High performance parallel computing in structural topology optimization

I. Colominas, J. París, F. Navarrina, M. Casteleiro

GMNI-Group of Numerical Methods in Engineering
Civil Engineering School. University of Coruña, SPAIN
email → icolominas@udc.es

Abstract

Minimum weight topology optimization of structures with stress constraints formulations have been recently proposed as an alternative to the traditional maximum stiffness statements. These minimum weight approaches offer some improvements and avoid unwanted phenomena associated to maximum stiffness approaches (mesh dependency, checkerboard layouts). In addition, minimum weight formulations address the most useful analysis from a practical point of view in engineering (to reduce the cost of the structure) and it imposes stress constraints. Thus, the feasibility of the optimal solutions obtained is guaranteed while the cost is minimized.

However, these formulations also require higher computing effort than the traditional maximum stiffness statements. Thus, it is necessary to introduce numerical and computational techniques that allow to reduce the computational requirements.

In this paper, we propose the use of parallelization techniques in order to reduce the computing time required to solve the topology optimization problem proposed.

Keywords: Parallel computing, topology optimization, stress constraints, minimum weight, OpenMP.

1 Introduction

Topology optimization of structures is the most recent branch in the structural optimization field. The main goal of this formulations is to define the most appropriate material distribution in a predefined domain that maximizes (or minimizes) a established function (or a property) related to the structural response. Thus, the most usual formulations proposed to solve the topology optimization problem tries to maximize the stiffness of the structure by using a predefined amount of material [1, 2, 3]. This formulation offers important computational advantages since the number of constraints involved is small (only one volume constraint). On the other hand, the rest of the formulations and applications about structural optimization usually state procedures that try to minimize the weight or the cost of the structure. In addition, minimum weight formulations usually impose stress and/or displacement constraints in order to verify the feasibility of the final solution obtained.

These formulations are more realistic from a practical point of view but they also require larger computing resources than the conventional maximum stiffness approaches. Minimum weight with stress constraints formulations usually include a large number of highly non-linear constraints [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. In addition, the number of design variables usually involved in topology optimization problems is also very large (usually more than one thousand). Thus, the whole optimization problem is much larger than the one defined by means of the maximum stiffness approach. The computing effort devoted to the structural analysis increases slightly since stress constraints need to be computed and verified. However, the sensitivity analysis involved requires much higher computational effort since the number of stress constraints is also much larger than one (the volume constraint associated to maximum stiffness approaches). Following with the same idea, the optimization algorithms required to solve the minimum weight problem with stress constraints are much more sophisticated than the ones used to solve the maximum stiffness approach. In addition, the computational effort required by the optimization algorithm increases with the number of constraints and design variables.

According to the previous explanations, the calculation of the optimal solution for the topology optimization problem with stress constraints requires higher effort, specially in computing time terms, than the maximum stiffness approaches. Thus, it is necessary to propose acceleration techniques that allow to reduce the computing time required. From a numerical point of view, this speed-up can be raised by adapting and developing specific algorithms for this specific problem. From a computational point of view, this speed-up can be addressed by using parallelization techniques [10]. In practice, both methodologies are usually connected since the parallelization of the algorithms is usually related to the numerical formulations used.

In this paper, we analyze an entire optimization procedure specifically designed for the topology optimization problem with stress constraints. This algorithm introduces computational advantages and in addition it allows to use parallelization techniques in order to accelerate the calculation of the optimal design.

2 Optimum design methodology

Topology optimization problems with stress constraints usually require very high computing effort to obtain the optimal design. However, this computational effort can be reduced by proposing specific algorithms and procedures. A specific optimization methodology is proposed in this paper in order to reduce the computing time required (see figure 1).

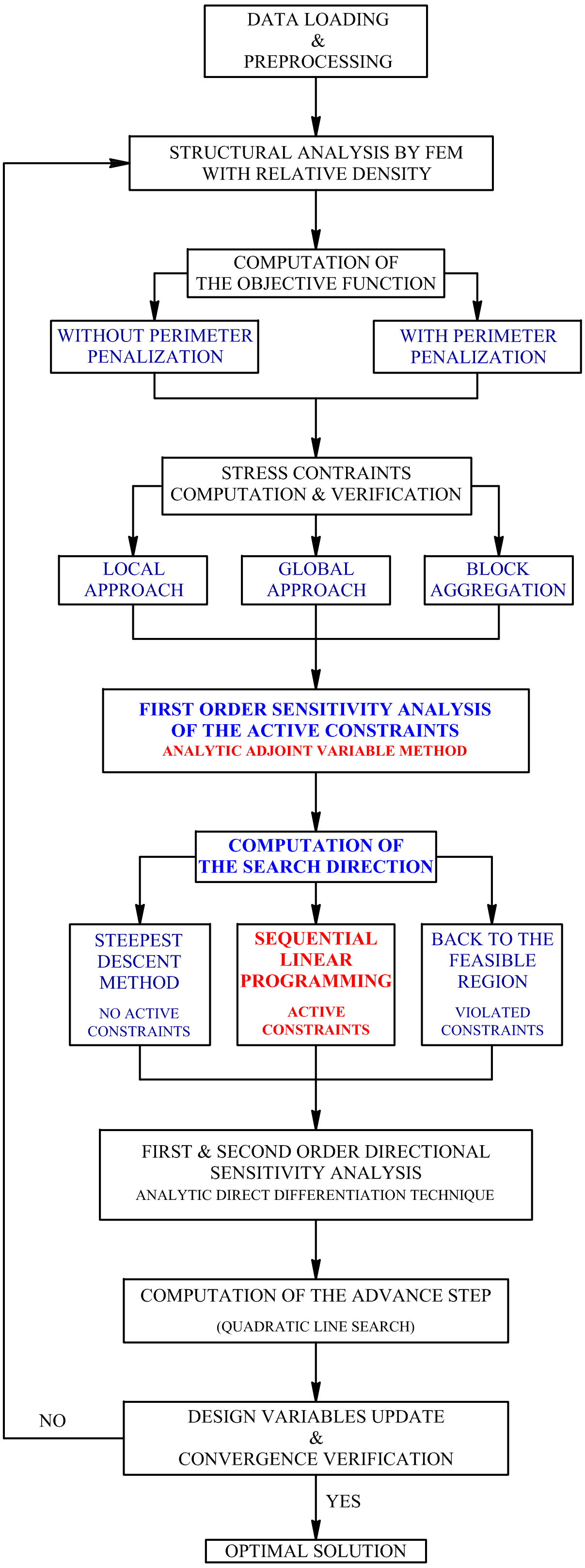

Figure 1: Optimization Methodology.

