Mecánica Computacional Vol. XXIV
A. Larreteguy (Editor)
Buenos Aires, Argentina, Noviembre 2005

# DOMAIN DECOMPOSITION METHODS IN STRUCTURAL OPTIMIZATION

**Victorio E. Sonzogni*and Angel Queizán†**

*CIMEC: Centro Internacional de Métodos Computacional en Ingeniería
INTEC, UNL, CONICET
Güemes 3450 (3000), Santa Fe, Argentina
email: sonzogni@intec.unl.edu.ar, web page: http://www.cimec.org.ar

†Facultad Regional La Plata
Universidad Tecnológica Nacional
Calle 60 entre 124 y 122, 1900 La Plata, Argentina
email: aqueizan@frlp.utn.edu.ar

**Key Words:** Domain decomposition methods, Structural optimization, Interior point algorithms.

**Abstract.** *Application of domain decomposition techniques for structural analysis and design are outlined in this paper. This technique divides the whole finite element problem into subdomains in order to solve for each subdomain a smaller problem, and interface unknowns should be explicitly treated. Procedures for structural analysis are described either in the so called primal or dual solution of the interface problem.*

*Structural size optimization problems are also addressed. Element size (i.e. plate thickness) are design variables of the optimization problem, and the structure volume the objective function. The nodal displacements may be introduced as optimization variables in the so called Structural ANalysis and Design (SAND) problems. The number of variables in this case is increased, but there is no need to solve the equilibrium equations, and sensibility matrices are explicit. Interior point algorithms are used to solve the minimization problem.*

## 1 INTRODUCTION

Domain decomposition techniques are efficient tools to solve large equation systems resulting from structural analysis. They proceed by solving separately the interface unknowns and those internals to each subdomain.[2,7,11,13] Iterative methods are followed, and domain decomposition offers a technique which range between direct methods (in the case of a single subdomain) and pure iterative ones (in the limit, if we think in a number of subdomain equal to the number of elements). Domain decomposition leads to efficient preconditioners for the interface problem.

These techniques have been successfully applied to analysis problems. In this paper their use in the frame of optimization problems is discussed.

There are problems where the structural elements size are the design variables (for instance, the area of the cross section of truss elements; or the thickness of plate elements; etc.). In other class of problems not only size, but also shape (node positions) make part of the design variables. Finally some kind of problems aim to find the structural topology which minimizes some objective function. Size optimization is addressed in this paper. The objective function will be the total volume of the structure.

Inequality constraints are present as limitations on the maximum stress and maximum displacements. Bounds on design variables are also stated.

In the so called SAND (Simultaneous ANalysis and Design) optimization, the state variables of the problem (for instance, nodal displacements) are solved together with the design variables (for instance, plate thickness). In this case there is no need to solve the state equations at each iteration, since they are solved together with the design variables in the optimization process. But the size of the problem is greatly increased. Some methods are required to solve such large equation systems.

Interior point algorithms have been proposed to solve such optimization problems with nonlinearities and inequality constraints. The Feasible Directions Interior Point Algorithm (FDIPA)[5] corrects the search direction at each iteration so as to let it to remain within the feasible region. The Feasible Arc Interior Point Algorithm (FAIPA) proposed by Herskovits[4] introduces a nonlinear correction to this aim.

In this paper a general formulation for domain decomposition methods is given, including the so called primal and dual solution of the interface problem. Then a brief presentation of the nonlinear constraint SAND structural problem is given. The ideas of the FDIPA and FAIPA algorithms are presented. The use of domain decomposition methods in optimization is introduced. Finally some examples are given for simple plane stress problems. They include optimization via interior point algorithms, and application of domain decomposition for analysis either in direct way or by minimizing the interface error in displacement compatibility.

## 2 DOMAIN DECOMPOSITION

In this section the nodal variables for the original finite element model and for the subdomains are defined. The description of this technique will be introduced with the aid of a simple example. Let's consider the plane stress problem discretized with a finite element mesh as shown

in figure 1-a, which includes the numeration of nodes and elements. The model degrees-of-freedom (*dof*) are numbered in figure 1-b.


Figure 1: Simple example of a finite element mesh. a) Node and element numbers; b) D.O.F. numbers

A non overlapping decomposition into four subdomains is shown in figure 2. As it can be seen, the interface *dof* are shared by different subdomains.

### 2.1 Global and subdomain variables

Let $\mathbf{u} \in \mathbb{R}^{n0}$ the vector containing the global nodal displacements:

$$\mathbf{u} = \left[u_1 \; u_2 \; \ldots \; u_{30}\right]^T \tag{1}$$

with $n0 = 30$ components.

Denoting $\mathbf{u}^s \in \mathbb{R}^{ns}$ the vector of nodal displacements for subdomain $s$, we can write for subdomain 1 :

$$\mathbf{u^1} = \left[u_1 \; u_2 \; u_3 \; u_4 \; u_7 \; u_8 \; u_9 \; u_{10} \; u_{13} \; u_{14} \; u_{15} \; u_{16} \; \right]^T \tag{2}$$

which has $n1 = 12$ components, and so, for the other subdomains. The subdomain displacements may be reorder:

$$\mathbf{u^{*1}} = \begin{bmatrix} \mathring{\mathbf{u}}^1 \\ \bar{\mathbf{u}}^1 \end{bmatrix} = \left[\left[ \; u_1 \; u_2 \; u_7 \; u_8 \; \right] \left[ \; u_3 \; u_4 \; u_9 \; u_{10} \; u_{13} \; u_{14} \; u_{15} \; u_{16} \; \right] \right]^T \tag{3}$$

being $\mathring{\mathbf{u}}^1$ a vector with *internal* displacements for subdomain 1, and $\bar{\mathbf{u}}^1$ a vector with *interface* displacements for the same subdomain.

