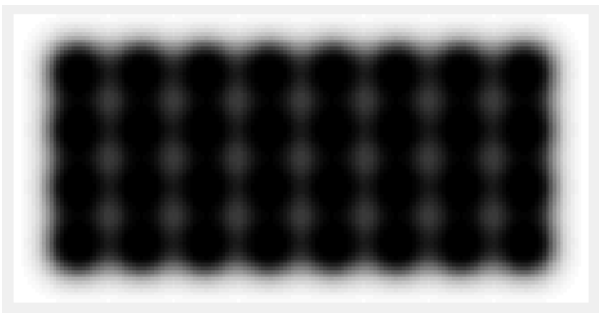


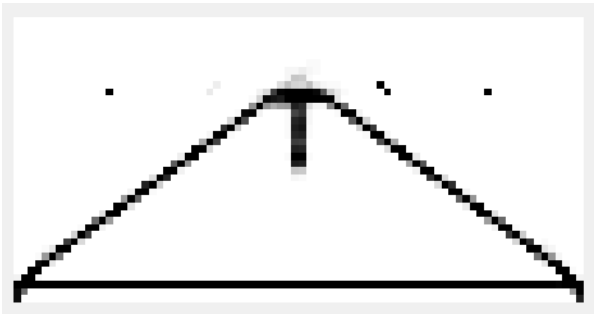
Following table summarizes the simulation results:

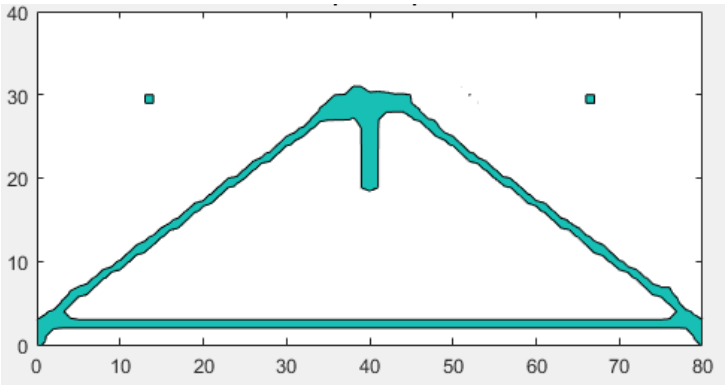
Final compliance	1.95
Number of components	8*4=32
Number of design variables	32*5=160
Stopping criterion	max number of iterations (300) reached
Total CPU time	320.6s
Average CPU time (per iteration)	1.06s
Final volume	9%

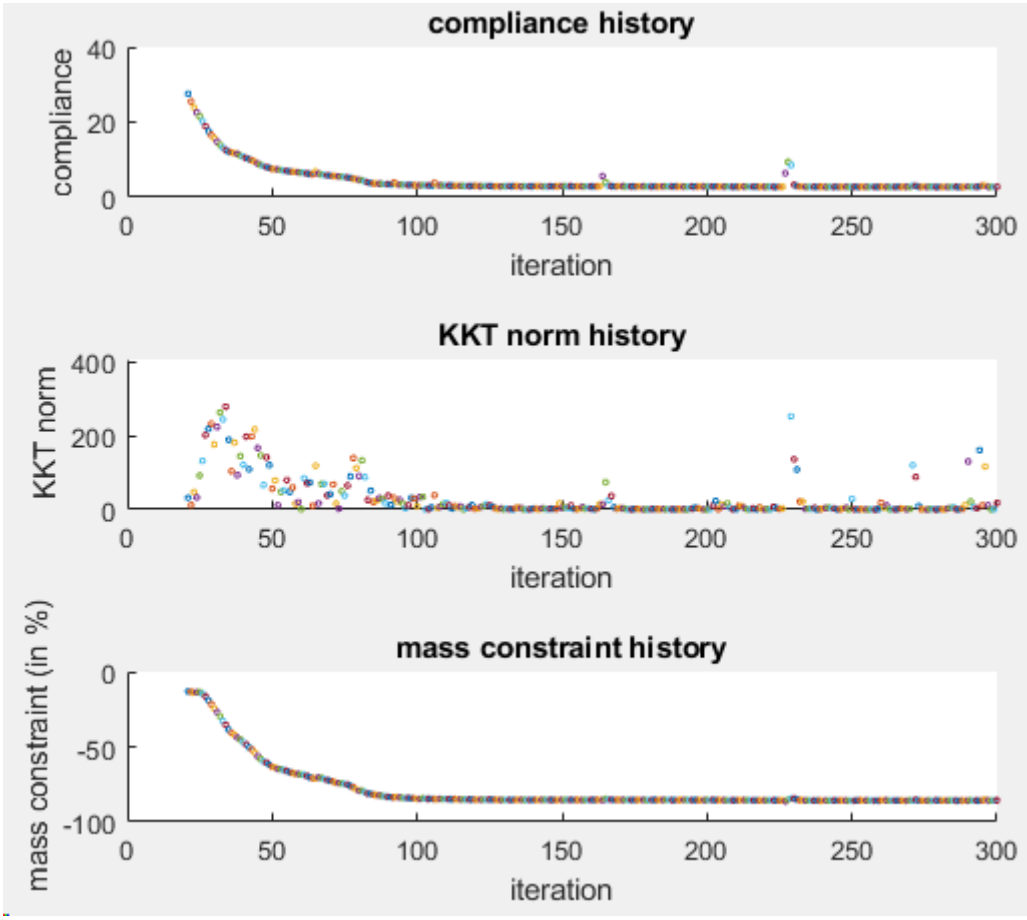
MNA with curvature results

Here we should mention that a 80x40 mesh is used (to reduce computational time: FD). Initial configuration, final design and convergence history are shows in following figures:









Following table summarizes the simulation results:

Final compliance	2.761
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	Max number of iterations (300) reached
Total CPU time	
Average CPU time (per iteration)	
Final volume	14.25%

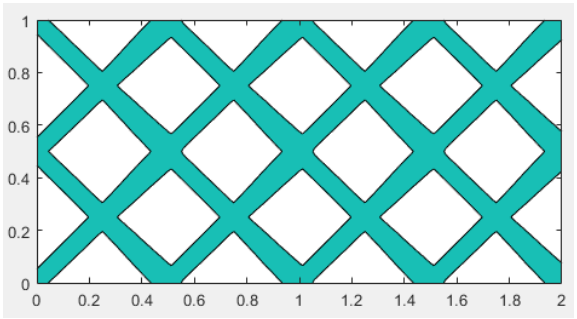
It is worth noting that volume constraint is not activated in this problem, indeed, low compliances are achieved with less volume, since the load decreases (even though stiffness decreases too, but load has “the upper hand”). And results will be the same if no volume constraint was imposed, thus, this formulation is an unconstrained problem.

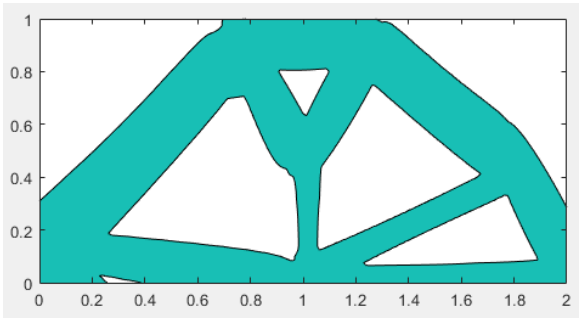
This test case may not be representative of real problems, however it allowed us to investigate the behavior of MMC and MNA methods at low volume fraction.

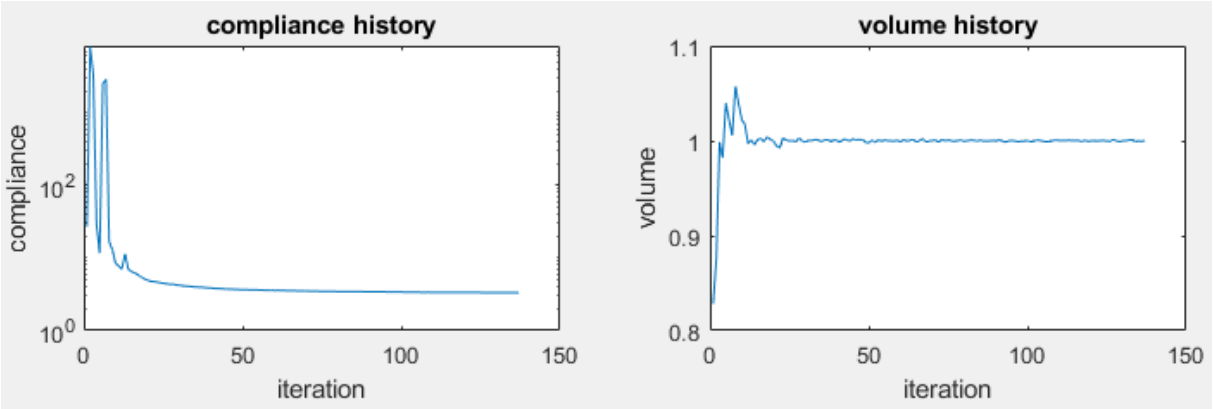
Now, we present results of the constrained problem: $\begin{cases} find\ x,\ \min C \\ st\ \ V(x) \geq 0.5 \end{cases}$

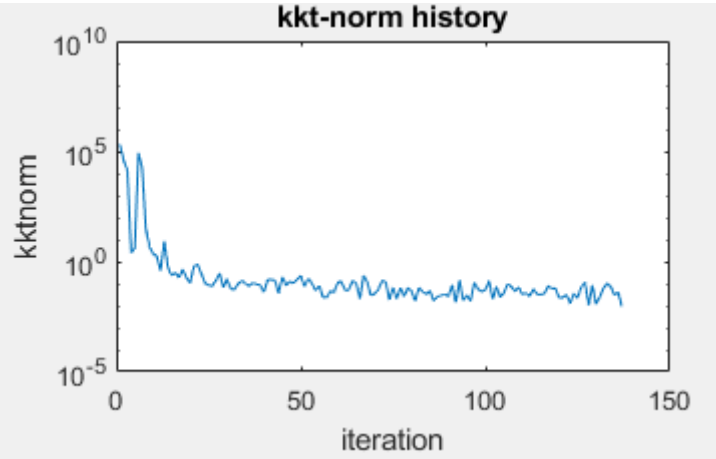
Enhanced MMC code results

Initial configuration, final design and convergence history are shows in following figures:







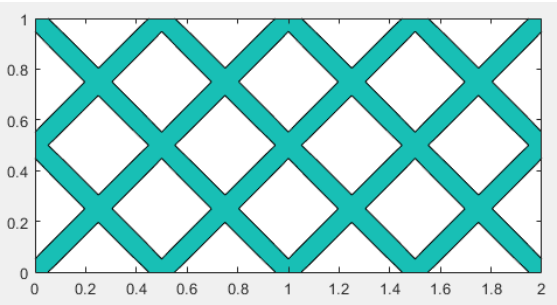


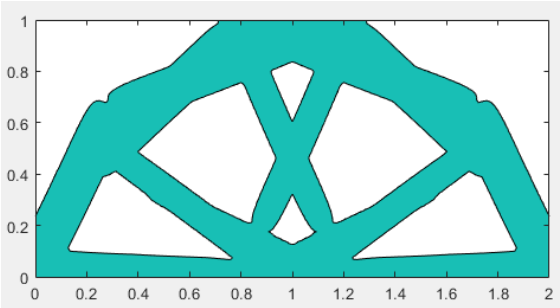
Following table summarizes the simulation results:

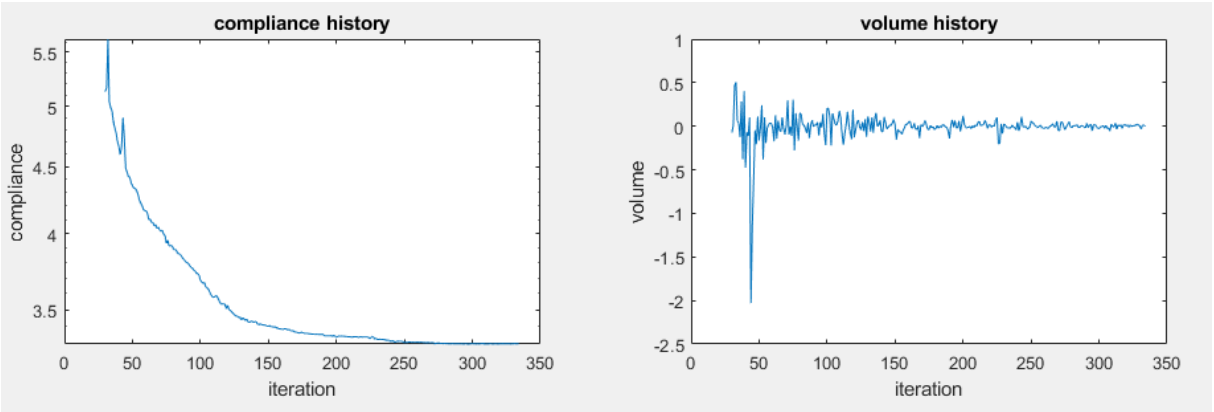
Final compliance	3.325
Number of components	8*4=32
Number of design variables	32*7=224
Stopping criterion	KKT tolerance (0.01) satisfied in 137 iterations
Total CPU time	126.16s
Average CPU time (per iteration)	0.92s

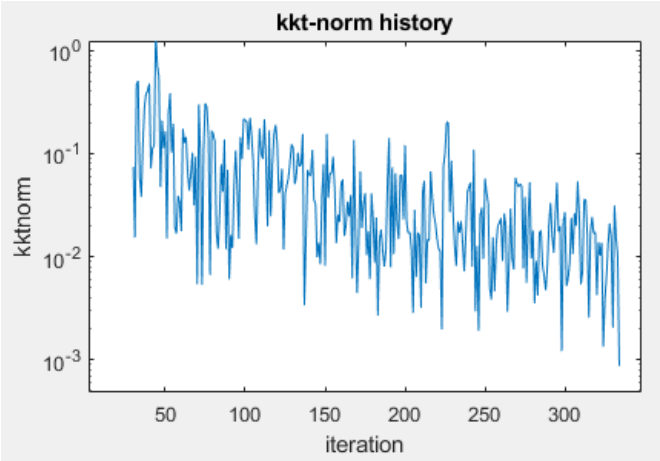
MMC with curvature results:

Initial configuration, final design and convergence history are shows in following figures:







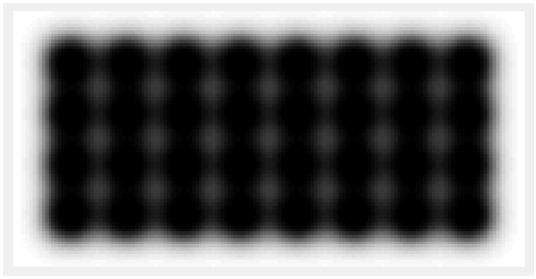


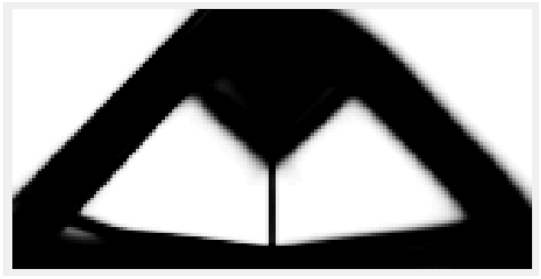
Following table summarizes the simulation results:

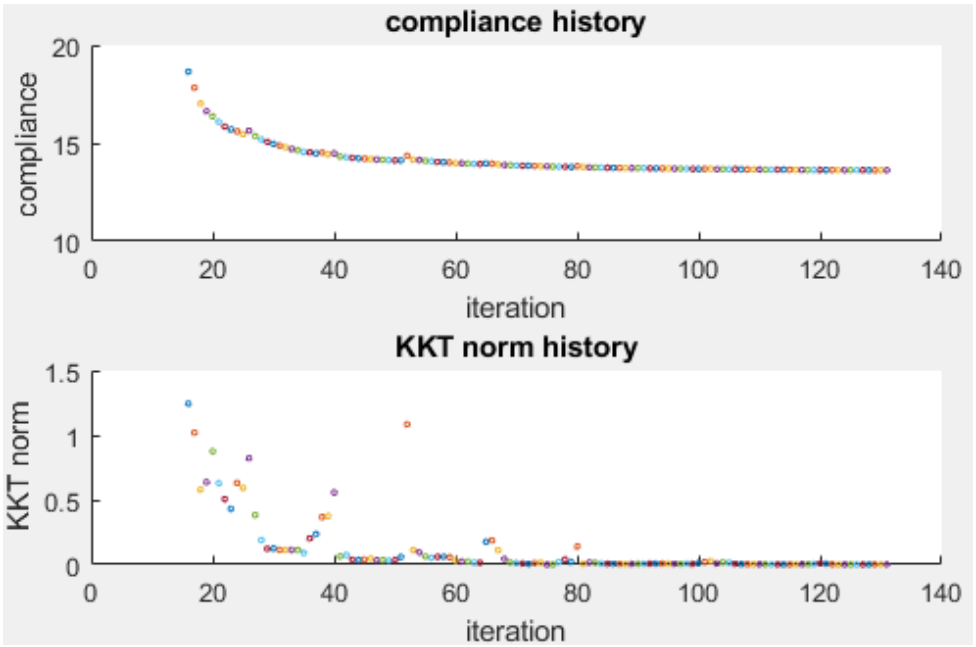
Final compliance	3.297
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	KKT tolerance (0.001) satisfied in 334 iterations
Total CPU time	252s
Average CPU time (per iteration)	0.754s