Note that the optimization procedure is iterative. In addition, several important stages are required to obtain the optimal design. First stage develops the structural analysis by means of the FEM including the relative density. Second stage is devoted to compute the objective function (the weight of the structure). Then, stress constraints are computed and verified. Thus, a set of active constraints (almost violated constraints) is obtained. This set of active constraints defines a set of first order derivatives of constraints required in the next stages of the optimization methodology. Thus, only the first order derivatives of active constraints are computed in order to reduce the number of calculations. The first order sensitivity analysis of the active stress constraints is developed by using an "adjoint variable" algorithm. This sensitivity analysis procedure will be explained in detail further on this paper. First order derivatives of the stress constraints and first order derivatives of the objective function are used to obtain the search direction. Three different algorithms are proposed depending on the degree of violation of the stress constraints. However, the most frequently used algorithm is Sequential Linear Programming (SLP). This SLP algorithm is also analyzed in detail further in this paper since the calculation of the search direction is the most expensive topic in computing time terms. This search direction can be used now to calculate first and second order directional derivatives of the stress constraints and the objective function. These directional derivatives are used to compute second order Taylor expansions of the stress constraints and the objective function. These Taylor expansions allow to perform a Quadratic Line Search (QLS) in the direction previously obtained and to obtain the most suitable advance factor. Finally, the search direction and the advance factor allows to obtain an improved value of the vector of design variables. This updated vector of design variables defines an improved structure. Then, the optimization process may begin again in the first stage with this improved value of the design variables.

The most expensive stages of this optimization procedure are dedicated to solve the structural analysis with relative density, to obtain the first order derivatives of the stress constraints and to obtain the search direction by means of the SLP algorithm. These stages need to be explained in detail in order to propose techniques that allows to reduce the computation effort.

3 Structural analysis

The structural analysis of the topology optimization problem proposed is developed by assuming the linear elasticity hypothesis with small displacements and small displacement derivatives. The domain of the structure is discretized by using a galerkin type finite element model.

This FEM formulation must include the effect of the relative density of the material being used. Thus, a conventional FEM formulation need to be modified in order to include the design variables of the optimization problem. This issue has been usually addressed by using predefined microstructures of material (e.g. hole-in-cell, rank-2 layered, SIMP,...) and by applying homogenization techniques [1, 2, 3, 4]. The authors have proposed a numerical formulation that includes the effect of the relative density [6, 7, 9, 10]. This formulation requires to introduce slight modifications in a conventional finite element formulation. In practice the modifications are reduced to take into account the effect of the relative density in the integrals of the contribution of each finite element of the mesh.

The whole FEM formulation including the effect of the relative density is developed in previous papers of the authors [6, 7].

According to the previous explanations, the calculation of the structural analysis with relative density consists in a conventional finite element analysis with slight modifications. Thus, the calculation in parallel of the structural analysis must only be considered in the calculation of the elemental contributions to the global stiffness matrix. The assembling of the elemental stiffness matrices in the global stiffness matrix and the resolution of the resulting system of linear equations can not be easily parallelized. In addition, adequate performance of the parallelization is not expected. The system of linear equations is solved by means of a Cholesky factorization since it allows to compute additional systems of equations with a small effort. These additional systems of linear equations will be required in order to obtain the sensitivity analysis. In addition, the computing time devoted to the computation of the structural analysis is much smaller than the required to obtain the sensitivity analysis or the required by the optimization algorithm.

4 First order sensitivity analysis of the stress constraints

First order sensitivity analysis of the stress constraints is developed analytically by applying the “adjoint variable” method. In this paper, we are going to analyze the sensitivity analysis of local stress constraints since the computing time is much larger than the computing time required by the global approach or by the block aggregation approach of stress constraints. All these three formulations have been previously studied in [4, 5, 6, 7, 8, 9, 10, 11, 12, 16].

According to [4, 5, 6, 7, 10, 12], the local stress constraints approach imposes one local constraint in the central point of each element of the finite element mesh. This local stress constraint is defined, according to [6, 7, 10], as

$$g_e(\boldsymbol{\rho}) = \tilde{g}_e(\widehat{\sigma}_{\text{VM,e}}, \rho_e)\Bigg|_{\substack{\widehat{\sigma}_{\text{VM},e}(\boldsymbol{\sigma}_e) \\ \boldsymbol{\sigma}_e(\boldsymbol{\alpha}) \\ \boldsymbol{\alpha}(\boldsymbol{\rho})}} \tag{1}$$

$$\tilde{g}_e(\widehat{\sigma}_{\text{VM,e}}, \rho_e) = \Big[\widehat{\sigma}_{\text{VM,e}} - \widehat{\sigma}_{max}\ \varphi_e\Big]\rho_e^q \leq 0. \tag{2}$$

where

$$\varphi_e = 1 - \varepsilon + \frac{\varepsilon}{\rho_e}, \tag{3}$$

$\widehat{\sigma}_{\text{VM,e}}$ is the Von Mises reference stress in the central point of element e, $\widehat{\sigma}_{max}$ is the failure stress of the material and ε is the so called "relaxation factor" and usually takes the values $(\varepsilon \in (0.001,\ 0.1))$ [4, 7, 14, 15, 17]. The exponent q usually take the value $q = 1$ in order to deal with effective stresses [6, 7].