Collecting vectors $\mathbf{u}^s$ for all subdomains, one can write:

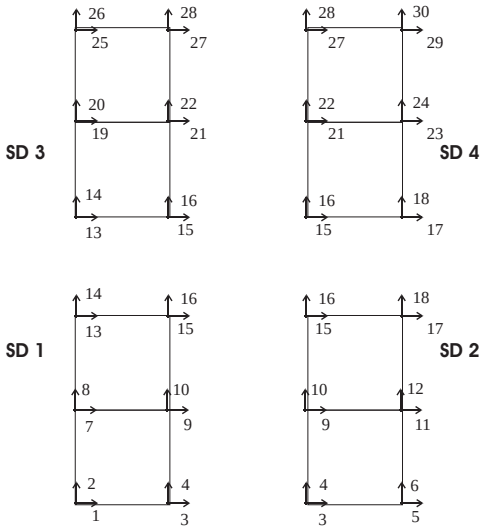


Figure 2: Decomposition of the finite element mesh into 4 subdomains

$$\boldsymbol{u} = \begin{bmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \\ \mathbf{u}^3 \\ \mathbf{u}^4 \end{bmatrix} \tag{4}$$

$\boldsymbol{u} \in \mathbb{R}^{nu}$ contains $nu = 48$ displacements instead of the $n0 = 30$ components of vector $\mathbf{u}$ since there are redundant interface *dof*.

Formally the subdomain displacement vector (for $s = 1$) may be written:

$$\mathbf{u^1} = \mathbf{L}^1 \, \mathbf{u} \tag{5}$$

$\mathbf{L}^1$ is a matrix with $n1 = 12$ rows and $n0 = 30$ columns full of zeros except a single 1 at each row, where the subdomain *dof* (rows) corresponds to the global *dof* (columns).

Therefore the previous formula may be written:

$$\boldsymbol{u} = \mathbf{L} \, \mathbf{u} = \begin{bmatrix} \mathbf{L}^1 \\ \mathbf{L}^2 \\ \mathbf{L}^3 \\ \mathbf{L}^4 \end{bmatrix} \, \mathbf{u} \tag{6}$$

The displacement for the *dof* of the subdomain *interfaces* are:

$$\bar{\mathbf{u}} = \left[ u_3 \; u_4 \; u_9 \; u_{10} \; u_{13} \; u_{14} \; u_{15} \; u_{16} \; u_{17} \; u_{18} \; u_{21} \; u_{22} \; u_{27} \; u_{28} \right]^T \tag{7}$$

a vector for the interface displacement for all subdomains may be written:

$$\bar{\boldsymbol{u}} = \bar{\mathbf{L}}\ \bar{\mathbf{u}} = \begin{bmatrix} \bar{\mathbf{L}}^1 \\ \bar{\mathbf{L}}^2 \\ \bar{\mathbf{L}}^3 \\ \bar{\mathbf{L}}^4 \end{bmatrix}\ \bar{\mathbf{u}} \tag{8}$$

where $\bar{\mathbf{L}}^1$ is a matrix with 8 rows and 14 columns, with zeros anywhere except for a unit value at rows and columns associating the subdomain to the global interface variables.

### 2.2 Multiplicity matrix

Let's define a *multiplicity* matrix $\mathbf{M}$. The simplest form for $\mathbf{M}$ is:

$$\mathbf{M} = diag(M_{ii}) \tag{9}$$

where $M_{ii}$ is the number of subdomains that share the *dof* $i$. Therefore $\mathbf{M}$ is, for this example, a $30 \times 30$ diagonal matrix whose entries are 4 for *dof* 15 and 16; 2 for *dof* along the rest of the interfaces (i.e. 3,4,9,10,13,14,17,18,21,22,27 and 28); and 1 for the rest.

### 2.3 Mean interface displacements

In the iterative process iterated displacements are obtained for each *dof* of each subdomain. At interface nodes, a *mean* value (from those computed at different subdomains) may be computed:

$$\tilde{\mathbf{u}} = (\mathbf{L}\mathbf{M}^{-1})^T \boldsymbol{u} \tag{10}$$

where $\tilde{\mathbf{u}}$ is an approximation to the global $\mathbf{u}$ vector, with the interface displacements taken as the mean value of those for each subdomain.

If we restrict to the interface *dof* this equation may be written:

$$\tilde{\bar{\mathbf{u}}} = (\bar{\mathbf{L}}\bar{\mathbf{M}}^{-\mathbf{1}})^{\mathbf{T}}\ \bar{\boldsymbol{u}} \tag{11}$$

here $\bar{\mathbf{M}}$ corresponds to the same definition as $\mathbf{M}$ but restricted to interface *dof*.

### 2.4 Nodal forces

Nodal forces applied on a *dof* of figure 1-b may be considered as shared by subdomains touching that *dof*. This may be stated:

$$\boldsymbol{f} = (\mathbf{L}\mathbf{M}^{-1})\ \mathbf{f} \tag{12}$$

here $\mathbf{f}$ is a vector with the global nodal forces (size $n0 = 30$, for the example of figure 1), while $\boldsymbol{f}$ is the vector with nodal forces for all subdomains (figure 2) (size $nu = 48$, for the example). It can be seen that a force belonging to vector $\mathbf{f}$ acting on an interface *dof* shared by

two subdomains, results in two entries in vector $\boldsymbol{f}$ each with half the value of the original nodal force.