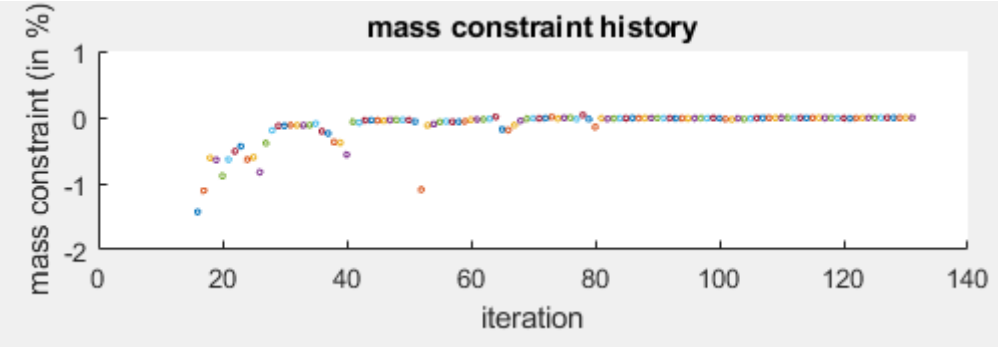
MNA code results

Initial configuration, final design and convergence history are shows in following figures:







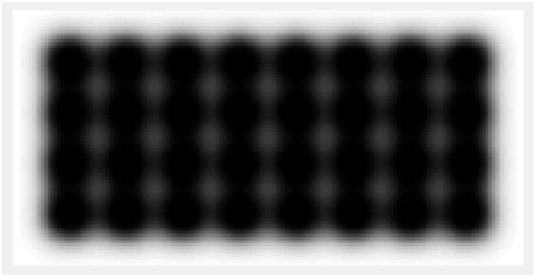


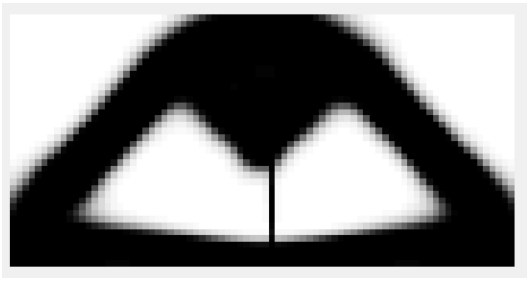
Following table summarizes the simulation results:

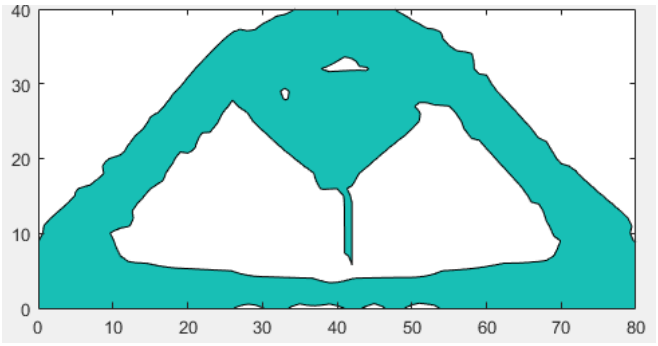
Final compliance	13.6
Number of components	8*4=32
Number of design variables	32*5=160
Stopping criterion	KKT tolerance (0.001) satisfied in 131 iterations
Total CPU time	97.9
Average CPU time (per iteration)	0.747s

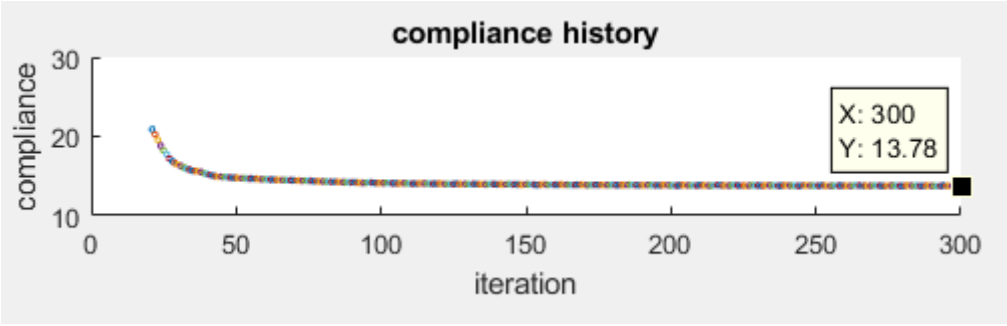
MNA with curvature results

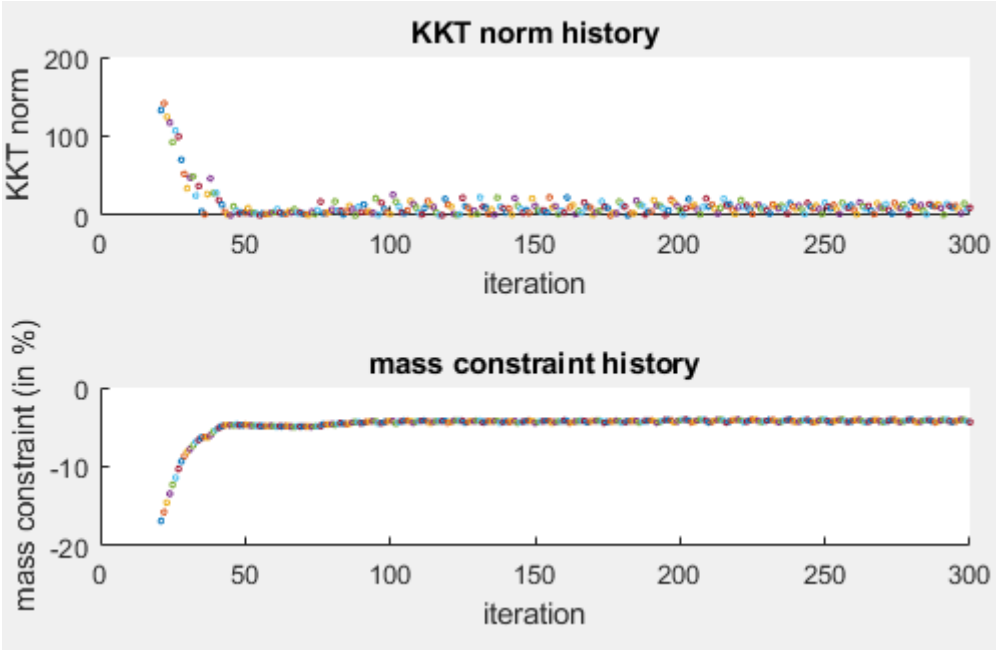
Initial configuration, final design and convergence history are shows in following figures:











Following table summarizes the simulation results:

Final compliance	13.78
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	max number of iterations (300) reached
Total CPU time	
Average CPU time (per iteration)	

In this second formulation, one can notice that volume constraint was all the times activated, indeed the problem is constrained under this formulation.

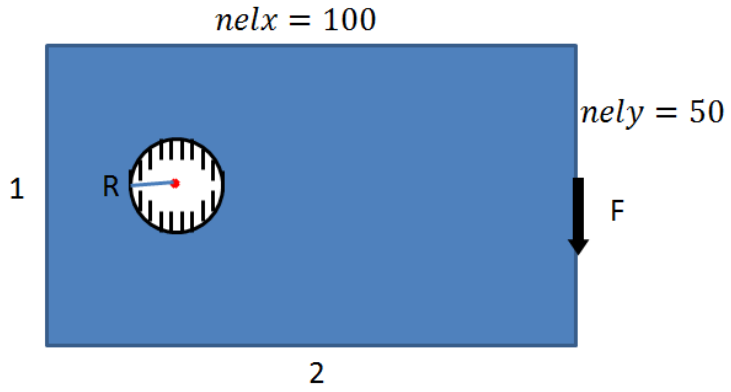
The self-supported beam is a challenging test case its particularity consist on the dependence of load on design variables by the mean of density field, which makes it different than the previous classic test cases. From results of this test case, some remarks could be drawn:

- MMC results were better for the constrained case, where volume fraction is not very low. Yet, one should keep in mind that more design variables are involved in MMC formulation than in MNA (224 against 160)
- In MNA with curvature code a coarser mesh was used (80x40 against 160x80), this results in higher compliance since refining the mesh in FE method increases the stiffness of a structure.
- Holding the symmetry of design becomes harder at low volume fraction
- The global optimum of the unconstrained problem is a zero density field (0% of volume) and compliance is equal to zero since there is no more load!
- The optimizer tries to reach that optimum, but finds difficulty to reach 0% of volume, because the structure becomes easily disconnected at low volume fractions and compliance increases drastically even if load is low.
- MNA has better ability to find a connected design under low volume fractions because of the presence of intermediates density which makes the disconnection progressive.

- KKT=NaN, means that KKT norm is not a number, this happened for the MMC with curvature case, in fact, in MMC method the Heaviside derivatives are evaluated by finite differences, and if the step of finite differences is equal to the thickness (or length) of a component, this component will disappear when perturbed negatively and the value of its TDF will be $-\infty$ (since $\Phi = 1 - \left(\frac{x_1}{L}\right)^p - \left(\frac{y_1}{t}\right)^p$) leading to a value of 0 for the Heaviside, except if its evaluated at a node that coincides with the center of the component, in which case $x_1 = y_1 = 0$, and both TDF and Heaviside are not numbers (NaN)! Moreover, if KKT condition is used, KKT norm will also be a NaN (since NaNs are present in gradients vectors) and the optimization process will end (because the KKT condition cannot be evaluated in the next iteration)

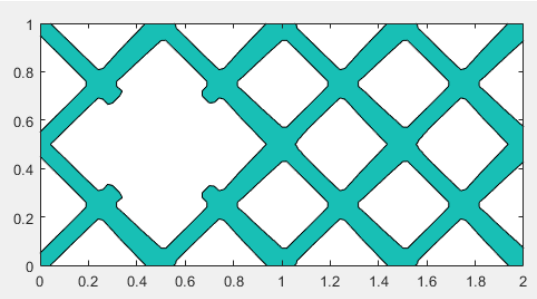
5) Mitchell truss

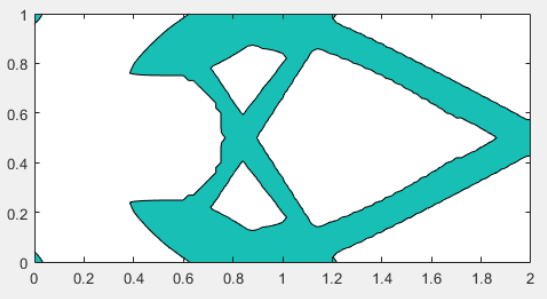
Here again, another academic test case, a mesh of 100x50 is used, volume fraction constraint is set to 0.3, a circular hole of center $(x_c, y_c) = (25,25)$ and radius $R = 12.5$, fixed boundary conditions are applied on the circumference of the hole and load is the same as cantilever beam, the following picture shows the geometry, load and boundary conditions of the Mitchell truss:

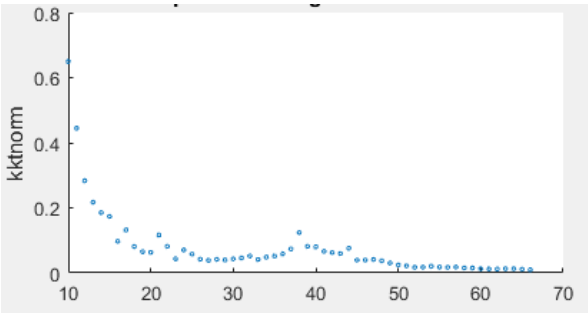


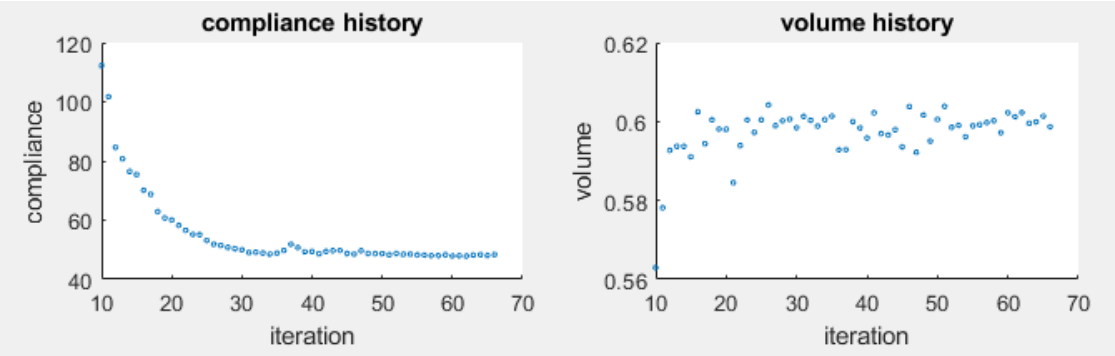
Enhanced MMC code results

Initial configuration, final design and convergence history are shows in following figures:







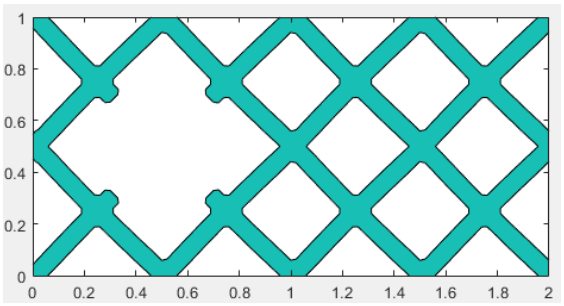


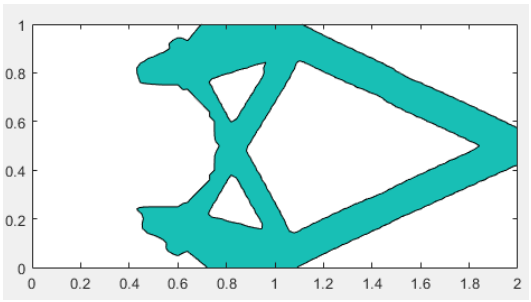
Following table summarizes the simulation results:

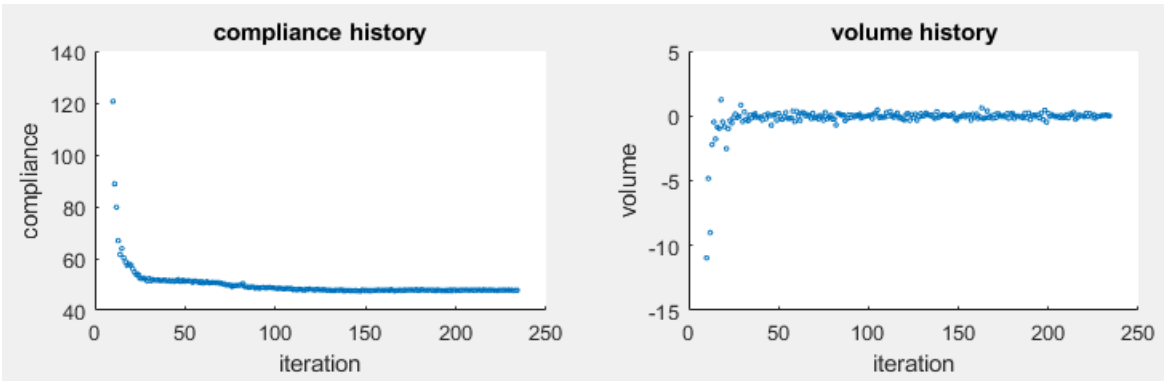
Final compliance	48
Number of components	8*4=32
Number of design variables	32*7=224
Stopping criterion	KKT tolerance (0.01) satisfied in 66 iterations
Total CPU time	33.11s
Average CPU time (per iteration)	0.5s

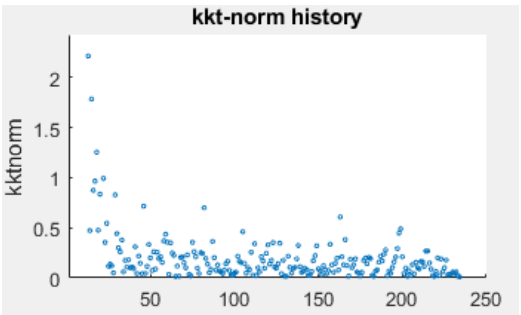
MMC with curvature results

Initial configuration, final design and convergence history are shows in following figures:







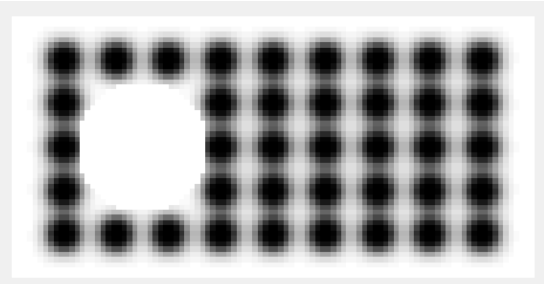


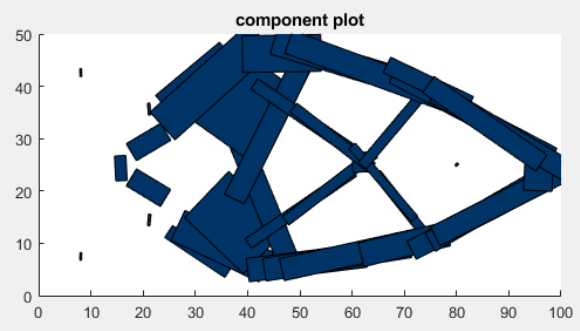
Following table summarizes the simulation results:

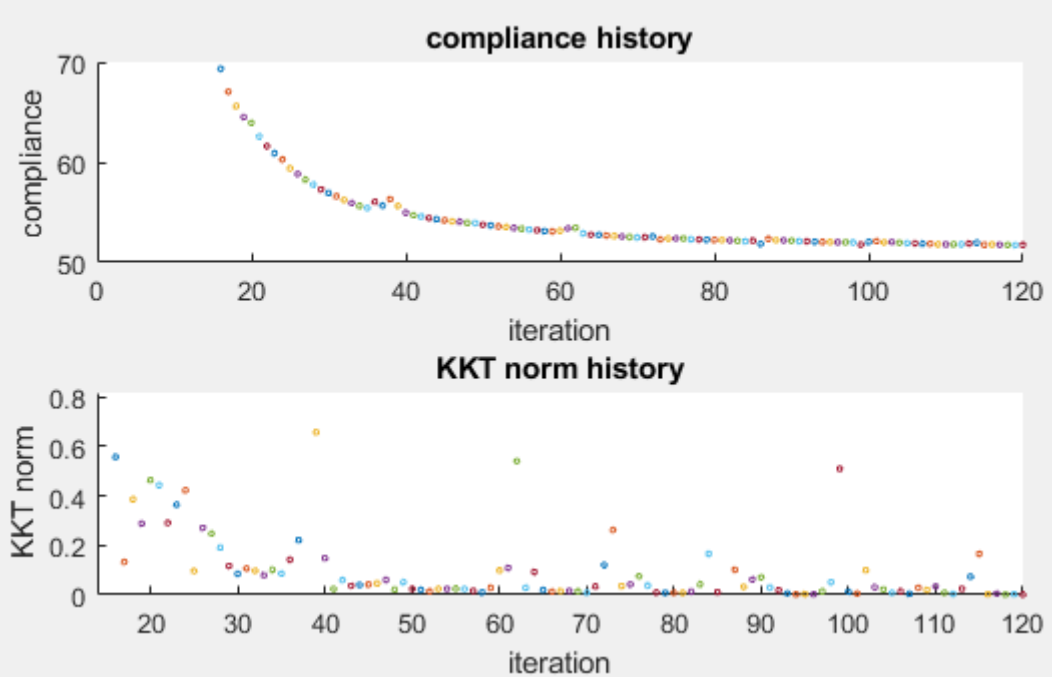
Final compliance	47.57
Number of components	8*4=32
Number of design variables	32*6=192
Stopping criterion	KKT tolerance (0.01) satisfied in 234 iterations
Total CPU time	74.8s
Average CPU time (per iteration)	0.32s

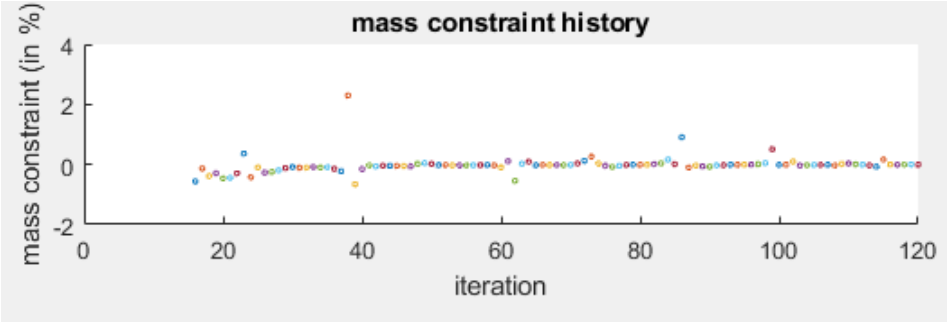
MNA code results

Initial configuration, final design and convergence history are shows in following figures:







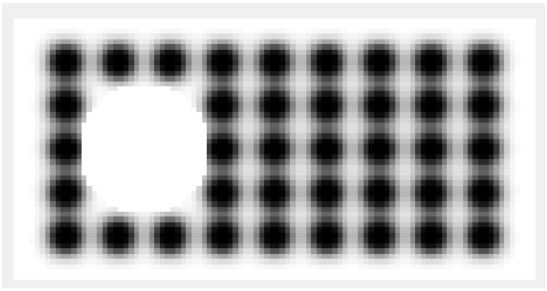


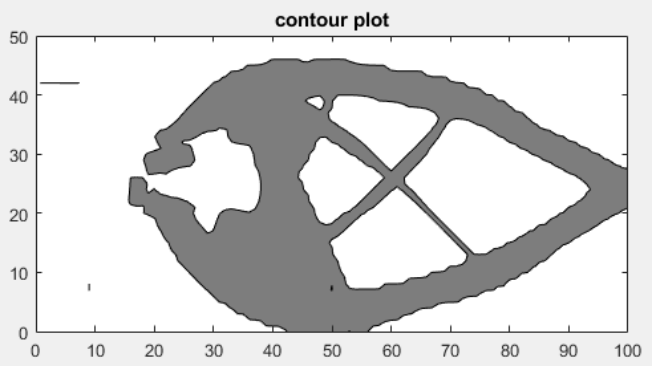
Following table summarizes the simulation results:

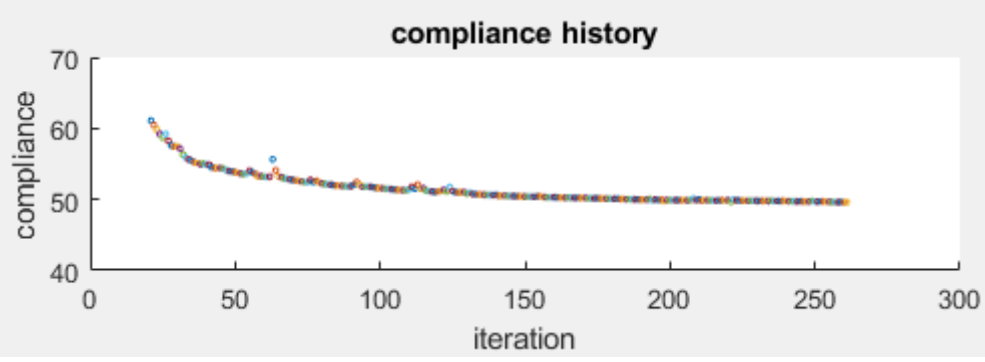
Final compliance	51.73
Number of components	9*5=45
Number of design variables	45*5=225
Stopping criterion	KKT tolerance (0.001) satisfied in 120 iterations
Total CPU time	41.11s
Average CPU time (per iteration)	0.34s

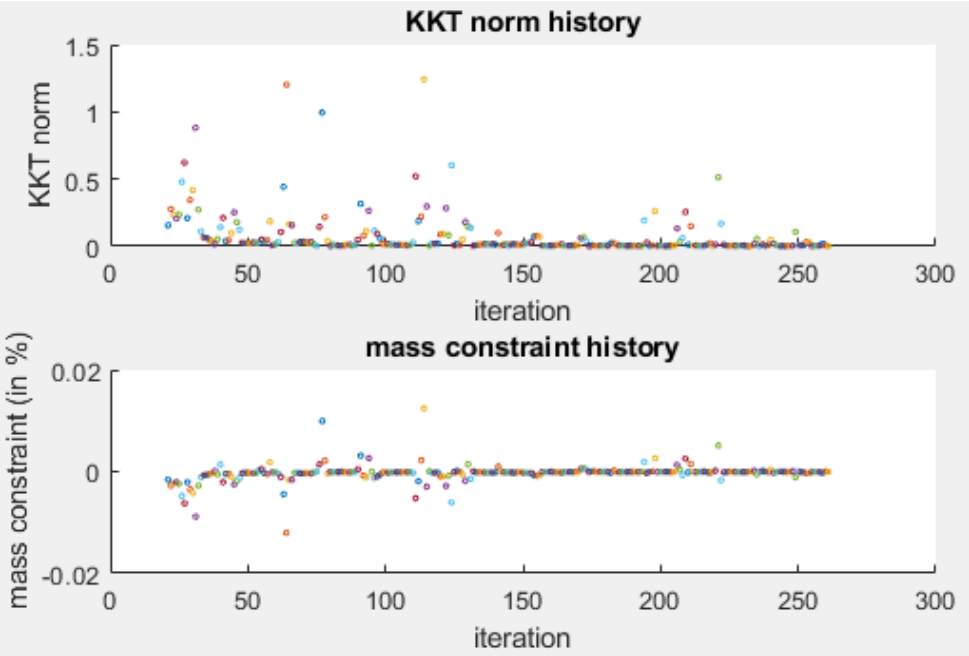
MNA with curvature results

Initial configuration, final design and convergence history are shows in following figures:









Following table summarizes the simulation results:

Final compliance	49.6
Number of components	9*5=45
Number of design variables	45*6=270
Stopping criterion	KKT tolerance (0.001) satisfied in 261 iterations
Total CPU time	943s
Average CPU time (per iteration)	3.6s

Here, a tolerance of 0.01 for KKT condition was very easy to satisfy, that why we set it to 0.001, and a better optimum was obtained, again this highlights the tolerance dependency on the problem.

Contour plot may not exactly correspond to density field, this is because components are not plotted in their real size (they are too big because of gray regions), a scaling factor is then used to plot dimensions, this scaling factor needs to be well tuned in order to have an appropriate explicit design.

6) Wing rib optimization

This test case is inspired from the thesis of G.PESARE (2013), where a rib model were optimized for minimum compliance under SIMP approach, design domain three circular voids to represent inspection holes (useful for the passage of electrical cables through the wing) and a square void representing the presence of a spar. The structure is fixed from the left and right edges, and a compression load is applied on both upper and lower edge of the rib (more exactly in some nodes of the edges).

Design domain and boundary conditions are depicted in the following picture:

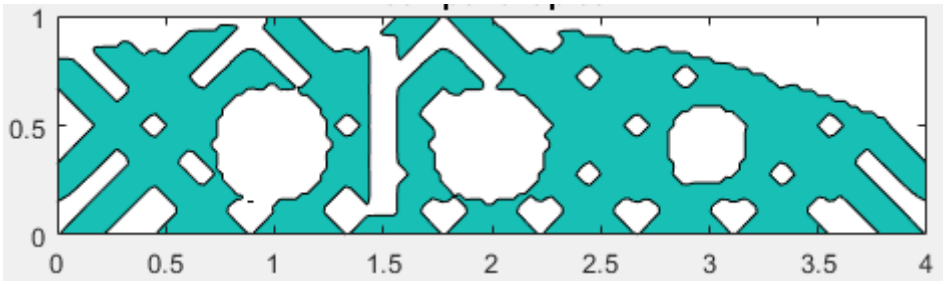
A 160x40 mesh is used, and volume fraction constraint is set to 0.45

Here is the optimal solution under SIMP method

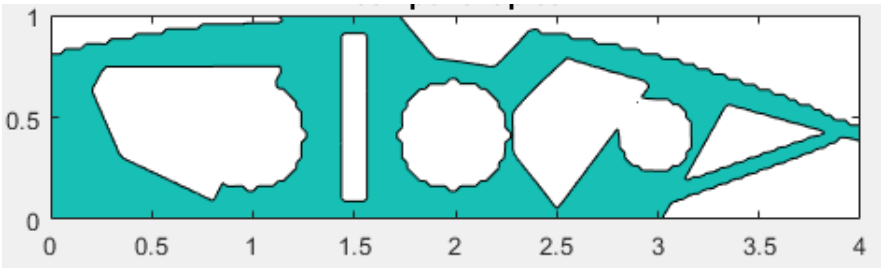
Enhanced MMC code results:

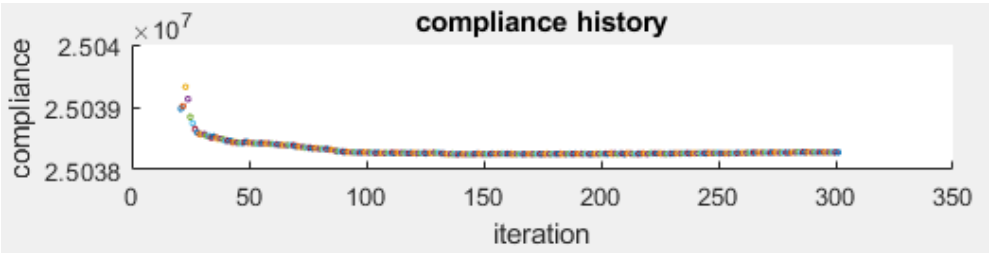
16x4 components are used but all the components that are inside passive regions are removed, hence, 28 components are kept. Straight components here have fixed thickness and only 5 design variables for each component, which gives 28*5=140 design variable.

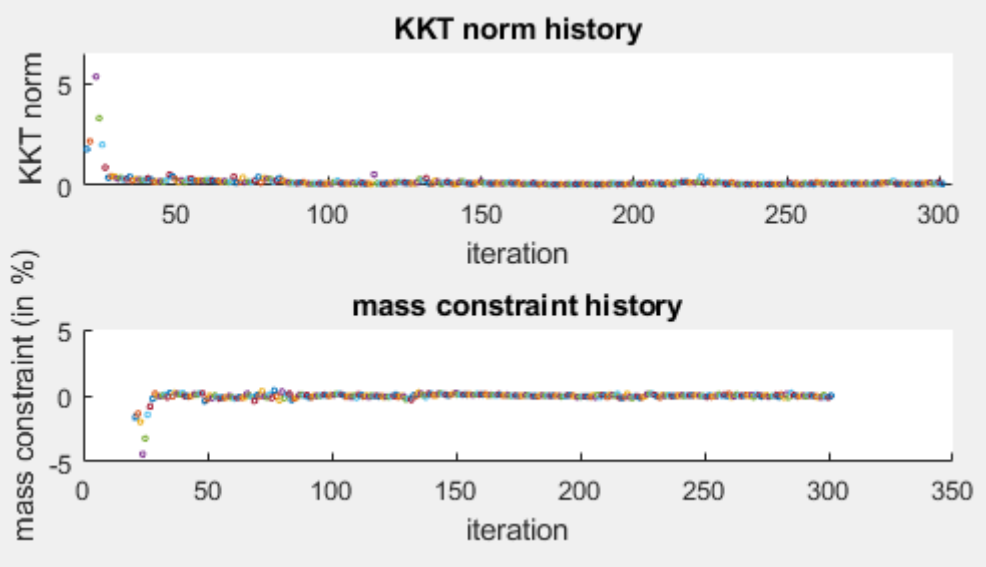
Initial design:



Density field and contour plot of optimal design, and convergence history:





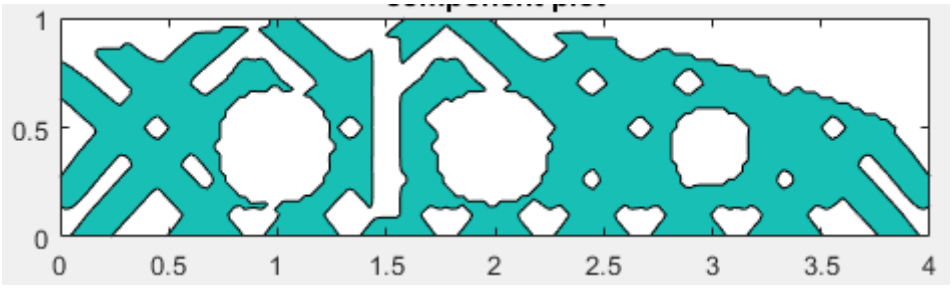


We can notice that the MMC design is a little bit similar to the SIMP one, also the density field of MMC solution has the advantage to show almost no gray regions (intermediate density) compared to the SIMP solution, this is one of the most interesting features of MMC approach.