In 2D problems and by assuming the plane stress hypothesis, the Von Mises stress criterion is defined as

$$\widehat{\sigma}_{\text{VM,e}} = \sqrt{\sigma_{x,e}^2 + \sigma_{y,e}^2 - \sigma_{x,e}\sigma_{y,e} + 3\tau_{xy,e}^2} \tag{4}$$

and

$$\boldsymbol{\sigma}_e = \left\{ \begin{array}{c} \sigma_{x,e} \\ \sigma_{y,e} \\ \tau_{xy,e} \end{array} \right\} \tag{5}$$

is the stress tensor in the central point of element e.

First order derivatives of the local stress constraints can be obtained as:

$$\frac{dg_e}{d\rho_i} = \frac{\partial \tilde{g}_e}{\partial \widehat{\sigma}_{\text{VM},e}} \Bigg|_{\substack{\widehat{\sigma}_{\text{VM},e}(\boldsymbol{\sigma}_e) \\ \boldsymbol{\sigma}_e(\boldsymbol{\alpha}) \\ \boldsymbol{\alpha}(\boldsymbol{\rho})}} \frac{d\widehat{\sigma}_{\text{VM},e}}{d\boldsymbol{\sigma}_e} \Bigg|_{\substack{\boldsymbol{\sigma}_e(\boldsymbol{\alpha}) \\ \boldsymbol{\alpha}(\boldsymbol{\rho})}} \frac{d\boldsymbol{\sigma}_e}{d\boldsymbol{\alpha}} \Bigg|_{\boldsymbol{\alpha}(\boldsymbol{\rho})} \frac{d\boldsymbol{\alpha}}{d\rho_i} + \delta_{ie} \frac{\partial \tilde{g}_e}{\partial \rho_i} \Bigg|_{\substack{\widehat{\sigma}_{\text{VM},e}(\boldsymbol{\sigma}_e) \\ \boldsymbol{\sigma}_e(\boldsymbol{\alpha}) \\ \boldsymbol{\alpha}(\boldsymbol{\rho})}}, \tag{6}$$

where

$$\frac{dg_e}{d\boldsymbol{\rho}} = \left\{ \frac{dg_e}{d\rho_i} \right\}_{i=1,\ldots,N_e} \tag{7}$$

and N_e is the number of elements of the mesh.

Now, the sensitivity analysis of the nodal displacements of the structure is required. These derivatives can be obtained analytically by differentiating the system of linear equations obtained when the finite element method is used. In this problem, we propose an isoparametric Galerkin formulation under the small displacement and small displacement gradient hypotheses. In addition, the set of shape functions used satisfy the essential boundary conditions (prescribed displacements) in order to avoid dealing with derivatives of the reactions obtained in the nodes where prescribed displacements were stated. According to that, the structural analysis can be obtained by solving the system of linear equations:

$$\boldsymbol{K}\,\boldsymbol{\alpha} = \boldsymbol{f} \tag{8}$$

where, in this case:

$$\boldsymbol{K} \equiv \boldsymbol{K}(\boldsymbol{\rho}), \quad \boldsymbol{\alpha} \equiv \boldsymbol{\alpha}(\boldsymbol{\rho}), \quad \boldsymbol{f} \equiv \boldsymbol{f}(\boldsymbol{\rho}). \tag{9}$$

The matrix $\boldsymbol{K}$ is the stiffness matrix of the structure, $\boldsymbol{\alpha}$ is the vector of nodal displacements and $\boldsymbol{f}$ is the vector of applied loads. Each one of the terms $\boldsymbol{K}_{ji}$ of the stiffness matrix $\boldsymbol{K}$ can be obtained as

$$\boldsymbol{K}_{ji} = \sum_{e=1}^{N_e} \boldsymbol{K}^e_{ji}, \qquad j = 1, \ldots, N \qquad i = 1, \ldots, N \tag{10}$$

where N_e is the number of elements and N is the number of nodes of the finite element mesh used. The elemental stiffness matrices $\boldsymbol{K}^e$ must be computed as usual by multiplying the corresponding integrand times the relative density of element e (See [6, 7, 8, 10] for more details)

The vector of applied forces $\boldsymbol{f}_j$ can be obtained as usual in a conventional finite element formulation by taking into account that the contribution of the forces per unit of volume must be multiplied times the relative density (see [6, 7, 8, 10] for more details).

As it can be observed in (8), reactions are not involved in this formulation since the trial functions are forced to satisfy the essential boundary conditions [6, 7]. Consequently, first order derivatives of the nodal displacements ($\boldsymbol{\alpha}$) versus the design variables ($\boldsymbol{\rho}$) can be obtained by differentiating (8) as:

$$\boldsymbol{K}\frac{d\boldsymbol{\alpha}}{d\boldsymbol{\rho}} = \frac{d\boldsymbol{f}}{d\boldsymbol{\rho}} - \frac{d\boldsymbol{K}}{d\boldsymbol{\rho}}\boldsymbol{\alpha}. \tag{11}$$

The terms on the right side of (11) can be directly obtained from a conventional approach of the FEM. If we take into account that the external distributed forces per unit of area do not usually depend on the design variables (in the topology optimization problem) the derivatives are null. On the other hand, the forces per unit of volume depends linearly on the relative density of each element [7, 10]. Thus, the derivatives of the nodal forces ($\boldsymbol{f}_k, k = 1, ..., N$) can be obtained by assembling the elemental contributions of the forces per unit of volume $\boldsymbol{f}^e_k$ and by considering that the relative density of element e is $\rho_e = 1$.

On the other hand, the derivatives of the stiffness matrix ($\boldsymbol{K}$) over the design variable ρ_e can be obtained by assembling the elemental stiffness matrices $\boldsymbol{K}^e$ with $\rho_e = 1$. Obviously, the contribution of the rest of the elemental stiffness matrices is null since they do not depend on the relative density ρ_e.