The same relations, for the interface values only results in:

$$\bar{\boldsymbol{f}} = (\bar{\mathbf{L}}\bar{\mathbf{M}}^{-1})\,\bar{\mathbf{f}} \tag{13}$$

### 2.5 Displacement compatibility

If solution is accomplished by subdomains, compatibility must be enforced at the interface displacements. This may be written:

$$\mathbf{B}\boldsymbol{u} = 0 \tag{14}$$

Matrix $\mathbf{B}$ has a number of columns equal to the total displacement for all subdomains ($nu = 48$ in the example), and a number of rows equal to the compatibility conditions. The latter are one for each *dof* shared by two subdomains and additional equations if there are more than two subdomains sharing interface *dof*. For instance *dof* 15 in figure 2 leads to 3 compatibility conditions: one equating displacements from subdomains 1 and 2; other for subdomains 1 and 3; and other for subdomains 1 and 4. The same could be said for *dof* 16. In the example the number of compatibility equations is 18.

Matrix $\mathbf{B}$ is formed by subdomains contributions. Therefore, in the example:

$$\mathbf{B} = \left[\, \mathbf{B^1}\ \mathbf{B^2}\ \mathbf{B^3}\ \mathbf{B^4} \,\right] \tag{15}$$

being $\mathbf{B^s}$ the $18 \times 12$ matrix for subdomain $s$.

Compatibility conditions restricted to interface *dof* are:

$$\bar{\mathbf{B}}\bar{\boldsymbol{u}} = 0 \tag{16}$$

where $\bar{\mathbf{B}}$ has 18 rows and 32 columns. It has zero everywhere except for two entries at each row: one with value 1 and the other -1. They are placed at columns corresponding to *dof* of vector $\bar{\boldsymbol{u}}$ intervening in that compatibility equation.

It may be verified that:

- $\bar{\mathbf{B}}\bar{\mathbf{L}} = 0$
- $\bar{\mathbf{L}}^T(\bar{\mathbf{L}}\bar{\mathbf{M}}^{-1}) = \mathbf{I}$

## 3 STRUCTURAL ANALYSIS WITH DOMAIN DECOMPOSITION

Applying the finite element method to a linear static structural problem the following equilibrium equations are obtained:

$$\mathbf{K}\,\mathbf{u} \;=\; \mathbf{f} \tag{17}$$

where $\mathbf{u}$ and $\mathbf{f}$ are the displacement and force vectors, and $\mathbf{K}$ is the $n0 \times n0$($30 \times 30$ in the example) stiffness matrix, which is formed by assembling the contribution of the element matrices.

### 3.1 Primal solution for the interface problem: Schur complement

The stiffness matrix for subdomain $s$, reordering variables as made for equation (3), may be put:

$$\mathbf{K}^{*s} = \begin{bmatrix} \mathring{\mathbf{K}}^s & \vec{\mathbf{K}}^s \\ \vec{\mathbf{K}}^{s,T} & \bar{\mathbf{K}}^{\mathbf{s}} \end{bmatrix} \tag{18}$$

where the size of matrix $\bar{\mathbf{K}}^{\mathbf{s}}$ is that of the interface *dof* of subdomain $s$.

Equation (17) after reordering variables is:

$$\mathbf{K}^* \, \mathbf{u}^* \; = \; \mathbf{f}^* \tag{19}$$

with vectors:

$$\mathbf{u} = \begin{bmatrix} \mathring{\mathbf{u}}^1 \\ \mathring{\mathbf{u}}^2 \\ \vdots \\ \mathring{\mathbf{u}}^s \\ \vdots \\ \bar{\mathbf{u}} \end{bmatrix} \quad \text{and} \quad \mathbf{f} = \begin{bmatrix} \mathring{\mathbf{f}}^1 \\ \mathring{\mathbf{f}}^2 \\ \vdots \\ \mathring{\mathbf{f}}^s \\ \vdots \\ \bar{\mathbf{f}} \end{bmatrix} \tag{20}$$

and the stiffness matrix:

$$\mathbf{K} = \begin{bmatrix} \mathring{\mathbf{K}}^1 & 0 & \dots & 0 & \dots & \vec{\mathbf{K}}^1 \bar{\mathbf{L}}^1 \\ 0 & \mathring{\mathbf{K}}^2 & \dots & 0 & \dots & \vec{\mathbf{K}}^2 \bar{\mathbf{L}}^2 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \mathring{\mathbf{K}}^s & \dots & \vec{\mathbf{K}}^s \bar{\mathbf{L}}^s \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \bar{\mathbf{L}}^{1,T} \vec{\mathbf{K}}^{1,T} & \bar{\mathbf{L}}^{2,T} \vec{\mathbf{K}}^{2,T} & \dots & \bar{\mathbf{L}}^{s,T} \vec{\mathbf{K}}^{s,T} & \dots & \bar{\mathbf{K}} \end{bmatrix} \tag{21}$$

being the size of $\bar{\mathbf{u}}$, $\bar{\mathbf{f}}$ and

$$\bar{\mathbf{K}} \; = \; \sum_{s=1}^{NSD} \bar{\mathbf{L}}^{s,T} \bar{\mathbf{K}}^s \bar{\mathbf{L}}^s \tag{22}$$

that of the global interface problem.

The equilibrium equation (19) may be partitioned into the following systems

$$\begin{cases} \mathring{\mathbf{K}}^s \mathring{\mathbf{u}}^s + \vec{\mathbf{K}}^s \bar{\mathbf{L}}^s \bar{\mathbf{u}} &=& \mathring{\mathbf{f}}^s & , s = 1, NSD \\ \sum_{s=1}^{NSD} \bar{\mathbf{L}}^{s,T} \vec{\mathbf{K}}^{s,T} \mathring{\mathbf{u}}^s \; + \; \bar{\mathbf{K}} \bar{\mathbf{u}} &=& \bar{\mathbf{f}} \end{cases} \tag{23}$$

being $NSD$ the number of subdomains.