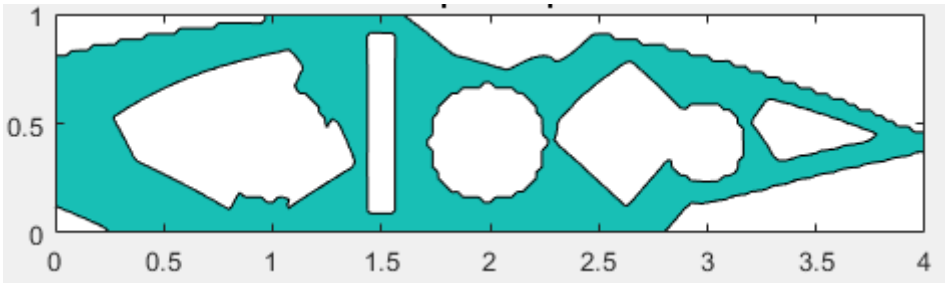
The simulation stopped after 301 iterations, this means that one stopping criterion had been achieved namely the change criterion, the KKT norm is then equal to 0.07 which is close to the tolerance (0.01) mass constraint is respect (there are some violation of around 0.01% so they can be neglected).

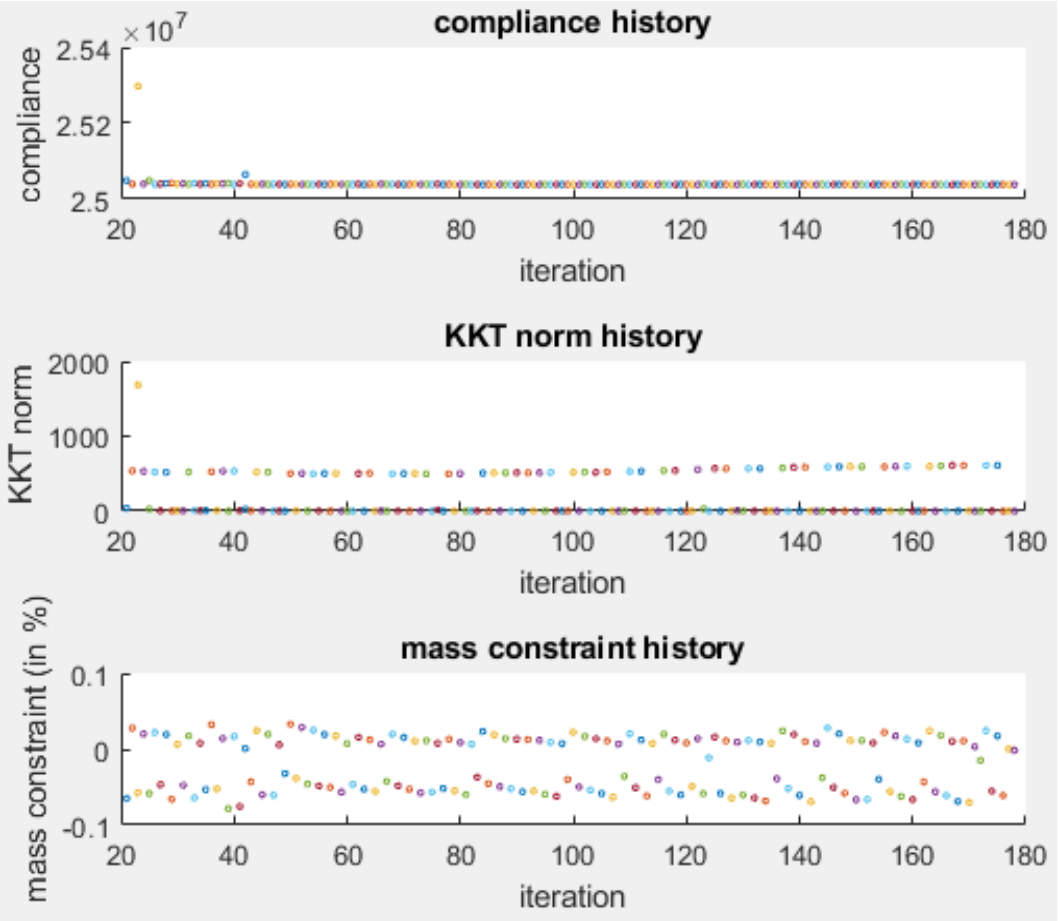
MMC with curvature results

Here again we start with 16x4 components but only 28 are kept, involving 28*6=168 design variables, the initial configuration is shown in figure below:



Density field and contour plot of optimal design, and convergence history:





Optimal design is similar to the one obtained with straight components, although curved features could be seen.

Values of compliance for both straight and curved codes are so big (around 2×10^7) that the relative variation of compliance (at every iteration) with respect to the initial compliance is very small.

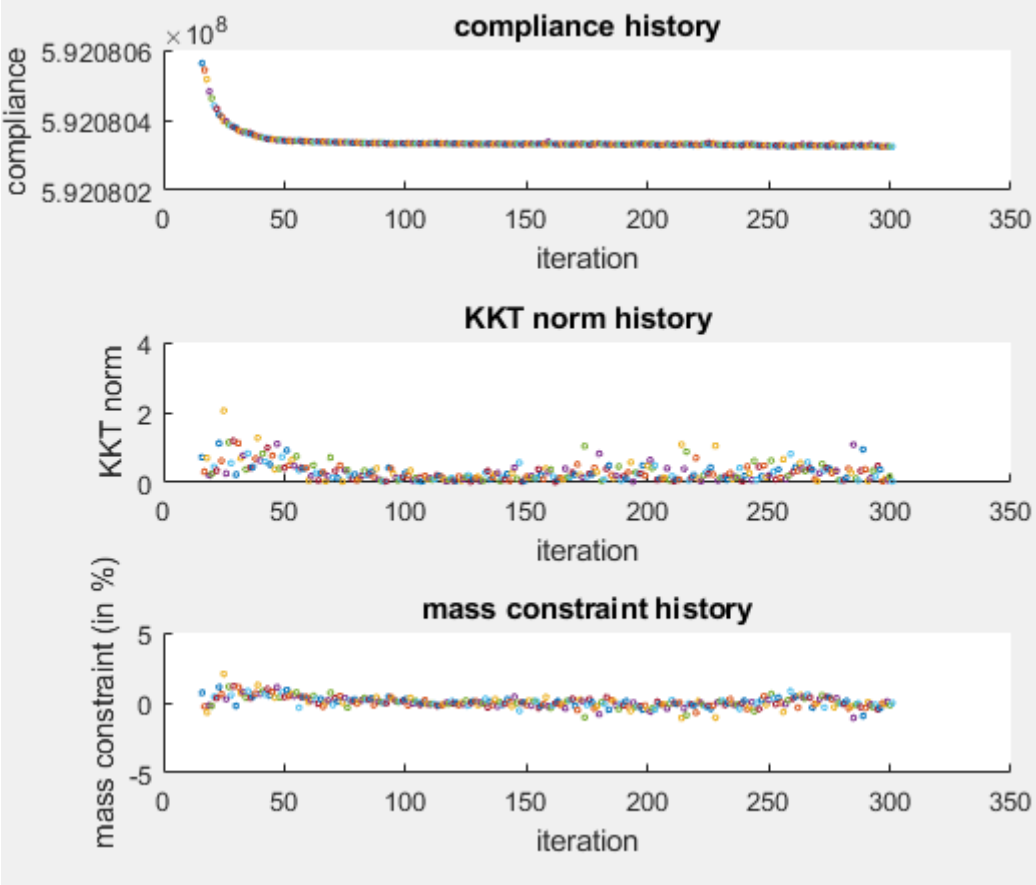
For the curved code, design kept changing but compliance almost didn't change. A possible cause is that stiffness of the structure (regardless of the design) is weak compared to the load (multiple points load), which results into such high compliances that no matter the design, compliance will not vary. The weak stiffness may come from the geometry of design domain, mainly the holes between upper and lower skin where compressive loads is applied.

MNA code results

Here, all the 16x4=64 components were used, since in MNA mass nodes have larger influence regions (larger components) so they can leave passive regions. 64x5=320 design variables are used.

The initial configuration is shown in figure below:

Density field of optimal design, and convergence history:

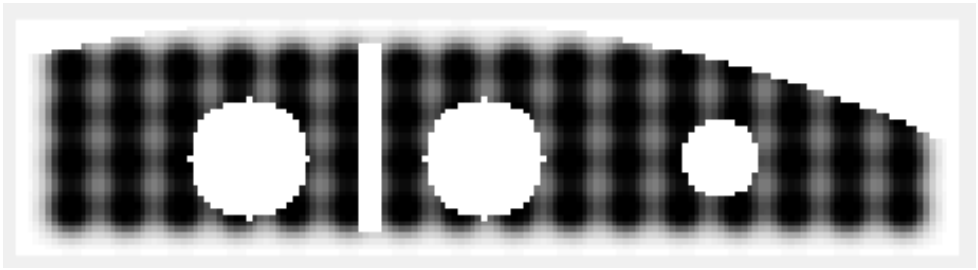


Optimization process stopped after reaching maximum number of iterations (300), yet KKT-norm was equal to 0.02 which is very close to the tolerance 0.01, this means that the code was converging to an optimum. The compliance value given by MNA and SIMP is a lot different from the one given by MMC (5.8×10^8 against 2.5×10^7) while normally they should have same magnitude, although different material laws are used (which is the case for all other test cases). This is maybe due to high load that led to big values of compliance.

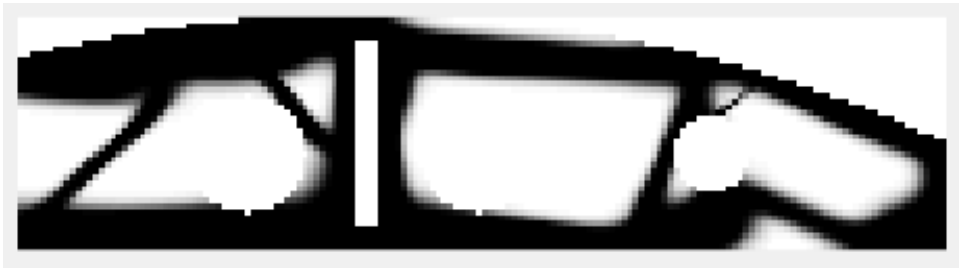
MNA with curvature results

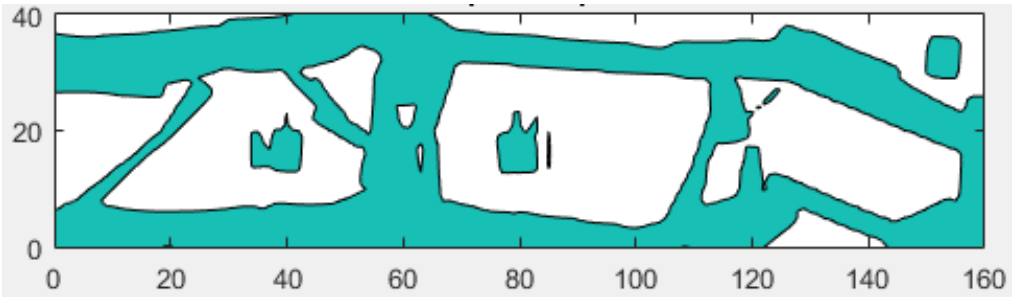
Here again all the 16x4=64 components were used, involving 64*6=384 design variables.

The initial configuration is shown in figure below:

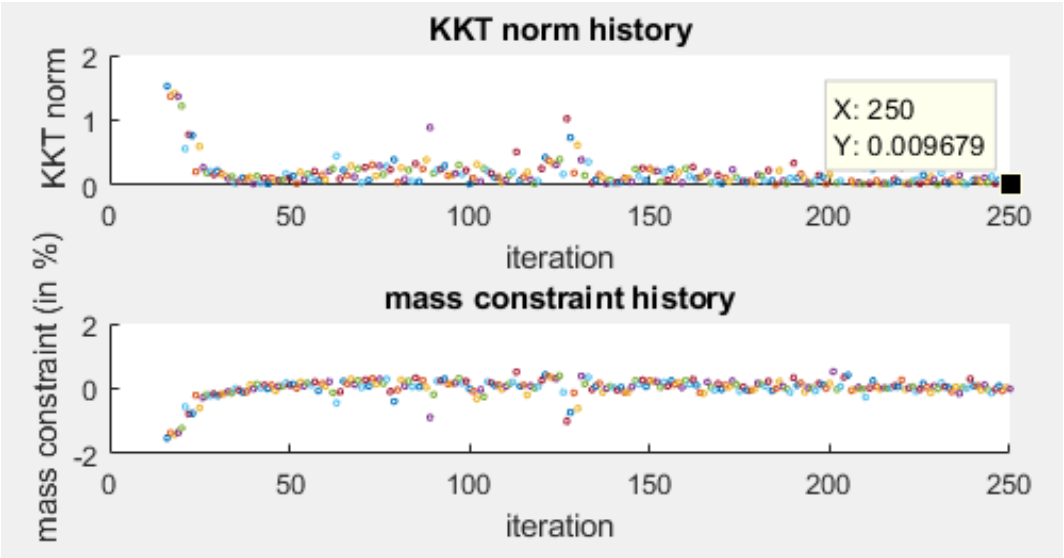


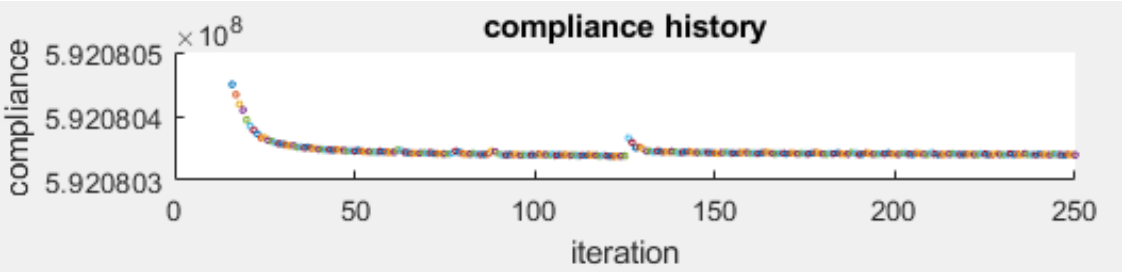
Density field and contour plot of optimal design:





Convergence history:





The simulation converged after 250 iterations, the KKT norm is satisfied and optimal design is close to previous results. Compliance is the same as in SIMP and MNA code with straight components.

Introducing stress constraint:

The topology optimized design is generally followed by a number of post-processing steps to make the design suitable for manufacturing and meet relevant failure criteria, such as stress and buckling constraints. Directly including stress and constraints in topology optimization has been an important field of study because this reduces the gap between the optimized and final design.

The stress constrained topology optimization problem in its nested form is defined as:

$$minimize\ V = \frac{1}{V0}\sum_{e=1}^{n_e}\rho_e v_e$$

$$subject\ to\ g_j = \frac{|\sigma_j|}{\sigma_{\lim}} - 1 \leq 0, \forall j \in \{1 \leq j \leq n_e | \rho_j > 0\}$$

Here, V_0 denotes the total volume of the design domain, v_e is the volume (area in 2D) of a finite element, $|\sigma|$ represents a positive scalar-valued equivalent stress criterion such as the Von Mises stress (used here) that depends on the symmetric stress tensor σ. The equivalent stress is bounded by the allowable stress σ_{lim}. The stress constraints g_j are only defined over the material domain where $\rho_j > 0$.

However, several difficulties arise when including the stress constraint in topology optimization under this form:

- The set of constraint g_j is design dependent (since its defined only for elements with $\rho_j > 0$
- Correct optima are often inaccessible to standard gradient-based optimization, these inaccessible optima are known as ‘singular optima’, they arise in optimization problems that are of the type ‘mathematical programs with vanishing constraints’ (MPVC’s)(Achtziger and Kanzow 2008)
- Stress is a local stress variable, which means a large number of constraints, this can lead to expensive sensitivity analysis

In order to circumvent the aforementioned difficulties, we choose to adopt the unified aggregation and relaxation approach proposed by (Verbart et al. 2017) using the lower bound KS function (Kreisselmeir 1979, Yang and cheng 1996):

$$\psi_{KS}^{l} = \frac{1}{P}\ln\left(\frac{1}{N}\sum_{i=1}^{N} e^{Pf_i}\right)$$

First the constraint g_i is relaxed as following:

$$\bar{g}_j = \rho_j\left(\frac{|\sigma_j|}{\sigma_{lim}} - 1\right) \leq 0$$

It is worth noting that the relaxation allows elements to disappear regardless of the stress constraint; if $\rho_j \longrightarrow 0$ then $\bar{g}_j \longrightarrow 0$, this solves the problem of vanishing constraint and the design-dependent set of constraint (since $\bar{g}_j$ can now be defined for any element).