Thus, the derivatives of the nodal displacements over each design variable ρ_i can be replaced in (6) according to (11) as:

$$\frac{dg_e}{d\rho_i} = \boldsymbol{\lambda}_e^T\left(\frac{d\boldsymbol{f}}{d\rho_i} - \frac{d\boldsymbol{K}}{d\rho_i}\boldsymbol{\alpha}\right) + \delta_{ie}\frac{\partial \tilde{g}_e}{\partial \rho_i}\Bigg|_{\substack{\widehat{\sigma}_{\text{VM},e}(\boldsymbol{\sigma}_e)\\ \boldsymbol{\sigma}_e(\boldsymbol{\alpha})\\ \boldsymbol{\alpha}(\boldsymbol{\rho})}} \tag{12}$$

where δ_{ie} is the Kronecker delta function and $\boldsymbol{\lambda}_e$ is the “adjoint variable”, which is defined as:

$$\boldsymbol{\lambda}_e^T = \left.\frac{\partial \tilde{g}_e}{\partial \widehat{\sigma}_{\mathrm{VM},e}}\right|_{\substack{\widehat{\sigma}_{\mathrm{VM},e}(\boldsymbol{\sigma}_e)\\ \boldsymbol{\sigma}_e(\boldsymbol{\alpha})\\ \boldsymbol{\alpha}(\boldsymbol{\rho})}} \left.\frac{d\widehat{\sigma}_{\mathrm{VM},e}}{d\boldsymbol{\sigma}_e}\right|_{\substack{\boldsymbol{\sigma}_e(\boldsymbol{\alpha})\\ \boldsymbol{\alpha}(\boldsymbol{\rho})}} \left.\frac{d\boldsymbol{\sigma}_e}{d\boldsymbol{\alpha}}\right|_{\boldsymbol{\alpha}(\boldsymbol{\rho})} \boldsymbol{K}^{-1}. \tag{13}$$

All the terms in the previous equation can be directly obtained by applying theoretical differentiating rules (see [10] for more details). In addition, the "adjoint variable" can be obtained by solving the system of linear equations proposed in (13).

The 'adjoint variable" method allows to compute the derivatives of each stress constraint over the relative densities directly if the "adjoint variable" is known. Thus, this calculation is developed in two independent steps. First step is devoted to compute the "adjoint variable" $\boldsymbol{\lambda}_e$ by solving the system of linear equations proposed in (13). This computation can be developed independently for each stress constraint. The second step is devoted to compute the sensitivity analysis of the stress constraints according to (12) and by using the value of $\boldsymbol{\lambda}_e$ computed in the first step. This calculation can be also performed independently for each stress constraint since the value of the "adjoint variable" is also computed independently for each stress constraint in the first step. Thus, the entire computation of the first order sensitivity analysis of the stress constraints can be developed independently for each constraint. This fact is very important if parallelization techniques are considered since the calculation of the derivatives of each stress constraint can be completely computed in a different processor without any interference with other derivatives. Thus, first order sensitivity analysis of the stress constraints can be easily parallelized by adequately distributing the iterations of the "DO" loop used to compute the derivatives of all the local stress constraints proposed (loop parallelization). Consequently, the speed-up obtained reaches in practice the expected value.

5 Sequential Linear Programming Algorithm

In the introduction section, the most expensive algorithms of the whole optimization procedure where indicated. In previous sections, the structural analysis and the sensitivity analysis was analyzed. In this section the optimization algorithm used is analyzed in order to propose some improvements that allow to reduce the computing time devoted to obtain the optimal solution.

The optimization algorithm proposed to obtain the optimal topology of the structure is a general procedure that can be applied to other different optimization problems. This algorithm is developed in two steps. First step computes the right search direction and the second step performs a Quadratic Line Search by using the previously obtained search direction [6, 7, 10, 18]. Thus, an improved value of the vector of design variables can be obtained as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \theta_k \, \boldsymbol{s}_k \tag{14}$$

where $\boldsymbol{s}_k$ is the search direction of the design variables in the k-th iteration and θ_k is the advance factor along the search direction [10]. The direction $\boldsymbol{s}_k$ is obtained by means of Sequential Linear Programming [19] and the advance factor θ_k is obtained by using a Quadratic Line Search in order to avoid zig-zag phenomena around the optimal solution [6, 10, 20, 21].

Sequential Linear Programming algorithm is based on the use of mathematical programming techniques (the Simplex algorithm [22]). This technique allows to solve optimization problems with non linear objective function and non linear stress constraints by solving a sequence of linearized optimization problems. Simplex algorithm is based on the properties of the linearized space obtained. According to this idea, the optimal solution of this linearized problem corresponds to one vertex of the polyhedron, in general. Thus, the optimization algorithm consists in

$$\begin{array}{ll} \textit{minimize} & \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \\ \textit{verifying} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{A} : m \times n \\ & l_i \leq x_i \leq u_i, \quad i = 1, \ldots, n \end{array} \tag{15}$$

where l_i and u_i are the lower and upper limits, respectively, of the design variables (x_i). The lower limit is usually stated as $l_i = 0$ in order to simplify the formulation. This Mathematical Programming problem can be presented in a standard form that facilitates the application of the Simplex Algorithm. This standard form proposed by Dantzig [22] is

$$\begin{array}{ccccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 & & \\ a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 & & \\ \vdots & & \vdots & & \ddots & & \vdots & & \vdots & & \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & = & b_m, & & \\ c_1x_1 & + & c_2x_2 & + & \cdots & + & c_nx_n & = & f & \text{(Min)} & \\ & & & & & & & & & & \\ x_1 \geq l_1 & , & x_2 \geq l_2 & , & \cdots & , & x_n \geq l_n, & & \text{and} & & \\ x_1 \leq u_1 & , & x_2 \leq u_2 & , & \cdots & , & x_n \leq u_n. & & & & \end{array} \tag{16}$$

that can be schematically represented by using matrix notation as:

$$\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m, \\ \hline c_1 & c_2 & \cdots & c_n & f \end{array} \tag{17}$$

The Simplex algorithm originally deals with equality constraints but, in practical applications of structural optimization, most of the constraints corresponds to inequality equations. Thus, slack variables need to be considered that transform the original constraints in equality constraints. These slack variables are required to obtain the search direction but they are not used in the following stages of the optimization method. The slack variables are included in the mathematical model in order to facilitate the use of the Simplex algorithm.