Eliminating $\mathring{\mathbf{u}}^s$ from the last equation:

$$\begin{cases} \mathring{\mathbf{K}}^s \mathring{\mathbf{u}}^s &=& \mathring{\mathbf{f}}^s - \vec{\mathbf{K}}^s \bar{\mathbf{L}}^s \bar{\mathbf{u}} & , s = 1, NSD \\ \mathbf{K}_I \; \bar{\mathbf{u}} &=& \mathbf{f}_I \end{cases} \tag{24}$$

where:

$$\mathbf{f}_I \; = \; \bar{\mathbf{f}} - \sum_{s=1}^{NSD} \bar{\mathbf{L}}^{s,T} \vec{\mathbf{K}}^{s,T} \left(\mathring{\mathbf{K}^s}\right)^{-1} \mathring{\mathbf{f}}^{s,} \tag{25}$$

and is $\mathbf{K}_I$ is the Schur complement matrix:

$$\mathbf{K}_I \; = \; [\bar{\mathbf{K}} - \sum_{s=1}^{NSD} \bar{\mathbf{L}}^{s,T} \vec{\mathbf{K}}^{s,T} \left(\mathring{\mathbf{K}^s}\right)^{-1} \vec{\mathbf{K}}^s \bar{\mathbf{L}}^{s,}] \tag{26}$$

and taking into account (22):

$$\mathbf{K}_I \; = \; \sum_{s=1}^{NSD} \bar{\mathbf{L}}^{s,T} (\bar{\mathbf{K}}^s - \vec{\mathbf{K}}^{s,T} \left(\mathring{\mathbf{K}^s}\right)^{-1} \vec{\mathbf{K}}^s) \bar{\mathbf{L}}^{s,} \tag{27}$$

Solving the equations system based on (24) may be realized as computing in the first step the global interface displacements ($\bar{\mathbf{u}}$) and then the subdomains internal displacements ($\mathring{\mathbf{u}}^s$). The Schur complement is not explicitly constructed. Instead procedures like the Dirichlet-Neumann are performed.[7,8]

### 3.2 Dual solution for the interface problem

The equilibrium equations may also be written in terms of the displacements for all subdomains $\boldsymbol{u}$ instead of $\mathbf{u}$. Remind that $\boldsymbol{u}$ have redundant displacements for the interface *dof*. The equilibrium systems may be written:

$$\boldsymbol{K} \; \boldsymbol{u} \; = \; \boldsymbol{f} \tag{28}$$

For the example of figure 2 the size of this problem is $nu = 48$. Since there are redundant displacements, compatibility conditions should be enforced:

$$\mathbf{B} \; \boldsymbol{u} \; = \; \mathbf{0} \tag{29}$$

(28) and (29) allows to obtain the solution of problem (17). Constraints (29) may be introduced into (28) via Lagrange multipliers:

$$\boldsymbol{K}\,\boldsymbol{u} \;=\; \boldsymbol{f} \;+\; \mathbf{B}^T\,\mathbf{z} \tag{30}$$

$\mathbf{z}$ contains Lagrange multipliers to enforce compatibility (29) and physically they represent the interaction forces needed to keep equal displacements of different subdomains nodes at common interfaces.

Matrix $\boldsymbol{K}$ is

$$\boldsymbol{K} = \begin{bmatrix} \mathbf{K}^1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \mathbf{K}^2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & \dots & \mathbf{K}^s & \dots & 0 \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & \mathbf{K}^{NSD} \end{bmatrix} \tag{31}$$

and eq. 30 may be written:

$$\mathbf{K}^s\,\mathbf{u}^s \;=\; \mathbf{f}^s \;+\; \mathbf{B}^{sT}\,\mathbf{z} \qquad , \qquad s = 1, NSD \tag{32}$$

If matrix $\mathbf{K}^s$ is not singular (that is, the subdomain is -at least- isostatically supported) one may write:

$$\mathbf{u}^s \;=\; (\mathbf{K}^s)^{-1}\,(\mathbf{f}^s \;+\; \mathbf{B}^{sT}\,\mathbf{z}) \qquad , \qquad s = 1, NSD \tag{33}$$

If however subdomain $s$ is a *floating* one, i.e. their rigid body motions are not prevent, (32) can be solved only if the rhs is orthogonal to the null space of the matrix. This means that:

$$(\mathbf{f}^s \;+\; \mathbf{B}^{sT}\,\mathbf{z})\,\mathbf{R}^s \;=\; 0 \tag{34}$$

where $\mathbf{R}^s$ is a matrix with the rigid body modes of subdomains $s$. It has $n_r$ columns, $n_r$ being the size of the null space: at most 3 for 2D structures or 6 for 3D ones.