Then lower bound KS-function is used to aggregate the constraints so that only one constraint is used for the stress, also this aggregation allows that an element may present a stress slightly superior to the limit value, in fact, having $\sum_{i=1}^{N} e^{P\bar{g}_j} \leq N$ doesn't necessary imply that $\forall 1 \leq i \leq N\ \bar{g}_j \leq 0$.

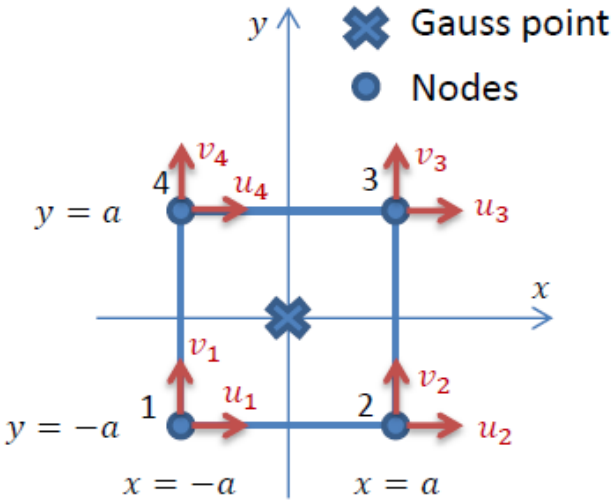
The stress constraint is then written as:

$$G_{KS}^{l} = \frac{1}{P}\ln\left(\frac{1}{N}\sum_{i=1}^{n_e} e^{P\bar{g}_i}\right) \leq 0$$

$$with\ \bar{g}_i = \rho_i\left(\frac{|\sigma_i|}{\sigma_{lim}} - 1\right)$$

Now, we present how the stress tensor in each element is computed.

Four-node quadrilateral elements are used in FE analysis, with one integration point (Gauss point) at the center of each element, an element geometry, nodes and degrees of freedom are shown in the following figure:



Bilinear shape functions are used to interpolate the displacement field:

$$u(x,y) = \sum_{i=1}^{4} N_i(x,y)u_i \quad and \quad v(x,y) = \sum_{i=1}^{4} N_i(x,y)v_i$$

$$with \qquad N_i(x,y) = \frac{1}{4}\left(1 + \frac{x_i}{a^2}x\right)\left(1 + \frac{y_i}{a^2}y\right)$$

Under the small deformations assumption, the linearized tensor of deformations is calculated as:

$$\begin{cases} \varepsilon_x(x,y) = \dfrac{\partial u}{\partial x}(x,y) \\ \varepsilon_y(x,y) = \dfrac{\partial v}{\partial y}(x,y) \\ 2\gamma_{xy}(x,y) = \dfrac{\partial v}{\partial x}(x,y) + \dfrac{\partial u}{\partial y}(x,y) \end{cases}$$

At the Gauss point $x = y = 0$, and deformations are:

$$\begin{cases} \varepsilon_x = \displaystyle\sum_{i=1}^{4} \frac{\partial N_i}{\partial x}(0,0)u_i \\ \varepsilon_y = \displaystyle\sum_{i=1}^{4} \frac{\partial N_i}{\partial y}(0,0)v_i \\ 2\gamma_{xy} = \displaystyle\sum_{i=1}^{4} \frac{\partial N_i}{\partial x}(0,0)v_i + \frac{\partial N_i}{\partial y}(0,0)u_i \end{cases}$$

$$with \quad \frac{\partial N_i}{\partial x}(0,0) = \frac{x_i}{4a^2} \quad and \quad \frac{\partial N_i}{\partial y}(0,0) = \frac{y_i}{4a^2}$$

We note $q = (u_1\ v_1\ u_2\ v_2\ u_3\ v_3\ u_4\ v_4)^T$ the vector of element degrees of freedom, and deformations at the Gauss point can be expressed as:

$$\begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} = \frac{1}{4a}\begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 & 0 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \end{pmatrix} q = Bq$$

Constraint tensor is given by Hooke's law in 2D linear elasticity:

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = D\begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} = DBq \ \ with \ \ D = \frac{E}{1-\nu^2}\begin{pmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{pmatrix}$$

Where E is the Young modulus of material and ν is the Poisson ration. And Von Mises equivalent stress is given by:

$$\sigma_{VM} = \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x\sigma_y + 3\tau_{xy}^2}$$

$$\sigma_{VM} = \sqrt{(\sigma_x \quad \sigma_y \quad \tau_{xy}) \begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix}} = \sqrt{q^T B^T D \begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix} DBq} = \sqrt{q^T S q}$$

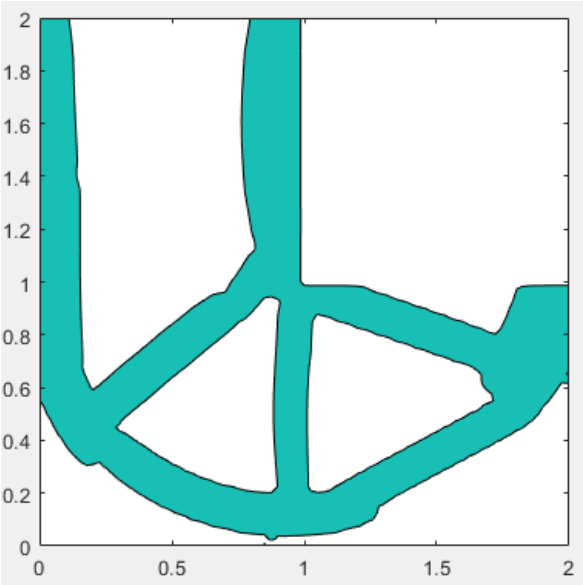
$$with \ S = B^T D \begin{pmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{pmatrix} DB$$

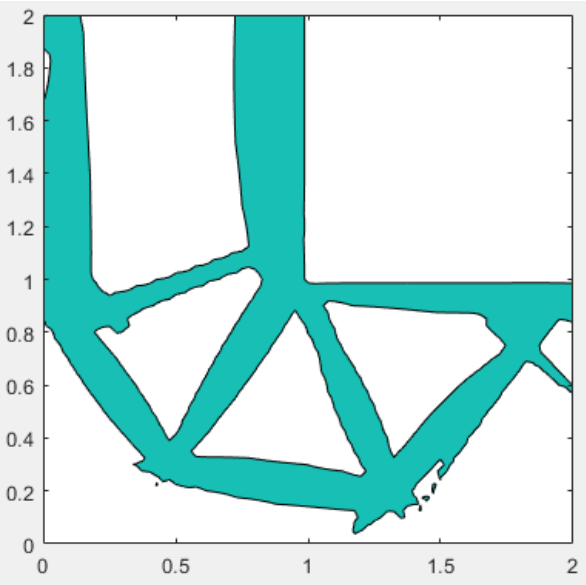
Stress constraint is tested on the L-shape test case for both MMC and MNA.

Example on MMC

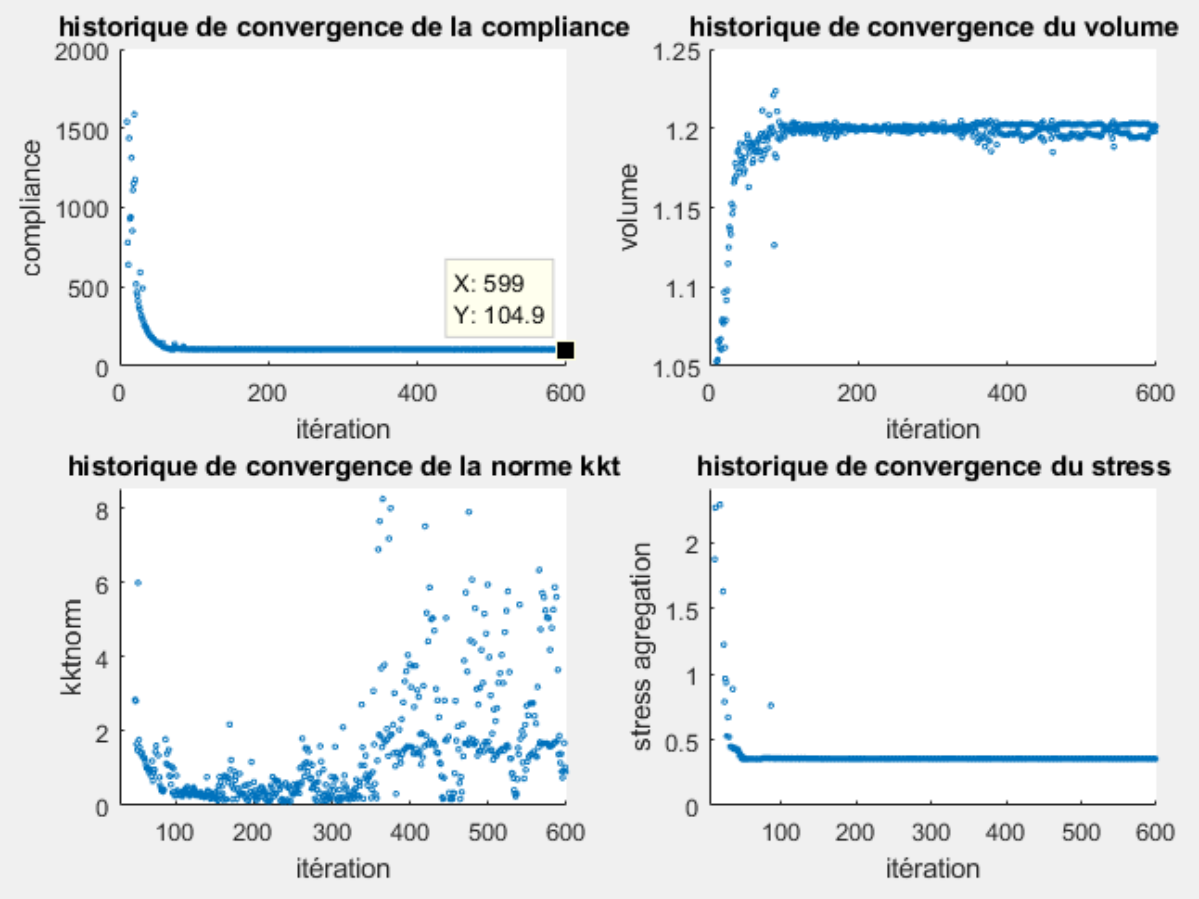
Here are results of the L-shape example for both codes with curvature (right) and without (left), volume fraction is 0.4 and stress limit is equal to 0.6.

Components plot:

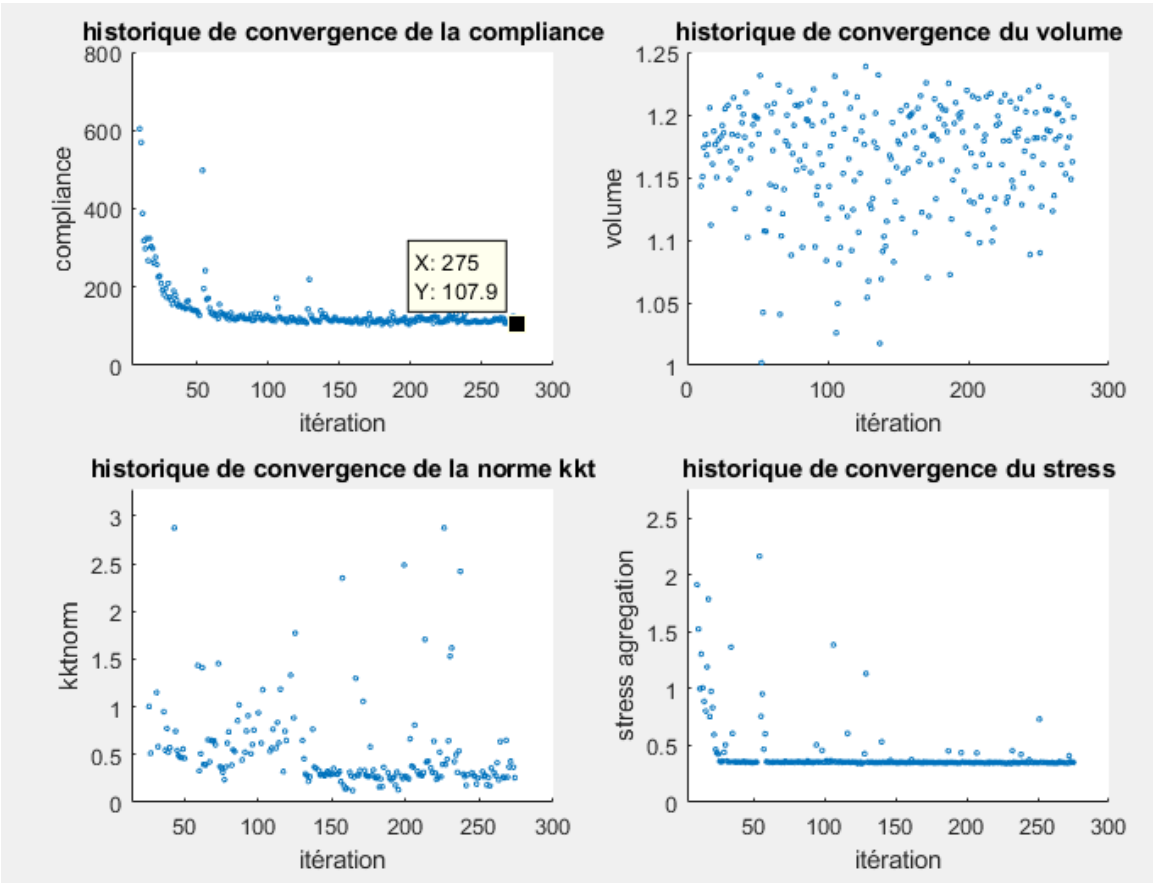




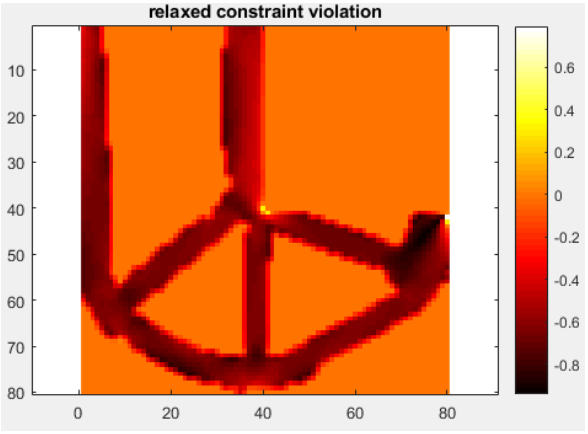
Convergence history (curved components):

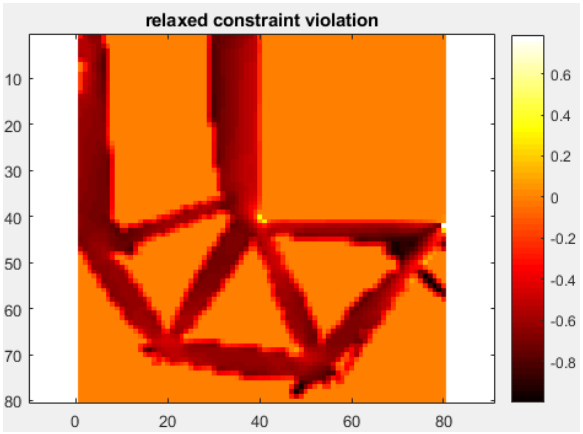


Convergence history (straight components):



Relaxed stress constraint:



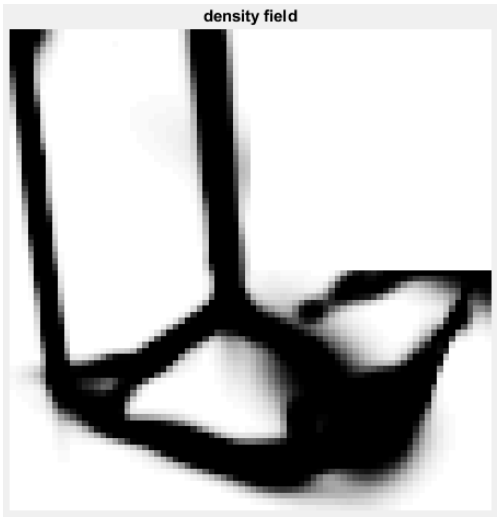


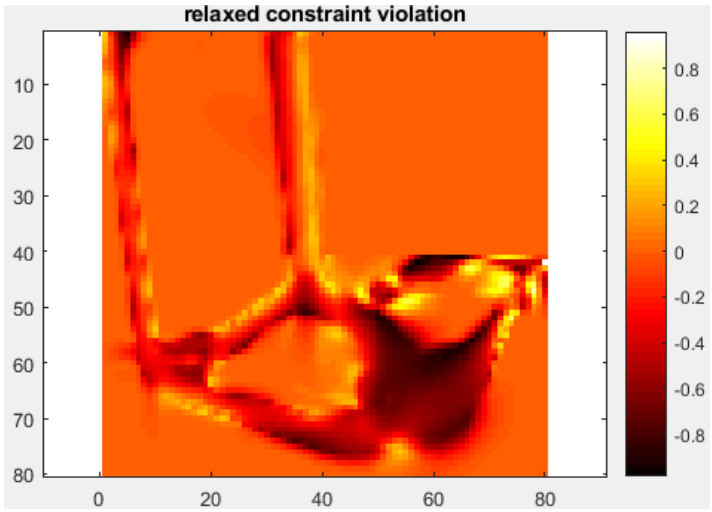
Curved features can be clearly seen in the final design, even though initial curvature was equal to zero for all components (initially straight). This is very different from the case without the stress constraint where components remain straight in the final design.