The standard form proposed in (17) with the additional slack variables defines a system of linear equations with $n + m$ unknown variables and m linear equations. Thus, this system has not an unique solution and the value of a number of variables must be defined in order to obtain a well defined system of linear equations. Thus, a group of variables with predefined values defines a set of non basic variables ($\boldsymbol{x}_{N}$). The rest of variables whose value has not been predefined forms the set of basic variables ($\boldsymbol{x}_{B}$). According to that, the main objective of the Simplex algorithm is to minimize the value of the objective function by adequately redefining the set of basic and non basic variables and the value of the non basic variables. Thus, the mathematical programming algorithm can be written during all this process as

$$\boldsymbol{I}\boldsymbol{x}_{B} = \widehat{\boldsymbol{b}} - \widehat{\boldsymbol{A}}\boldsymbol{x}_{N}, \tag{18}$$

where $\boldsymbol{I}$ is an identity matrix that multiplies times the set of basic variables ($\boldsymbol{x}_{B}$) and $\widehat{\boldsymbol{A}}$ is the matrix obtained by applying transformations in the rows of the original matrix of the standard form (17) according to the Simplex Algorithm. This matrix multiplies times the set of non-basic variables ($\boldsymbol{x}_{N}$). The non-basic variables will take the lowest or the highest value allowed according to the Simplex algorithm since these ones are the values that produces the highest reduction possible in the objective function. The basic variables take the values stated in the right term of (18).

Some additional aspects need to be addressed in order to entirely explain the Simplex algorithm (initial solution, low and high limits of the variables, matrix operations, ...). All these topics are not addressed in this paper due to extension limits. However, an entire explanation with further details can be found in [10, 22]. The computation effort of the Simplex algorithm is devoted to decide the most suitable modification of the set of basic and non-basic variables and the application of this modification by performing linear combinations of rows like in the Gauss elimination algorithm (see [10, 22] for more details).

Thus, the entire standard form matrix need to be modified for each update of the sets of basic and non-basic variables. This modification requires about 97 % of the total computing time dedicated to obtain the search direction for the cantilever beam problem with 7200 design variables and the local approach of the stress constraints. Thus, the entire Simplex Algorithm can not be parallelized since each modification of the sets of basic and non-basic variables depends on the previous modifications and in the corresponding matrix of the standard form. However, each modification requires to update the entire matrix of the standard form. This update requires to perform linear combinations of all the rows of the matrix and these linear combinations can be developed in parallel since the modification of one row does not influence the modification of the other rows. According to that, each row can be modified in a different processor. Obviously this parallelization does not produce as appropriate performance as the one obtained for the first order sensitivity analysis of the local stress constraints since some specific calculations of the algorithm need to be computed sequentially and a large number of synchronizations between all the processors need to be included.

6 Parallelization techniques

Previous sections explain the most expensive algorithms in computational terms used in the topology optimization of structures with stress constraints. These algorithms (first order sensitivity analysis and SLP) can be computed in parallel by dividing the iterations of the "do" loops in different processors. First order derivatives of each stress constraint can be computed in a different processor and the matrix modifications proposed in the previous section can be performed in parallel since each row of the matrix can be modified in a different processor. This type of parallelization of the iterations of "do" loops is very easy to implement in practice and produce excellent performance.

In this paper we have used OpenMP directives in order to parallelize the algorithms proposed. This parallelization methodology allows to develop a source code sequentially. Then, the computation in parallel can be developed with slight modifications in the original source code by adding a reduced number of additional lines: the OpenMP directives. This fact means a very important advantage since other parallelization strategies like MPI require to develop all the source code in parallel.

Figure 2 shows the speed-up obtained for the topology optimization of a cantilever beam with 7200 design variables by using the local approach of the stress constraints.

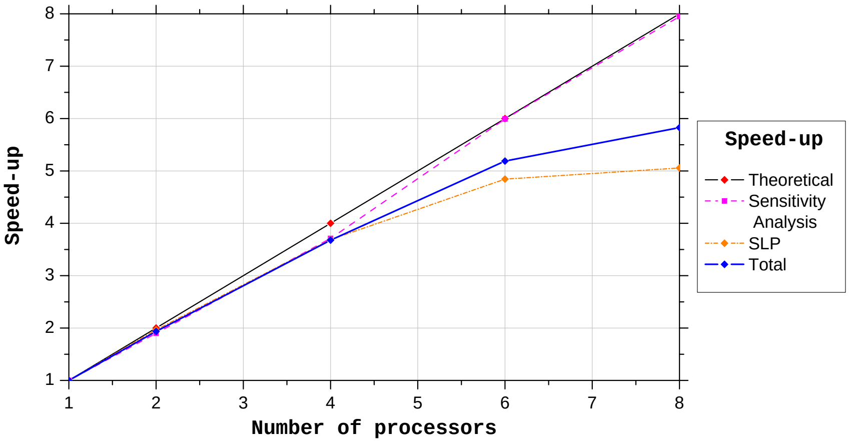

Figure 2: Speed up obtained for the cantilever beam problem with 7200 design variables by using the local approach of the stress constraints in a computing node with 4 Dual Core Intel Xeon 7120 M processors and 16 Gb of RAM.