Physically the orthogonality condition means that (32) has a non trivial solution if the forces are self-equilibrated. In this case displacements may computed as:

$$\mathbf{u}^s \;=\; \mathbf{K}^{s(+)}\,(\mathbf{f}^s + \mathbf{B}^{s,T}\,\mathbf{z}) \;+\; \mathbf{R}^s\,\boldsymbol{\alpha}^s \qquad , \qquad s = 1, NSD \tag{35}$$

$\mathbf{K}^{s(+)}$ is a generalized inverse of $\mathbf{K}^s$, defined such as:[2]

$$\mathbf{K}^s\,\mathbf{K}^{s(+)}\,\mathbf{K}^s \;=\; \mathbf{K}^s \tag{36}$$

$\boldsymbol{\alpha}^s$ is a vector of amplitudes which gives a linear combination of the rigid modes $\mathbf{R}^s$

Introducing (35) into the second equation of (24) and taking into account (34), one can get:

$$\begin{bmatrix} \mathbf{F}_I & \mathbf{G}_I \\ \mathbf{G}_I^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{c} \end{bmatrix} \tag{37}$$

where

$$\mathbf{F}_I = \sum_{s=1}^{NSD} \mathbf{B}^s \mathbf{K}^{*s} \mathbf{B}^{s,T}$$

$$\mathbf{G}_I = \begin{bmatrix} \mathbf{R}_I^1 & \mathbf{R}_I^2 & \dots & \mathbf{R}_I^{N_f} \end{bmatrix}$$

$$\mathbf{R}_I^s = \mathbf{B}^s \mathbf{R}^s \qquad (s = 1, N_f)$$

$$\mathbf{b} = \sum_{s=1}^{NSD} \mathbf{B}^s \mathbf{K}^{s\ inv} \mathbf{f}^s$$

$$\mathbf{c}^s = \mathbf{R}^{s,T} \mathbf{f}^s \qquad (s = 1, N_f)$$

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}^1 \\ \mathbf{c}^2 \\ \vdots \\ \mathbf{c}^{N_f} \end{bmatrix}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}^1 \\ \boldsymbol{\alpha}^2 \\ \vdots \\ \boldsymbol{\alpha}^{N_f} \end{bmatrix}$$

$$\mathbf{K}^{s\ inv} = \begin{cases} (\mathbf{K}^s)^{-1} & \text{if subdomain s is not floating} \\ \mathbf{K}^{s(+)} & \text{if subdomain s is floating} \end{cases}$$

and $N_f$ the number of floating subdomains.

Equation (37) allows to compute the Lagrange multipliers and the amplitude of the rigid modes for each subdomain, while equation (35) allows to compute the subdomain displacements.

This is the dual Schur complement method or FETI, proposed by Farhat and Roux.[2]

### 3.3 Solving the analysis problem by minimization of the error in compatibility

The problem stated for all subdomains (28) required conditions (29) due to redundant degrees of freedom at interfaces.

Instead of introducing (29) an iterative solution may be performed minimizing the error in compatibility equations. This problem may be stated as an optimization problem:

Problem 1:

*find* $\boldsymbol{u}$ *and* $\mathbf{z}$ *such that:*

$$\min f(\boldsymbol{u}) \quad \forall \boldsymbol{u} \in \mathbb{R}^{nu}$$

*subject to:*

$$\mathbf{e}^s(\boldsymbol{u}, \mathbf{z}) = 0 \qquad s = 1, \ldots NSD$$

where $\boldsymbol{u} \in \mathbb{R}^{nu}$ is the vector with nodal displacements; $f$ is the objective function which, in this case, is a norm of the error in compatibility conditions:

$$f(\boldsymbol{u}) \; = \; \|\mathbf{B}\boldsymbol{u}\|$$

and

$$\mathbf{e}^s \; = \; \mathbf{K}^s \, \mathbf{u}^s \; - \; \mathbf{f}^s \; - \; \mathbf{B}^{sT}\mathbf{z} = 0 \qquad , \qquad s = 1, NSD \tag{38}$$

represent the equilibrium equations for subdomain $s$, and $\mathbf{z} \in \mathbb{R}\mathbf{nz}$ are the interaction forces between subdomains at interface *dof*.

The minimization problem is solved as described in the next section, for optimization problems.

## 4 STRUCTURAL OPTIMIZATION

Structural optimization problems may be expressed as:

Problem 2:

*find* $\mathbf{x}$ *such that:*

$$\min f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^n$$

*subject to:*

$$\mathbf{g}(x) \leq 0$$
$$\mathbf{h}(\mathbf{x}) = 0$$

where $\mathbf{x} \in \mathbb{R}^{nx}$ is the vector of design variables; $f$ is the objective function; $\mathbf{g} \in \mathbb{R}^{ng}$ is a set of inequality constraints; and $\mathbf{h} \in \mathbb{R}^{nh}$ a set of equality constraints.

In the so called *one shot optimization* or *simultaneous analysis and design*, the state variable $u$ are included among the problems unknowns and the state equations are included among constraints:

Problem 3:

*find* $\mathbf{x}$ *and* $\mathbf{u}$ *such that:*

$$\min f(\mathbf{x}, \mathbf{u}) \quad \forall \mathbf{x} \in \mathbb{R}^{nx} \; and \; \forall \mathbf{u} \in \mathbb{R}^{n0}$$

*subject to:*

$$\mathbf{g}(\mathbf{x}, \mathbf{u}) \leq 0$$

$$\mathbf{h}(\mathbf{x},\mathbf{u}) = 0$$
$$\mathbf{e}(\mathbf{x},\mathbf{u}) = 0$$

where $\mathbf{u} \in \mathbb{R}^{n0}$ is the vector of state variables and $\mathbf{e} \in \mathbb{R}^{ne}$ a set of equality constraints representing the state equations.