The aggregated stress constraint converges to a value of 0.35 for both designs, and the curved solution shows a better compliance (104.9 against 107.9 for straight components).
From the stress plot, we can see that more yellow regions appear in the straight solution, those are regions where the stress is higher and constraint is violated. Hence, curvature helped finding better design in terms of stress and compliance in this case.

Example on MNA

Density field and stress violation distribution:

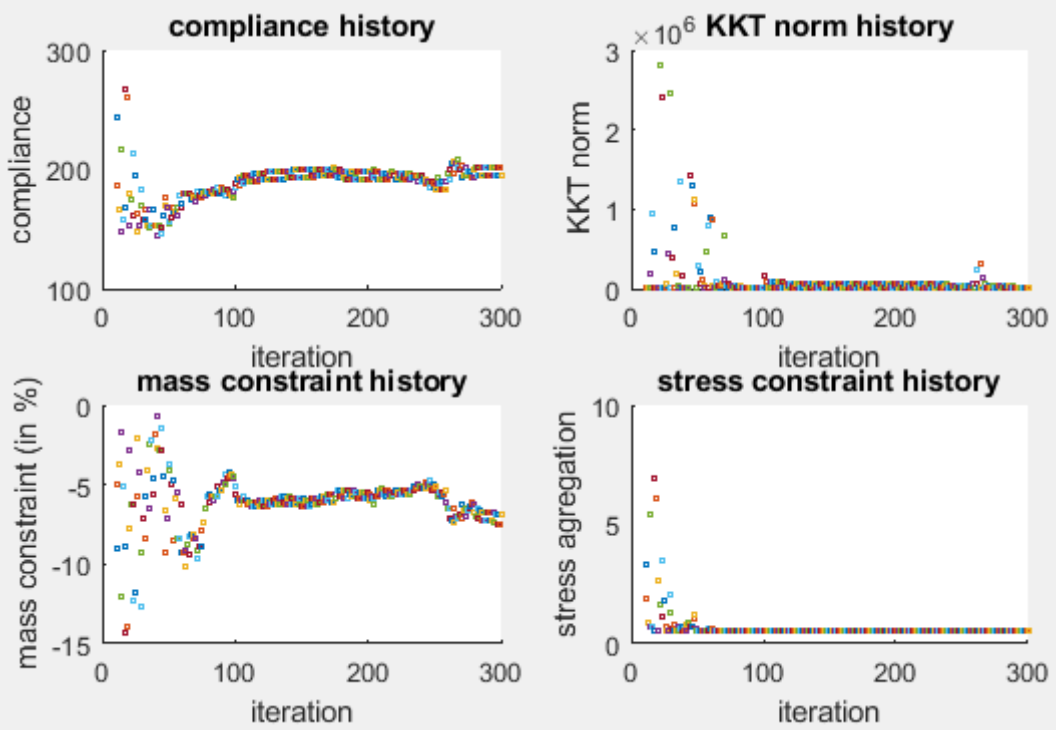




Compared to MMC results, there is less constrations of the constraint, since sharpe features are naturally avoided in MNA. However constraint violation is a little bit bigger and located almost every where near the borders of the structure, probably because of gray regions in MNA.

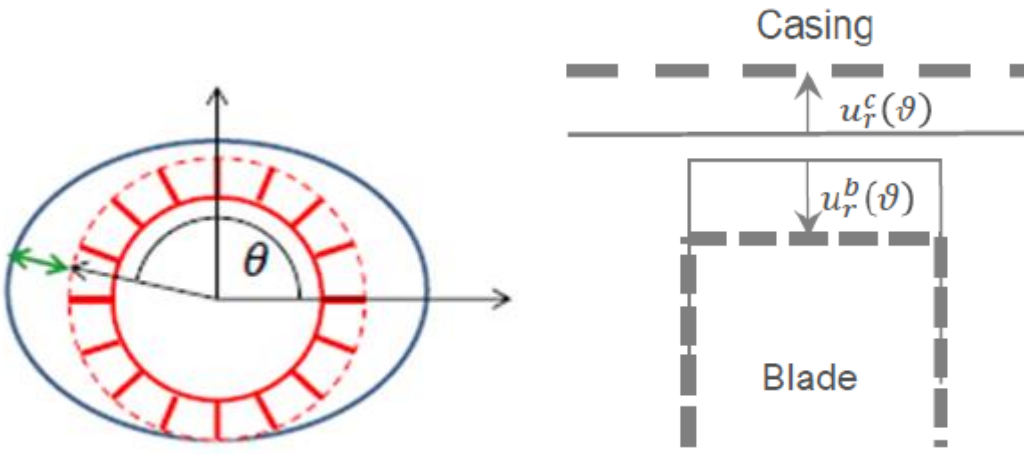
This problem could be tackled by adding more components to the design, in fact not only it will improve results in terms of compliance and stress constraint violation, but also it will decrease the amount of gray regions in the final design, since more overlapping of components is possible.

Convergence history:



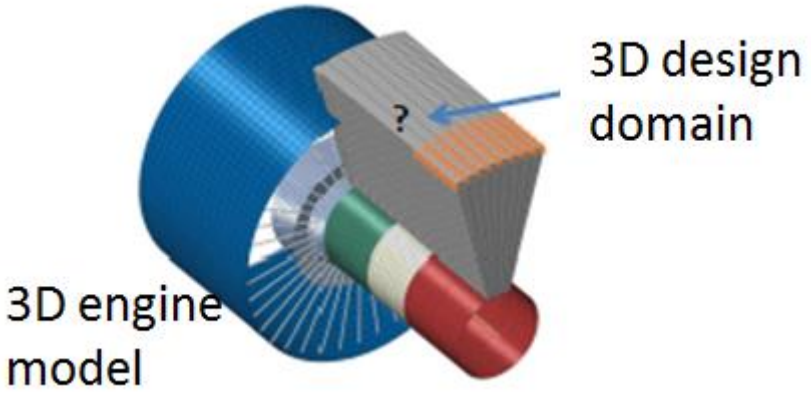
Airbus project:

One of the goals of the internship was to apply explicit topology optimization approaches to an industrial test case taken from (S.CONIGLIO thesis) as part of an airbus project, which concerns the optimization of the engine-pylon attachment structure in order to minimize fuel consumption. Relative displacements of the blades to the casing of the engine (during flight), called tip clearances appear to increase the fuel consumption. Following figures show casing and blades deformations:

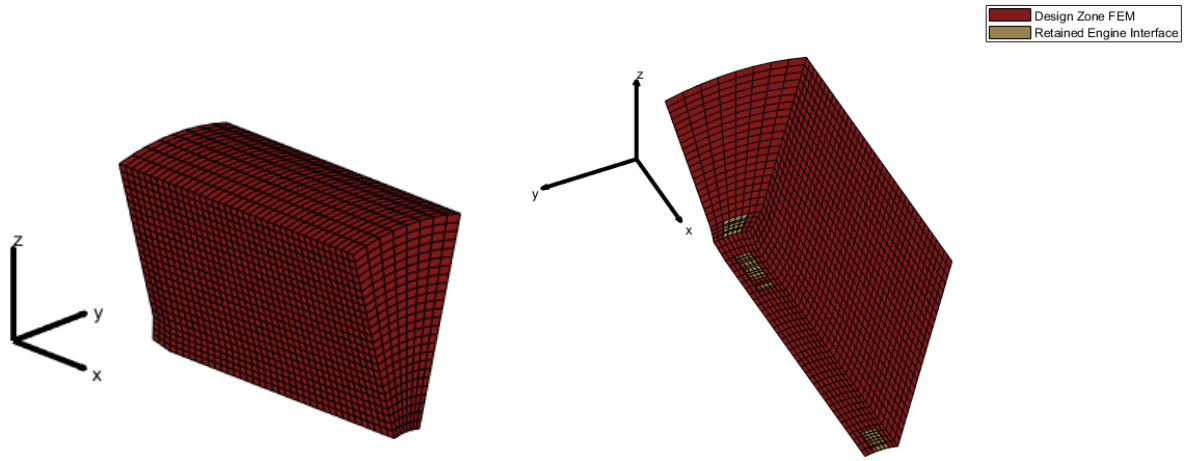


Thus the thrust specific fuel consumption (TSFC) is introduced as a measure of tip clearances, and considered as a function of interest in the optimization process, along with compliance stress and mass (or volume).

For confidential reasons, the real engine model (from Rolls Royce) was not used in our work; instead a 3D model under ABAQUS software is used. The design domain mesh is also in 3D and under ABAQUS, using B-Spline surfaces, and then all necessary data are exported to MATLAB where optimization is carried out, for more details we refer the reader to (S.CONIGLIO 2018). The geometry and boundary conditions of the 3D model are depicted in following picture:



Hypothesis of pure axial thrust load on the engine is applied. Moreover, the load is static (which corresponds to steady state conditions). The wing is considered as fixed and the engine architecture is unchanged. These hypotheses allow a static condensation of the engine model. The static condensation allows the reduction of degrees of freedom in FEM analysis, since the engine model contains a lot of degrees of freedom and its stiffness matrix is unchanged during optimization process, it could be condensed into the interface (between engine and design domain) degrees of freedom, the irregular meshing of the design domain and the retained engine interface are shown in the two figures below:

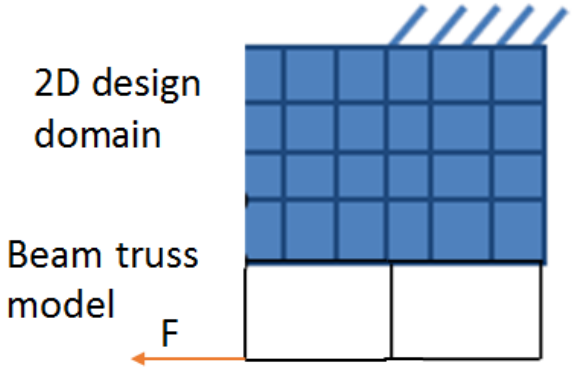


The non-regularity of the mesh arises some challenges:

- It's no longer possible to identify coordinates of a node or an element directly from its ID number (in the connectivity table), as a result, the definition of loads and boundary conditions becomes harder (compared to a regular mesh).
- Elements have different geometries and shapes, therefore different elementary stiffness matrix, and should be calculated for each element, this causes some difficulties when adapting MMC method to the code, for instance the computation of elementary energy matrix.

- The components plot! The MATLAB functions used to make the components plot need as input, the coordinates and TDF matrices in a format similar to that given by 'meshgrid' function, which is impossible without a regular mesh. Thus another way should be found in order to plot components.

In addition to this 3D model, a 2D model of both design domain and engine is considered, here the engine is represented by a beam truss model and design domain is discretized by a regular mesh, which makes the implementation of MMC and MNA approaches easier. The following picture shows geometry, load and boundary conditions of the 2D model:



A possible formulation of the optimization problem could be minimizing TSFC function under constraints of compliance, volume and stress:

$$\min \Delta TSFC, \quad st \begin{cases} \dfrac{V}{V_0} \leq V_f \\ Gks\big((\sigma_i), \sigma_l\big) \leq 0 \\ C \leq C_0 \end{cases}$$

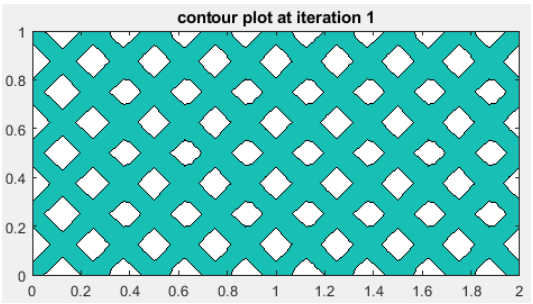
2D case results

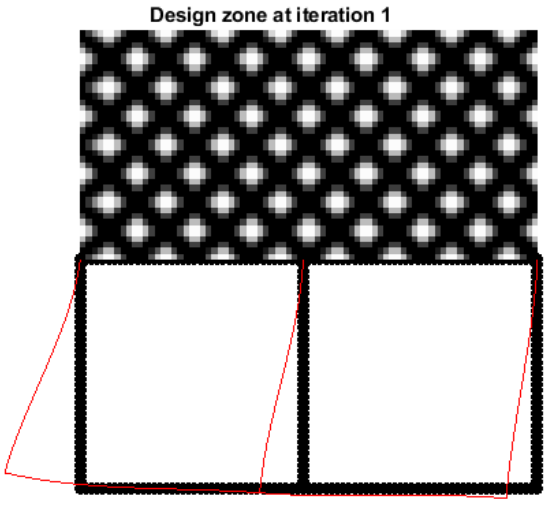
First we present results of the 2D model under our MMC and MNA codes and compare them to existing results under SIMP approach.

MMC results

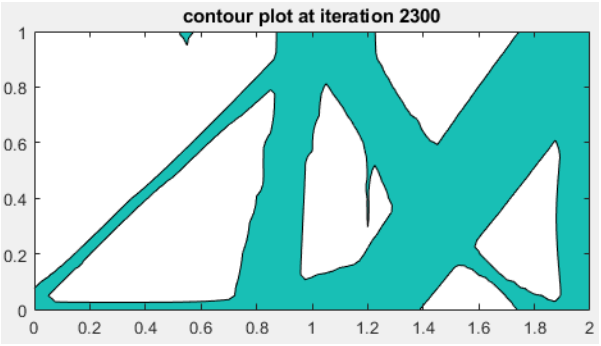
Results of MMC code with straight components are presented in following figures.