Figures 3, 4, 5, 6 and 7 show the distribution of the computing time per algorithm (figure 1) at each iteration of the optimization process (for the cantilever beam problem) by using 1, 2, 4, 6 and 8 processors, respectively.

```
----------------------------------------------------------------------------------------------------

 Computing time per iteration

Finite Element Analysis ----------------->    8 s ( 0 h  0 m  8,29 s ) [=                    ]  0.10 %
Stress constraints derivatives ----------> 3044 s ( 0 h 50 m 44,16 s ) [========             ] 38.54 %
Search direction calculation (SLP) ------> 4839 s ( 1 h 20 m 39,37 s ) [=============        ] 61.27 %
Dir. deriv. of the objective function --->    0 s ( 0 h  0 m  0,03 s ) [=                    ]  0.00 %
1st order derivatives of displacements -->    0 s ( 0 h  0 m  0,96 s ) [=                    ]  0.01 %
2nd order derivatives of displacements -->    1 s ( 0 h  0 m  1,57 s ) [=                    ]  0.02 %
1st & 2nd order dir. deriv. of constr. -->    0 s ( 0 h  0 m  0,54 s ) [=                    ]  0.01 %
Quadratic Line Search & update ---------->    2 s ( 0 h  0 m  2,93 s ) [=                    ]  0.04 %

         Total computing time -----------> 7897 s ( 2 h 11 m 37,85 s )

----------------------------------------------------------------------------------------------------
```

Figure 3: Distribution of computing time per algorithm at each iteration for the cantilever beam problem by using 1 processor

```
----------------------------------------------------------------------------------------------------

 Computing time per iteration

Finite Element Analysis ----------------->    7 s ( 0 h  0 m  7,11 s ) [=                    ]  0.17 %
Stress constraints derivatives ----------> 1603 s ( 0 h 26 m 43,27 s ) [========             ] 39.30 %
Search direction calculation (SLP) ------> 2462 s ( 0 h 41 m  2,86 s ) [=============        ] 60.38 %
Dir. deriv. of the objective function --->    0 s ( 0 h  0 m  0,07 s ) [=                    ]  0.00 %
1st order derivatives of displacements -->    1 s ( 0 h  0 m  1,04 s ) [=                    ]  0.03 %
2nd order derivatives of displacements -->    1 s ( 0 h  0 m  1,50 s ) [=                    ]  0.04 %
1st & 2nd order dir. deriv. of constr. -->    0 s ( 0 h  0 m  0,50 s ) [=                    ]  0.01 %
Quadratic Line Search & update ---------->    2 s ( 0 h  0 m  2,85 s ) [=                    ]  0.07 %

         Total computing time -----------> 4079 s ( 1 h  7 m 59,20 s )

----------------------------------------------------------------------------------------------------
```

Figure 4: Distribution of computing time per algorithm at each iteration for the cantilever beam problem by using 2 processors

```
----------------------------------------------------------------------------------------------------

 Computing time per iteration

Finite Element Analysis ----------------->    7 s ( 0 h  0 m  7,07 s ) [=                    ]  0.33 %
Stress constraints derivatives ---------->  820 s ( 0 h 13 m 40,78 s ) [========             ] 38.23 %
Search direction calculation (SLP) ------> 1312 s ( 0 h 21 m 52,51 s ) [=============        ] 61.13 %
Dir. deriv. of the objective function --->    0 s ( 0 h  0 m  0,16 s ) [=                    ]  0.01 %
1st order derivatives of displacements -->    1 s ( 0 h  0 m  1,36 s ) [=                    ]  0.06 %
2nd order derivatives of displacements -->    1 s ( 0 h  0 m  1,55 s ) [=                    ]  0.07 %
1st & 2nd order dir. deriv. of constr. -->    0 s ( 0 h  0 m  0,50 s ) [=                    ]  0.02 %
Quadratic Line Search & update ---------->    3 s ( 0 h  0 m  3,13 s ) [=                    ]  0.15 %

         Total computing time -----------> 2147 s ( 0 h 35 m 47,06 s )

----------------------------------------------------------------------------------------------------
```

Figure 5: Distribution of computing time per algorithm at each iteration for the cantilever beam problem by using 4 processors

7 Application example

In this paper we study the example used to test the performance of the parallelization techniques proposed. Thus, we analyze the optimum design of a cantilever beam with null displacements in the left border and with a vertical force applied in the middle of the right border. Figure 8 shows the dimensions of the domain and the position of the vertical forces applied. In this example, self-weight of the structure has also been included as a structural load.

The domain of the structure has been discretized by using a homogeneous mesh

```
-----------------------------------------------------------------------------------------------------

 Computing time per iteration

Finite Element Analysis ----------------->    7 s ( 0 h  0 m  7,11 s ) [=                   ]  0.47 %
Stress constraints derivatives ---------->  508 s ( 0 h  8 m 28,25 s ) [=======             ] 33.38 %
Search direction calculation (SLP) ------>  999 s ( 0 h 16 m 39,93 s ) [==============      ] 65.70 %
Dir. deriv. of the objective function --->    0 s ( 0 h  0 m  0,24 s ) [=                   ]  0.02 %
1st order derivatives of displacements -->    1 s ( 0 h  0 m  1,52 s ) [=                   ]  0.10 %
2nd order derivatives of displacements -->    1 s ( 0 h  0 m  1,58 s ) [=                   ]  0.10 %
1st & 2nd order dir. deriv. of constr. -->    0 s ( 0 h  0 m  0,50 s ) [=                   ]  0.03 %
Quadratic Line Search & update ---------->    2 s ( 0 h  0 m  2,87 s ) [=                   ]  0.19 %

         Total computing time -----------> 1522 s ( 0 h 25 m 22,00 s )

-----------------------------------------------------------------------------------------------------
```