First order optimality conditions, with Karush-Kuhn-Tucker conditions reads:

$$\begin{cases} \nabla_{\bar{x}} f(\bar{\mathbf{x}}) + \nabla_{\bar{x}} \mathbf{g}(\bar{\mathbf{x}})\lambda + \nabla_{\bar{x}} h(\bar{\mathbf{x}})\mu + \nabla_{\bar{x}} e(\bar{\mathbf{x}})\nu = 0 \\ \mathbf{G}(\bar{\mathbf{x}})\lambda = 0 \\ \mathbf{h}(\bar{\mathbf{x}}) = 0 \\ \mathbf{e}(\bar{\mathbf{x}}) = 0 \\ \lambda \geq \mathbf{0} \\ \mathbf{g}(\bar{\mathbf{x}}) \leq 0 \end{cases} \tag{39}$$

where $\bar{\mathbf{x}}^T = [\mathbf{x}^T \mathbf{u}^T]$ is the vector of unknowns; $\lambda \in \mathbb{R}^{ng}$, $\mu \in \mathbb{R}^{nh}$, $\nu \in \mathbb{R}^{ne}$ are Lagrange multipliers; $\mathbf{G}(\bar{\mathbf{x}}) = diag(\mathbf{g}(\bar{\mathbf{x}}))$; and $\nabla_{\bar{x}}$ denotes derivation with respect to $\bar{\mathbf{x}}$.

The equations in system (39) may be put as:

$$\phi(\mathbf{y}) = \mathbf{0} \tag{40}$$

with $\mathbf{y}^T = [\mathbf{x}^T \mathbf{u}^T \lambda^T \mu^T \nu^T]$. Newton iteration to solve (40) leads to:

$$\nabla_y \phi(\mathbf{y})\ \mathbf{d}^k\ =\ -\phi(\mathbf{y}) \tag{41}$$

being

$$\mathbf{d^k} = \begin{bmatrix} \mathbf{x}^{k+1} - \mathbf{x}^k \\ \mathbf{u}^{k+1} - \mathbf{u}^k \\ \lambda^{\mathbf{k+1}} - \lambda^{\mathbf{k}} \\ \mu^{\mathbf{k+1}} - \mu^{\mathbf{k}} \\ \nu^{\mathbf{k+1}} - \nu^{\mathbf{k}} \end{bmatrix} \tag{42}$$

superscript $k$ refers to iteration $k$. Equation (41) may be written:

$$\begin{bmatrix} \mathbf{S}_{xx} & \mathbf{S}_{xu} & \nabla_x \mathbf{g} & \nabla_x \mathbf{h} & \nabla_x \mathbf{e} \\ \mathbf{S}_{ux} & \mathbf{S}_{uu} & \nabla_u \mathbf{g} & \nabla_u \mathbf{h} & \nabla_u \mathbf{e} \\ \Lambda \nabla_x \mathbf{g}^T & \Lambda \nabla_u \mathbf{g}^T & \mathbf{G} & 0 & 0 \\ \nabla_x \mathbf{h}^T & \nabla_u \mathbf{h}^T & 0 & 0 & 0 \\ \nabla_x \mathbf{e}^T & \nabla_u \mathbf{e}^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}_x^k \\ \mathbf{d}_u^k \\ \lambda^{k+1} \\ \mu^{k+1} \\ \nu^{k+1} \end{bmatrix} = - \begin{bmatrix} \nabla_x f \\ \nabla_u f \\ 0 \\ \mathbf{h} \\ \mathbf{e} \end{bmatrix} \tag{43}$$

Matrix

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{xx} \mathbf{S}_{xu} \\ \mathbf{S}_{ux} \mathbf{S}_{uu} \end{bmatrix}$$

is the hessian of the lagrangian function

$$\mathbf{S} = \nabla^2 f + \lambda^T \nabla^2 \mathbf{g} + \mu^T \nabla^2 \mathbf{h} + \nu^T \nabla^2 \mathbf{e}$$

and $\Lambda$ is a diagonal matrix with $\Lambda_{ii} = \lambda_i$.

The same equation may be used by replacing $\mathbf{S}$ by the quasi-Newton matrix; or by the identity matrix, in a first order iteration scheme.

Eliminating $\mathbf{d}_u^k$ and $\nu^{k+1}$ the system may be expressed as:

$$\begin{bmatrix} \mathbf{S}_{xx} - \mathbf{S}_{xu}\Delta\mathbf{u} - \Delta u^T S_{ux} + \Delta u^T \mathbf{S}_{uu}\Delta\mathbf{u} & \nabla_x \mathbf{g} - \Delta\mathbf{u}^T \nabla_u \mathbf{g} & \nabla_x \mathbf{h} - \Delta\mathbf{u}^T \nabla_u \mathbf{h} \\ \Lambda(\nabla_x \mathbf{g}^T - \nabla_u \mathbf{g}^T \Delta\mathbf{u}) & \mathbf{G} & 0 \\ (\nabla_x \mathbf{h}^T - \nabla_u \mathbf{h}^T \Delta\mathbf{u}) & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}_x^k \\ \lambda^{k+1} \\ \mu^{k+1} \end{bmatrix}$$

$$= \begin{bmatrix} -\nabla_x f - \mathbf{S}_{xu}\delta\mathbf{u} + \Delta u^T(\nabla_u f + \mathbf{S}_{uu}\delta\mathbf{u}) \\ -\Lambda \nabla_u \mathbf{g}^T \delta\mathbf{u} \\ -\mathbf{h} - \nabla_u h^T \delta\mathbf{u} \end{bmatrix} \quad (44)$$

where

$$\delta\mathbf{u} \;=\; [\nabla_u e^T]^{-1}\mathbf{e} \quad (45)$$

$$\Delta\mathbf{u} \;=\; [\nabla_u \mathbf{e}^T]^{-1}\nabla_x \mathbf{e}^T \quad (46)$$