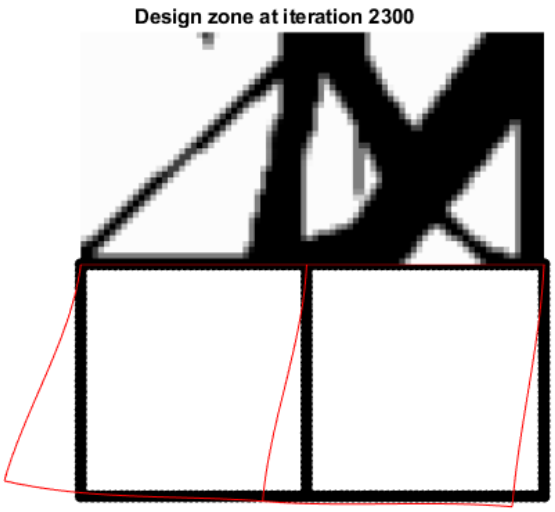
Initial configuration:



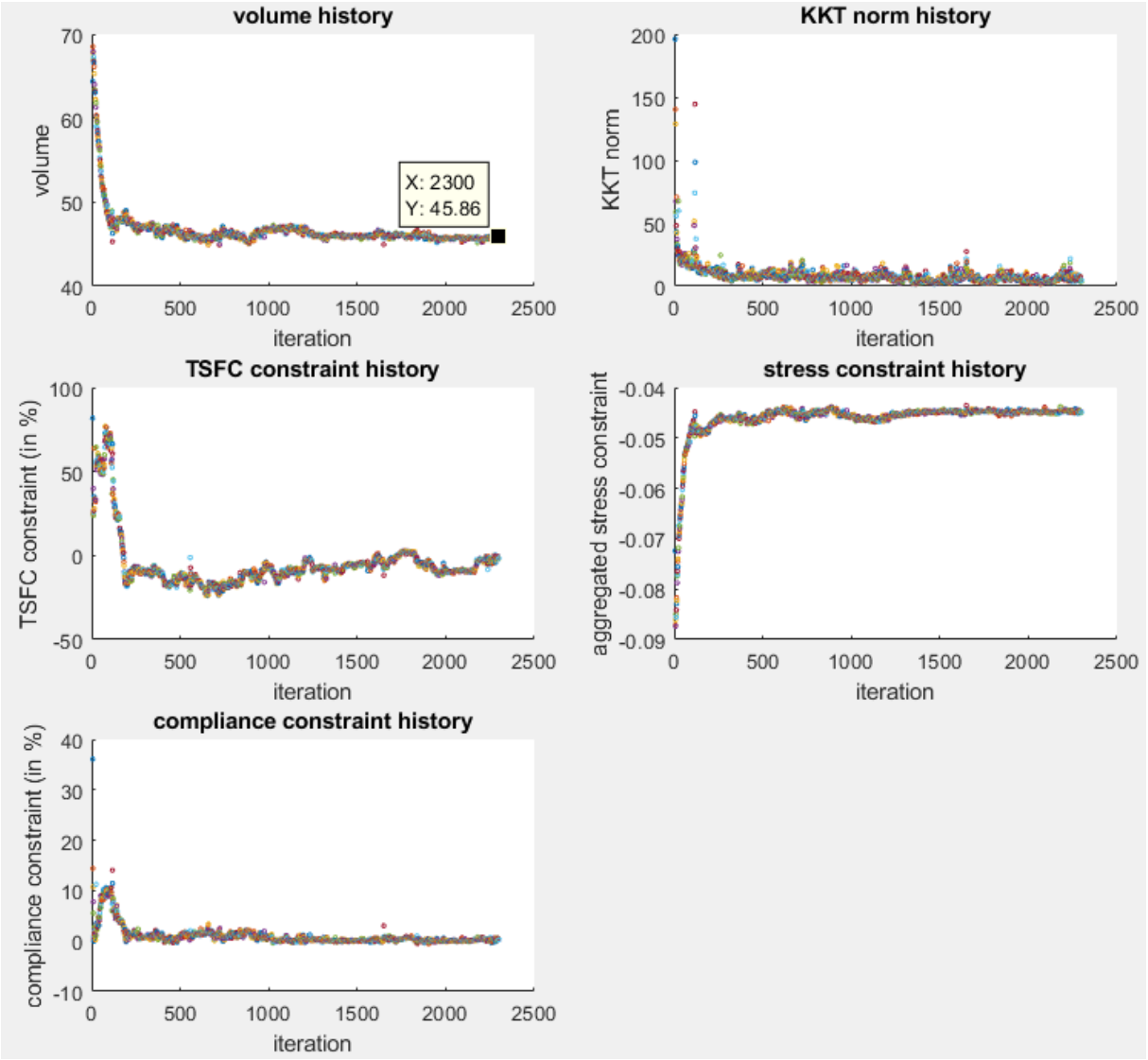


Contour plot (left) and density field (right):





Convergence history:



16x8=128 components were used leading to 128x7=896 design variable. After approximately 200 iterations all the constraints are respected, KKT norm start with big values (when constraints are

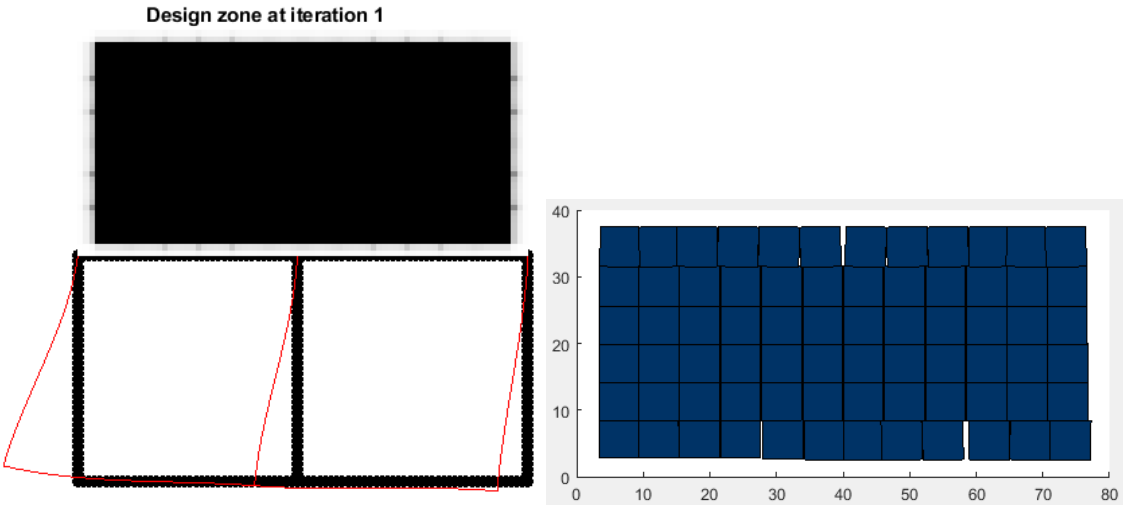
violated) and then decrease and keeps oscillating between 2 and 5, while usually it oscillates between 0.01 and 0.1 (that's why KKT tolerance is set to 0.01). In fact, for the last 1000 iterations (or more) the range of variation of KKT norm didn't change, the simulation stopped after reaching the maximum number of iterations (here 2300), and even if we go further this value the optimization process will not converge because simply the KKT tolerance is unachievable for this problem and a bigger tolerance should be used instead. This test case shows that the value to put as a KKT tolerance depends on the considered test case.

The design obtained by MMC and the optimal solution (volume, 45%) are very similar to those obtained under MNA and SIMP approach, which proves the efficiency of MMC formulation for this test case also.
The simulation lasted 11046s (3 hours and 4 minutes). We should note that first iterations were very fast (around 1s) compared to last ones (around 4s)

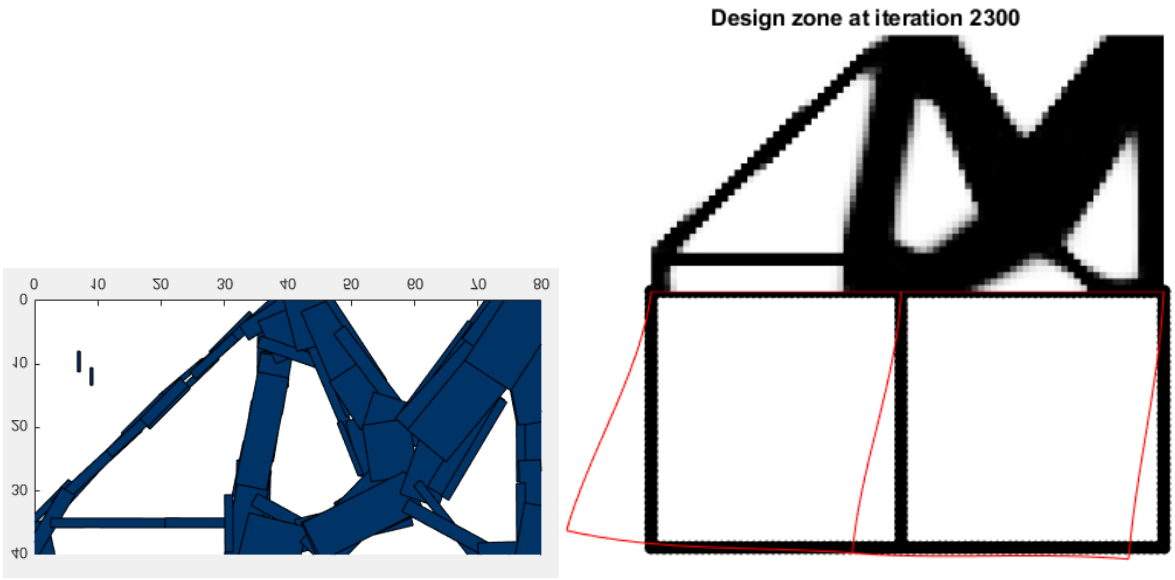
MNA results

Here 12x6=72 components were used with 72x5=360 design variables. Results of MNA code with straight components are presented in following figures.

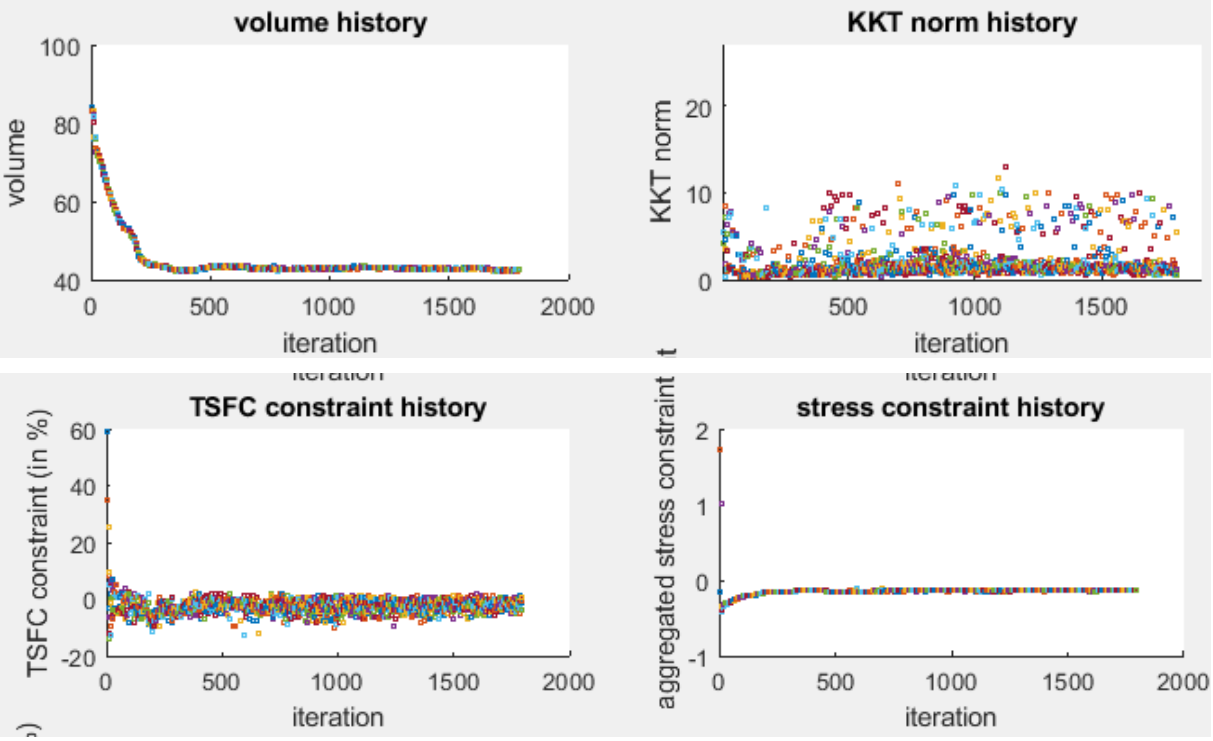
Initial design:



Component plot and density field of the optimal design:



Convergence history:

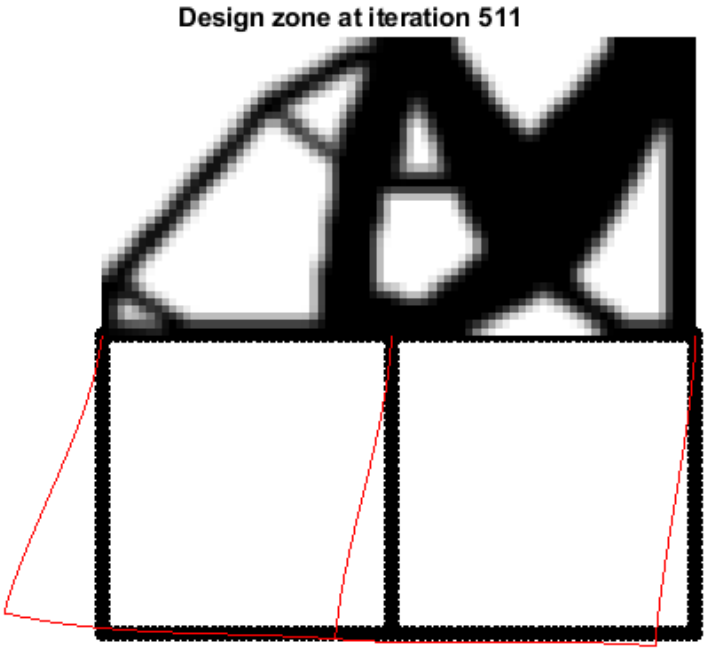


Same remarks can be enounced as for MMC case, design is very close to that obtained by SIMP and MNA, the final volume fraction is 42%, which better than MMC, and of course, at the end all constraints are satisfied.

SIMP results:

For the sake of comparison, we give results of the SIMP approach. These results are considered as reference for explicit approaches.

Density field:

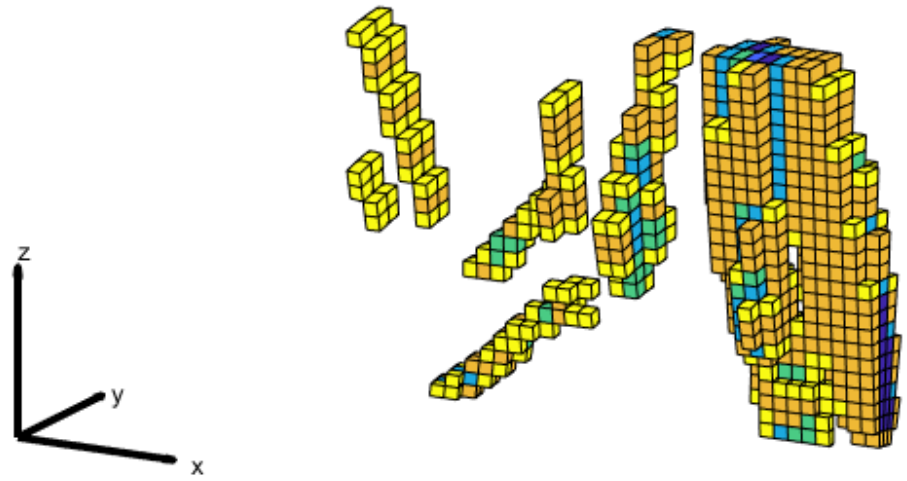


The simulation was stopped after 511 iterations because volume was no more decreasing, constraints are all satisfied and minimal volume is 50.6%. It is worth noting that better optimums were achieved by both explicit approaches MNA and MMC.

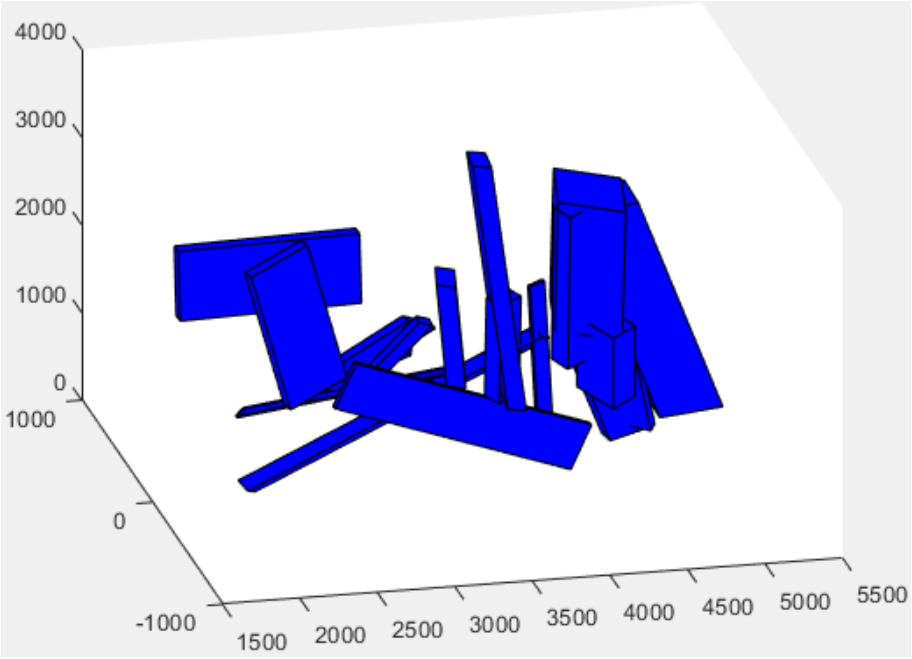
3D case results

MMC results

Density field obtained by MMC code (straight)



As mentioned before, we couldn't obtain a proper contour plot because of the irregularity of the mesh however, we could still make a components plot (figure below) from design variables using the Matlab "patch" function.