Figure 6: Distribution of computing time per algorithm at each iteration for the cantilever beam problem by using 6 processors

```
-----------------------------------------------------------------------------------------------------

 Computing time per iteration

Finite Element Analysis ----------------->    7 s ( 0 h  0 m  7,26 s ) [=                   ]  0.54 %
Stress constraints derivatives ---------->  383 s ( 0 h  6 m 23,47 s ) [======              ] 28.28 %
Search direction calculation (SLP) ------>  957 s ( 0 h 15 m 57,96 s ) [===============     ] 70.68 %
Dir. deriv. of the objective function --->    0 s ( 0 h  0 m  0,31 s ) [=                   ]  0.02 %
1st order derivatives of displacements -->    1 s ( 0 h  0 m  1,43 s ) [=                   ]  0.11 %
2nd order derivatives of displacements -->    1 s ( 0 h  0 m  1,54 s ) [=                   ]  0.11 %
1st & 2nd order dir. deriv. of constr. -->    0 s ( 0 h  0 m  0,50 s ) [=                   ]  0.04 %
Quadratic Line Search & update ---------->    2 s ( 0 h  0 m  2,87 s ) [=                   ]  0.21 %

         Total computing time -----------> 1355 s ( 0 h 22 m 35,34 s )

-----------------------------------------------------------------------------------------------------
```

Figure 7: Distribution of computing time per algorithm at each iteration for the cantilever beam problem by using 8 processors

with $120 \times 60 = 7200$ 8-node quadrilateral elements. The thickness of the structure is $0.2\ m$.

The external force applied (2 10^3 kN) has been distributed on four contiguous elements in order to avoid stress accumulation phenomena.

The material being used in this problem is steel with density $\gamma_{mat} = 7650\ \mathrm{kg/m^3}$, Young Modulus $E = 2.1\ 10^5$ MPa, Poisson ratio $\nu = 0.3$ and elastic limit $\widehat{\sigma}_{max} = 230$ MPa.

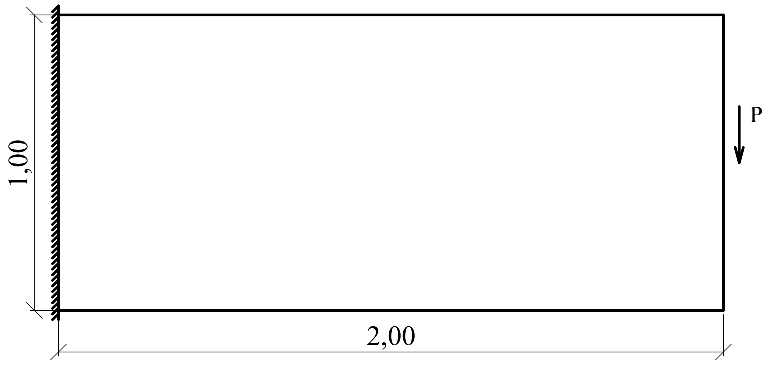

Figure 8: Scheme of the cantilever beam problem (units in m).

Figures 9 and 10 show the optimal solution and the normalized stress state for the cantilever beam problem obtained by using the local approach of stress constraints.

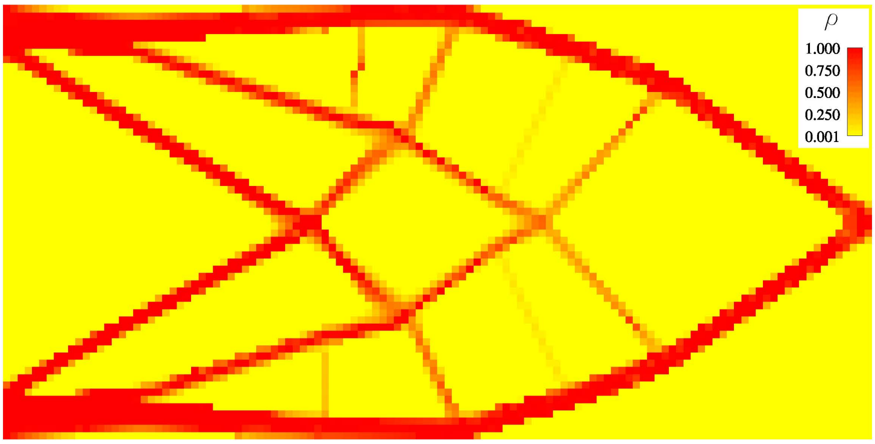

Figure 9: Optimal solution of the cantilever beam problem by using the local approach of stress constraints.

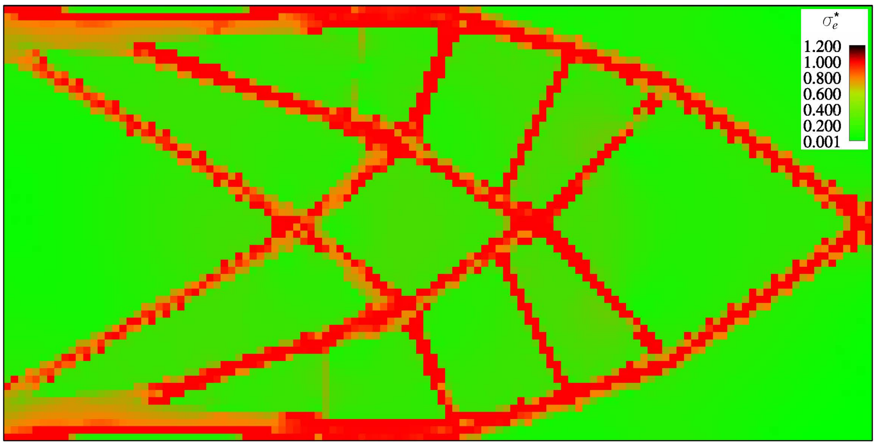

Figure 10: Normalized stress state of the cantilever beam problem by using the local approach of stress constraints.

Table 1 shows the most important parameters of this problem.