### 4.1 Interior Point Algorithms

Interior point algorithms are based on starting with a feasible variable vector, satisfying all constraints, and moving along descent directions always within the feasible region. A linear correction to the search direction is introduced:[5]

$$\mathbf{d} \;=\; \mathbf{d}_0 \;+\; \rho\, \mathbf{d}_1 \quad (47)$$

$\mathbf{d}_0$ results from solving (43), and $\mathbf{d}_1$ results from

$$\begin{bmatrix} \mathbf{S}_{xx} & \mathbf{S}_{xu} & \nabla_x \mathbf{g} & \nabla_x \mathbf{h} & \nabla_x \mathbf{e} \\ \mathbf{S}_{ux} & \mathbf{S}_{uu} & \nabla_u \mathbf{g} & \nabla_u \mathbf{h} & \nabla_u \mathbf{e} \\ \Lambda\nabla_x \mathbf{g}^T & \Lambda\nabla_u \mathbf{g}^T & \mathbf{G} & 0 & 0 \\ \nabla_x \mathbf{h}^T & \nabla_u \mathbf{h}^T & 0 & 0 & 0 \\ \nabla_x \mathbf{e}^T & \nabla_u \mathbf{e}^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}_x^k \\ \mathbf{d}_u^k \\ \lambda^{k+1} \\ \mu^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\lambda \\ 0 \\ 0 \end{bmatrix} \quad (48)$$

being $\rho$ a measure associated to $\|\mathbf{d}_0\|$ (see Herskovits and Santos[4]).

A further improvement to the search direction is introduced in the FAIPA algorithm,[3] where a line search is made in the direction

$$\mathbf{x}^{k+1} \;=\; \mathbf{x}^{k} \;+\; t\,\mathbf{d} \;+\; t^2\,\tilde{\mathbf{d}} \tag{49}$$

given $\mathbf{d}$ by (47) and $\tilde{\mathbf{d}}$ resulting from equation:

$$\begin{bmatrix} \mathbf{S}_{xx} & \mathbf{S}_{xu} & \nabla_x\mathbf{g} & \nabla_x\mathbf{h} & \nabla_x\mathbf{e} \\ \mathbf{S}_{ux} & \mathbf{S}_{uu} & \nabla_u\mathbf{g} & \nabla_u\mathbf{h} & \nabla_u\mathbf{e} \\ \Lambda\nabla_x\mathbf{g}^T & \Lambda\nabla_u\mathbf{g}^T & \mathbf{G} & 0 & 0 \\ \nabla_x\mathbf{h}^T & \nabla_u\mathbf{h}^T & 0 & 0 & 0 \\ \nabla_x\mathbf{e}^T & \nabla_u\mathbf{e}^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{d}}_x^k \\ \tilde{\mathbf{d}}_u^k \\ \lambda^{k+1} \\ \mu^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\Lambda\,\tilde{\omega}^g \\ -\tilde{\omega}^h \\ -\tilde{\omega}^e \end{bmatrix} \tag{50}$$

$\tilde{\omega}^g \in \mathbb{R}^m$, $\tilde{\omega}^h \in \mathbb{R}^p$, and $\tilde{\omega}^e \in \mathbb{R}^q$ are given by

$$\tilde{\omega}^g \;=\; \mathbf{g}(\mathbf{x}+\mathbf{d}) \;-\; \mathbf{g}(\mathbf{x}) \;-\; \nabla\mathbf{g}^T(\mathbf{x})\,\mathbf{d} \tag{51}$$

$$\tilde{\omega}^h \;=\; \mathbf{h}(\mathbf{x}+\mathbf{d}) \;-\; \mathbf{h}(\mathbf{x}) \;-\; \nabla\mathbf{h}^T(\mathbf{x})\,\mathbf{d} \tag{52}$$

$$\tilde{\omega}^e \;=\; \mathbf{e}(\mathbf{x}+\mathbf{d}) \;-\; \mathbf{e}(\mathbf{x}) \;-\; \nabla\mathbf{e}^T(\mathbf{x})\,\mathbf{d} \tag{53}$$

### 4.2 Multidisciplinary Optimization

Optimization including two different fields may be stated as:

Problem 4:

*find* $\mathbf{x}$, $\mathbf{u}$ *and* $\mathbf{z}$ *such that:*

$$\min f(\mathbf{x},\mathbf{z},\mathbf{u}) \quad \forall\mathbf{x}\in\mathbb{R}^{nx}\;\forall\mathbf{u}\in\mathbb{R}^{n0}\;and\;\forall\mathbf{z}\in\mathbb{R}^{nz}$$

*subject to:*

$$\mathbf{g}(\mathbf{x},\mathbf{z},\mathbf{u}) \le 0$$
$$\mathbf{h}(\mathbf{x},\mathbf{z},\mathbf{u}) = 0$$
$$\mathbf{e^1}(\mathbf{x},\mathbf{z},\mathbf{u}) = 0$$
$$\mathbf{e^2}(\mathbf{x},\mathbf{z},\mathbf{u}) = 0$$

being $\mathbf{z}\in\mathbb{R}^{nz}$ a set of variables representing the interaction between both fields; and $\mathbf{e}^1$,$\mathbf{e}^2$ the state equations of each filed.

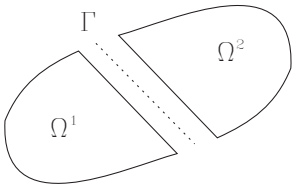


Figure 3: Problem divided into two subdomains

## 5 DOMAIN DECOMPOSITION IN STRUCTURAL AND MULTIDISCIPLINARY OPTIMIZATION

We focus on a linear elastic 2D, plane stress, optimization problem. Let the whole structure be divided into two subdomains (figure 3).

The optimization problem may be stated as in Problem 4, where both, the design (element thickness) and the state variables (nodal displacements), are split into those belonging to each subdomain:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \end{bmatrix} \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}^1 \\ \mathbf{u}^2 \end{bmatrix}$$

being $\mathbf{x}^1$ and $\mathbf{u}^1$ the design and state variables for subdomain 1 and $\mathbf{x}^2$ and $\mathbf{u}^2$ those for subdomain 2.