Contrary to the 2D case where the objective function was the volume, here we minimize the compliance under constraints of aggregated stress, TSFC and volume constraints

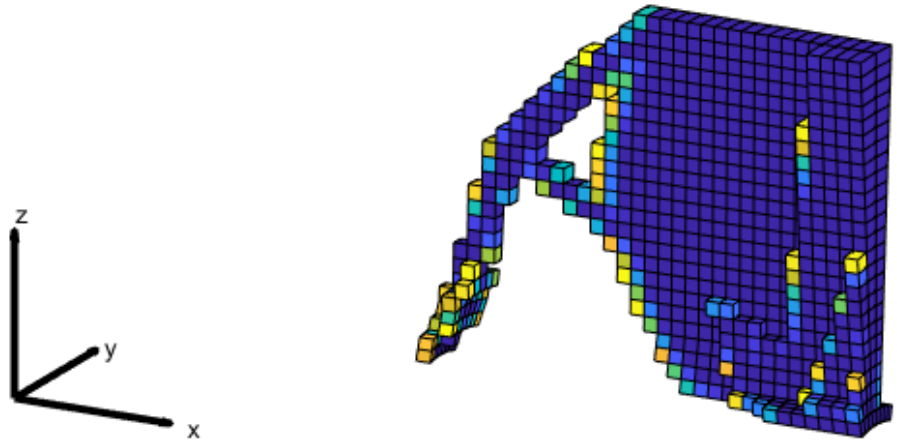
The code was stopped after 237 iterations, and results are presented in the following table:

function	compliance	TSFC	volume	GKSl (stress)
Value at iter. 237	48640	0.167	0.099	0.188
Admissible value		0.15	0.1	0

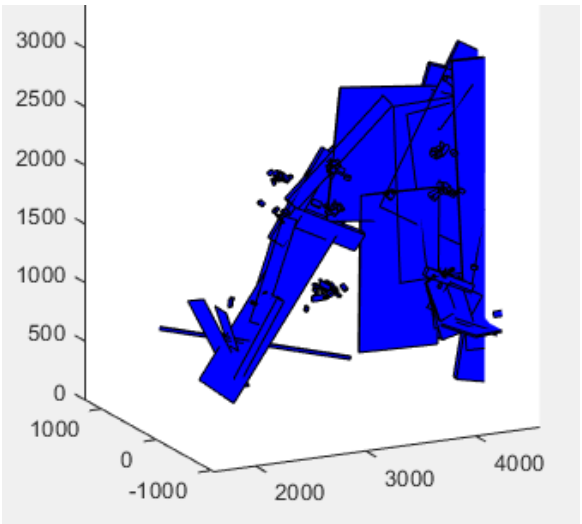
Results are not satisfying, only volume constraint is respected, and compliance value is equal to 48640 while it decreases to around 5000 for SIMP and MNA. This hard convergence is due to the fact that MMC doesn't allow gray regions which facilitate the convergence. This last point will be explained in detail in a following section.

MNA results:

Density field:



Component plot:



The code was stopped after 400 iterations, and results are presented in the following table:

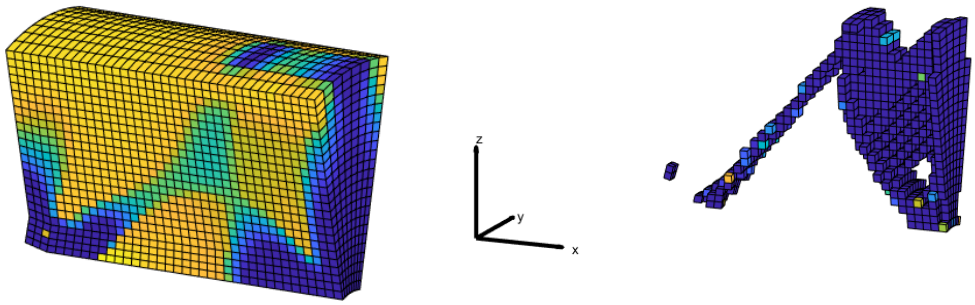
function	compliance	TSFC	volume	GKSl (stress)
Value at iter. 400	5105	0.149	0.1	-0.031
Admissible value		0.15	0.1	0

All constraints are satisfied and compliance value is close to that given by SIMP, even though design is a little bit different. 5x5x5=125 components were used, each component having 9 design variables, a total of 125x9=1125 design variable are used. The simulation took approximately 8s per iteration, this could still be improved by vectorizing the *for* loops involved in the density field computation.

SIMP results:

For the sake of comparison, original results obtained under SIMP approach are shown below.

Initial configuration and final design:



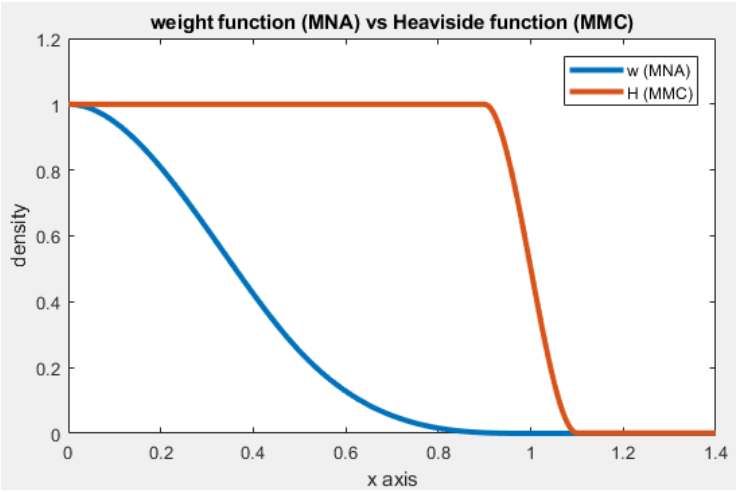
The code was stopped after 230 iterations, and results are presented in the following table:

function	compliance	TSFC	volume	GKSl (stress)
Value at iter. 400	4611	0.15	0.1	-0.028
Admissible value		0.15	0.1	0

The average CPU time per iteration is 4.5s

Discussion on MNA/MMC differences

MNA and MMC approach are both explicit, they allow the same reduction of design variables number, yet a fundamental difference between the two approaches is that the polynomial function used for density field in MNA drops quickly from 0 to 1, and thus allow much gray regions than MMC where the Heaviside function is equal to 1 everywhere inside the component, except for a small regions of size 2ϵ where it decreases to 0. Following figure plots both the weight function of MNA and Heaviside function of MMC.



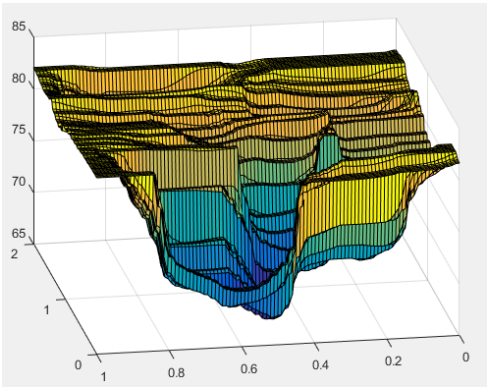
This fundamental difference leads to several repercussions:

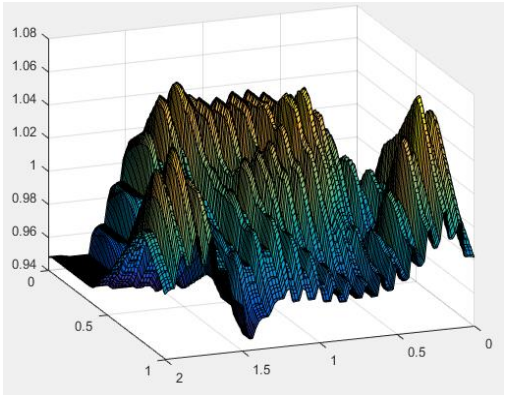
- Better and easier convergence for MNA
- Better control over the structure's shape for MNA
- Design gets connected quickly
- Structure become disconnected progressively

These points could be justified by following:

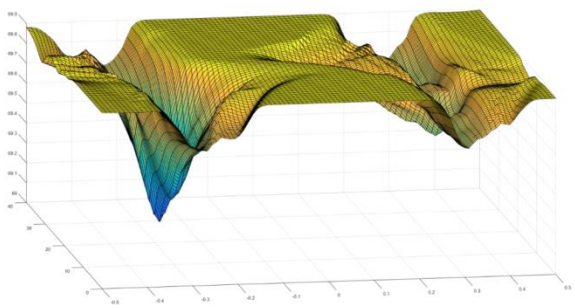
- More gray regions in MNA than MMC → for the initial same volume/mass: the region of influence of MNA's mass nodes (length and thickness of MNA's components) are bigger than MMC's components (in practice variables L and t have bigger values in MNA than MMC) → components can meet each other quickly and easily in MNA than MMC → the structure gets connected quickly.
- Also, in MMC, in order to connect an initial configuration that is disconnected, the optimizer has tendency to increase lengths and thickness without moving the components (because their derivatives are higher) this requires adding a lot of mass in MMC compared to MNA.
- In MMC a small perturbations can disconnect a structure, leading to strong variations in compliance, this not the case in MNA where components separate progressively and thus no strong variations on compliance are induced by small perturbations.
- this facilitates convergence,(that's why MMC needs more iterations to converge) since objective function and constraint are smooth in MNA and not in MMC, following figures shows response surfaces for both compliance and volume with respect to two random design variables:

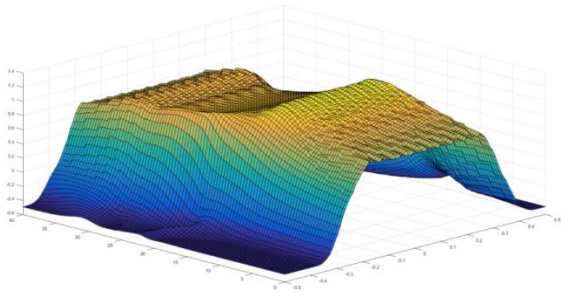
In MMC:





In MNA:

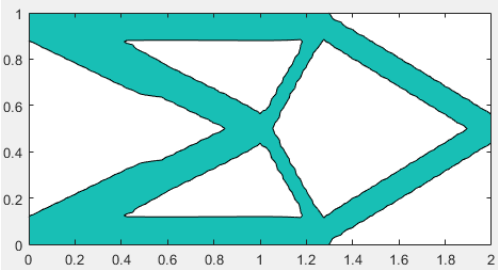


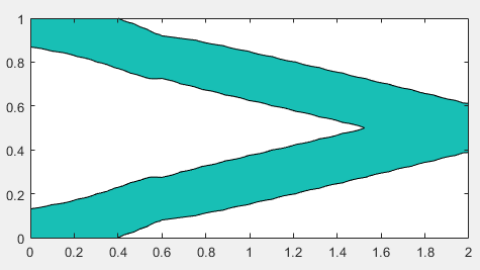


Other repercussions were discussed in the section of test cases. Besides, some remarks that have been noted within the different test cases could be mentioned:

- the ability to reproduce the same optimal solution regardless of the initial design is better for MNA than MMC
- MMA external move limits are very important parameters that needs to be tuned: small values can lead to slow convergence and thus a big number of iterations, in that case if a change stopping criterion is used, it can be easily satisfied giving a non-optimal design, in fact the criterion is satisfied not because the design is getting optimal but just because external move limits doesn't allow enough change. In the opposite case; if external move limits have big values changes (perturbations) become so big that it's hard to find a good optimum (strong oscillations occur near optimum), and the optimizer has more tendencies to give local optimums. To show this effect, we had a look again on the cantilever test case, this time we change the external move limits to bigger values (from 0.05 to 0.1)

  Optimal design (left) and final design (right):





  One can notice that final design is very different from initial one, indeed final compliance is equal to 92.9 while the optimal one is equal to 74.9. This can be compared with the results of previous section where external move limits was equal to 0.05, in that case final and optimal design were very similar and the optimizer didn't leave global optimum to a local one.

- components change according to design variables which have the most important derivatives, in addition, derivatives with respect to center's position $\frac{\partial .}{\partial x_0}$ $\frac{\partial .}{\partial y_0}$, length $\frac{\partial .}{\partial L}$ and thickness $\frac{\partial .}{\partial t}$ are in general bigger than those with respect to curvature $\frac{\partial .}{\partial cr}$ that's why curved patterns are more luckily to happen within few initial number of components.
- using finite differences in MMC is problematic, in fact, parameter alpha which represents the void density, if this parameter is too small the error between analytical gradients and finite differences increase, on the other hand too high alpha creates unrealistic void stiffness
- MMC has more ability to explore design domain since it can leave local optimum easier than MNA while MNA has more risk to get stuck in a local minimum.
- MMC is very accurate gives good results, but needs to be tuned (MMA optimizer) and doesn't work for all test cases, while MNA is "softer", less need for tuning and works for all test cases.
- The structure of better optimums is often constituted of higher number of "structural components". This is why MMC finds good optimum (in some cases) easier than MNA (when same number of initial components is used) especially that MNA needs by default to overlap components in order to create full density regions in a black and white design.

Conclusion:

During this internship, we handled explicit topology optimization methods MMC and MNA, and applied them on a bench of academic test cases as well as industrial ones. We tried through these applications to understand different features of both methods and explain the results. The results

were compared to reference solution given by the SIMP approach. Efficiency of both MMC and MNA methods had been proved, in addition, we tried to understand the limitations of each method, the difference between MMC and MNA as well as resulting characteristics of each approach.

As for the future perspectives of this work, computing sensitivities analytically may be investigated for the MNA with curvature code. Also, a better way to plot the contour of the design needs to be developed, this important for exploiting the full potential of the explicit geometrical description of the structure's topology, in MNA as well as in MMC when irregular mesh is considered. Furthermore, a multi-grid resolution approach could be developed under MNA formulation to improve its efficiency.

References:

A 99 line topology optimization code written in Matlab, (O.Sigmund 2001)

A Moving Morphable Void (MMV)-based explicit approach for topology optimization considering stress constraints (Guo et al 2018)

A new three-dimensional topology optimization method based on moving morphable components (MMCs) (Guo et al 2016)

A new topology optimization approach based on Moving Morphable Components (MMC) and the ersatz material model (Guo et al 2015)

An efficient MMC-based approach for multi-resolution TO (Guo et al 2018)

Doing Topology Optimization Explicitly and Geometrically-A New Moving Morphable Components Based Framework (Guo et al 2014)

Efficient topology optimization in MATLAB using 88 lines of code (O.Sigmund et al 2010)

Explicit structural topology optimization based on moving morphable components (MMC) with curved skeletons (Guo et al 2016)

Feature-driven topology optimization method with signed distance function (Guo et al 2016)

Improving multiresolution TO via multiple discretizations (Paulino et al 2012)

the method of moving asymptotes (K.Svanberg 1987)

Some modelling aspects for the Matlab implementation of MMA (K.Svanberg 2004)

Simultaneous shape and topology optimization of shell structures (Hassani et al 2013)

Explicit TO through the moving node approach: beam elements recognition largely inspired by Overvelde's thesis (Morlier et al 2018)

Aeroelastic Topology Optimization of Blade-Stiffened Panels (B.Stanford et al. 2014)

Design of cellular based structures in sandwiched morphing skin via topology optimization (Chang & Shen 2018)

DECOMPOSITION-BASED ASSEMBLY SYNTHESIS OF STRUCTURAL PRODUCTS: PRELIMINARY RESULTS (Saitou & Yetis)

General topology optimization method with continuous and discrete orientation design using isoparametric projection (Nomura et al. 2014)

Guide-Weight method for topology optimization of continuum structures including body forces (Guan et al. 2013)

Sensitivity analysis for optimization design of non-uniform curved grid stiffened composite (NCGC) structures (Wang et al. 2018)

Coniglio, S., Gogu, C., Amargier, R., and Morlier, J., "Pylon and engine mounts performance driven structural topology optimization," WCSMO12 12thWorld Congress of Structural and Multidisciplinary Optimization 5 - 9 June 2017, Braunschweig, Germany, 2017

S.CONIGLIO, C.GOGU, R.AMARGIER, J.MORLIER, "Original pylon architecture design using 3D HPC topology optimization", AIAA SciTech Forum, 8–12 January 2018, Kissimmee, Florida, 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference

Stress-based shape and topology optimization with the level set method (Picelli et al. 2017)

Topology Optimization of Compliant Mechanisms (thesis of G.Pesare 2013)

X. Wang, K. Long, V. Hoang, P. Hu, An explicit optimization model for integrated layout design of planar multi-component systems using moving morphable bars, Comput. Methods Appl. Mech. Engrg. (2018), https://doi.org/10.1016/j.cma.2018.07.032

Topology optimization with supershapes (J.Norato 2018)

A geometry projection method for continuum-based topology optimization with discrete elements (Norato et al. 2015)

Verbart, A., Langelaar, M., & Keulen, F. V. (2017). A unified aggregation and relaxation approach for stress constrained topology optimization. Structural and Multidisciplinary Optimization, 1-17. DOI: 10.1007/s00158-016-1524-0.

A study on X-FEM in continuum structural optimization using a level set model (Wei et al. 2009)

GENERATING OPTIMAL TOPOLOGIES IN STRUCTURAL DESIGN USING A HOMOGENIZATION METHOD (Bendsoe & Kikuchi 1988)

Implementation of topological derivative in the moving morphable components approach (Yoon & Takalloozadeh 2017)

The Moving Node Approach in Topology Optimization: An Exploration to a Flow-inspired Meshless Method-based Topology Optimization Method (thesis of J.Overvelde)