Table 1: Summary of the cantilever beam solution.

	q	ε	p	Final Weight (%)
Local approach (Fig. 9)	1	0.01	4	18.27 %

8 Conclusions

This paper introduces some important computational aspects for the topology optimization of structures with local stress constraints. The large number of design variables and stress constraints associated to this formulation of the topology optimization problem requires high computational effort, specially in terms of computing time. Thus, specific algorithms and a whole optimization procedure adapted to this problem need to be addressed. In addition, computational aspects have been taken into account in order to reduce the computing time: parallel computing.

This paper analyses the most expensive algorithms of the whole optimization procedure and studies the feasibility of develop the computations in parallel. Thus, the parallelization of the first order sensitivity analysis of the stress constraints and the parallelization of the computation of the search direction by means of SLP algorithms. The parallelization of the first order derivatives of the stress constraints present an excellent performance since the derivatives of each local stress constraint can be computed in a different processor. In addition, no sequential computations are required. On the other hand, the SLP algorithm can be obtained in parallel but some specific operations of the method need to be computed sequentially. Thus, the performance obtained for the computation in parallel of the SLP algorithm does not reach the theoretical values and, when the number of processors increases, the performance decreases due to the sequential operations required.

However, the parallelization of these two algorithms produces adequate benefits in computing time terms. The speed-up obtained improves even more when the number of design variables and stress constraints increases. Consequently, the use of parallelization techniques reports important benefits and it only requires an acceptable additional programming effort when OpenMP directives are used.

Acknowledgements

This work has been partially supported by Grant Numbers DPI-2006–15275, DPI-2007–61214 of the *"Ministerio de Ciencia y Tecnología"* of the Spanish Government, by Grant Numbers PGIDIT03–PXIC118001PN and PGIDIT03–PXIC118002PN of the *"Dirección Xeral de I+D"* of the *"Consellería de Innovación, Industria e Comercio"* of the *"Xunta de Galicia"*, and by research fellowships of the *"Universidade da Coruña"* and the *"Fundación de la Ingeniería Civil de Galicia"*.

References

[1] M. P. Bendsøe and N. Kikuchi, 'Generating optimal topologies in structural design using a homogenization method", Computer Methods in Applied Mechanics and Engineering, 71, 197-224, 1988.

[2] M. P. Bendsøe, “Optimal shape design as a material distribution problem”, Structural Optimization, 1, 193-202, 1989.

[3] M. P. Bendsøe, “Optimization of structural topology, shape, and material”, Springer-Verlag, Heidelberg, 1995.

[4] P. Duysinx and M.P. Bendsøe, “Topology optimization of continuum structures with local stress constraints” Int. Journal for Numerical Methods in Engineering, 43, 1453-1478, 1998.

[5] P. Duysinx, “Topology optimization with different stress limits in tension and compression”, International report: Robotics and Automation, Institute of Mechanics, University of Liège, Liège, Belgium, 1998.

[6] F. Navarrina et al., “Topology optimization of structures: a minimum weight approach with stress constraints”, Advances in Engineering Software, 36, 599-606, 2004.

[7] J. París et al., “Topoloy optimization of continuum structures with local and global stress constraints”, Structural and Multidisciplinary Optimization, DOI: 10.1007/s00158-008-0336-2, 2008.

[8] J. París et al., “Block Aggregation of Stress Constraints in Topology Optimization of Structures”, Computer Aided Optimum Design of Structures X, 25-34, WIT Press, Southampton, UK, (ISBN: 978-1-84564-070-5), 2008.

[9] J. París et al., “Advances in the statement of stress constraints in structural topology optimization”, Proceedings of the 4th International Conference on Advanced Computational Methods in Engineering ACOMEN 2008, Liège, Belgium, 2008.

[10] J. París, “Restricciones en tensión y minimización del peso: una metodología general para la optimización topológica de estructuras continuas”, PhD Thesis (in Spanish), University of A Coruña, Spain, 2008.

[11] Q. Q. Liang et al., “Optimal selection of topologies for the minimum-weight design of continuum structures with stress constraints”, Journal of Mechanical Engineering Science, Proceedings of The Institution of Mechanical Engineers, Part C, 213(8), 755-762, UK, 1999.

[12] J. T. Pereira et al., “Topology Optimization of Continuum structures with Material Failure Constraints”, Structural and Multidisciplinary Optimization, 26 (1-2), 50-66, 2004.

[13] R. J. Yang and C. J. Chen, “Stress-based topology optimization”, Structural Optimization, 12, 98-105, 1996.

[14] G. Cheng and Z. Jiang, “Study on topology optimization with stress constraints”, Engineering Optimization, 20, 129-148, 1992.

[15] G. D. Cheng and X. Guo, “ε-relaxed approach in structural topology optimization”, Structural Optimization, 13, 258-266, 1997.

[16] J. R. R. A. Martins and N. M. K. Poon, “On Structural Optimization Using Constraint Aggregation”, (WCSMO6) Proceedings VI World Congress on Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil, 2005.

[17] M. Stolpe and K. Svanberg, “On the trajectories of the epsilon-relaxation approach for stress-constrained truss topology optimization”, Structural and Multidisciplinary Optimization, 21 (2), 140-151, 2001.

[18] F. Navarrina and M. Casteleiro, “A general methodologycal analysis for optimum design”, International Journal for Numerical Methods in Engineering, 31, 85-111, 1991.

[19] R. Fletcher, “Practical Methods of Optimization”, Wiley, Scotland, UK, 1987.

[20] F. Navarrina et al., “High order shape design sensitivity: A unified approach”, Computer Methods in Applied Mechanics and Engineering, 188, 681-696, 2000.

[21] F. Navarrina et al., “An efficient MP algorithm for structural shape optimization problems”, Computer Aided Optimum Design of Structures VII, S. Hernández and C.A. Brebbia (Eds.), WIT Press, Southampton, 247-256, 2001.

[22] G. B. Dantzig and M. N. Thapa, “Linear Programming I:Introduction”, Springer-Verlag, New York, 1997.