Including the Karush-Kuhn-Tucker conditions, as in Problem 3, the following system, similar to (43) is got:

$$\begin{bmatrix} \mathbf{S}_{x_1x_1} & \mathbf{S}_{x_1x_2} & \mathbf{S}_{x_1u_1} & \mathbf{S}_{x_1u_2} & \nabla_{x_1}\mathbf{g}^1 & \nabla_{x_1}\mathbf{g}^2 & \nabla_{x_1}\mathbf{h} & \nabla_{x_1}\mathbf{e}^1 & \nabla_{x_1}\mathbf{e}^2 \\ \mathbf{S}_{x_2x_1} & \mathbf{S}_{x_2x_2} & \mathbf{S}_{x_2u_1} & \mathbf{S}_{x_2u_2} & \nabla_{x_2}\mathbf{g}^1 & \nabla_{x_2}\mathbf{g}^2 & \nabla_{x_2}\mathbf{h} & \nabla_{x_2}\mathbf{e}^1 & \nabla_{x_2}\mathbf{e}^2 \\ \mathbf{S}_{u_1x_1} & \mathbf{S}_{u_1x_2} & \mathbf{S}_{u_1u_1} & \mathbf{S}_{u_1u_2} & \nabla_{u_1}\mathbf{g}^1 & \nabla_{u_1}\mathbf{g}^2 & \nabla_{u_1}\mathbf{h} & \nabla_{u_1}\mathbf{e}^1 & \nabla_{u_1}\mathbf{e}^2 \\ \mathbf{S}_{u_2x_1} & \mathbf{S}_{u_2x_2} & \mathbf{S}_{u_2u_1} & \mathbf{S}_{u_2u_2} & \nabla_{u_2}\mathbf{g}^1 & \nabla_{u_2}\mathbf{g}^2 & \nabla_{u_2}\mathbf{h} & \nabla_{u_2}\mathbf{e}^1 & \nabla_{u_2}\mathbf{e}^2 \\ \Lambda\nabla_{x_1}\mathbf{g}^{1T} & \Lambda\nabla_{x_2}\mathbf{g}^{1T} & \Lambda\nabla_{u_1}\mathbf{g}^{1T} & \Lambda\nabla_{u_2}\mathbf{g}^{1T} & G^1 & 0 & 0 & 0 & 0 \\ \Lambda\nabla_{x_1}\mathbf{g}^{2T} & \Lambda\nabla_{x_2}\mathbf{g}^{2T} & \Lambda\nabla_{u_1}\mathbf{g}^{2T} & \Lambda\nabla_{u_2}\mathbf{g}^{2T} & 0 & G^2 & 0 & 0 & 0 \\ \nabla_{x_1}\mathbf{h}^T & \nabla_{x_2}\mathbf{h}^T & \nabla_{u_1}\mathbf{h}^T & \nabla_{u_2}\mathbf{h}^T & 0 & 0 & 0 & 0 & 0 \\ \nabla_{x_1}\mathbf{e}^{1T} & \nabla_{x_2}\mathbf{e}^{1T} & \nabla_{u_1}\mathbf{e}^{1T} & \nabla_{u_2}\mathbf{e}_1{}^T & 0 & 0 & 0 & 0 & 0 \\ \nabla_{x_1}\mathbf{e}^{2T} & \nabla_{x_2}\mathbf{e}^{2T} & \nabla_{u_1}\mathbf{e}^{2T} & \nabla_{u_2}\mathbf{e}_2{}^T & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}^k_{x_1} \\ \mathbf{d}^k_{x_2} \\ \mathbf{d}^k_{u_1} \\ \mathbf{d}^k_{u_2} \\ \lambda_1^{k+1} \\ \lambda_2^{k+1} \\ \mu^{k+1} \\ \nu_1^{k+1} \\ \nu_2^{k+1} \end{bmatrix} = - \begin{bmatrix} \nabla_{x_1} f \\ \nabla_{x_2} f \\ \nabla_{u_1} f \\ \nabla_{u_2} f \\ 0 \\ 0 \\ \mathbf{h} \\ \mathbf{e}^1 \\ \mathbf{e}^2 \end{bmatrix} \tag{54}$$

$\mathbf{g}^s$ are the inequality constraints associated to subdomain $s$, in this case we consider bounds on elements stress:

$$\sigma_j - \sigma_{adm} \leq 0$$

for element $j$. The Lagrange multipliers associated to this constraints are $\lambda_s$.

$\mathbf{h}$ are equality constraints for the displacement compatibility at interface $\Gamma$.

$\mathbf{e}^s$ are the equality constraints for equilibrium of nodal forces at subdomain $s$ and $\nu_s$ their Lagrange multipliers.

The terms of the hessian matrix are of the form

$$\mathbf{S}_{\alpha\beta} = \nabla^2_{\alpha\beta}\, f + \lambda_1^T \nabla^2_{\alpha\beta}\, \mathbf{g}^1 + \lambda_2^T \nabla^2_{\alpha\beta}\, \mathbf{g}^2 + \mu^T \nabla^2_{\alpha\beta}\, \mathbf{h} + \nu_1^T \nabla^2_{\alpha\beta}\, \mathbf{e}^1 + \nu_2^T \nabla^2_{\alpha\beta}\, \mathbf{e}^2$$

In this case, the gradients of the objective function contains the element area while the hessian is null.

Derivatives with respect to $\mathbf{x}$ are null for constraints $\mathbf{g}$ (stress) and $\mathbf{h}$ (compatibility) in this SAND formulation. Derivatives with respect to $\mathbf{u}$ are non zero for both constraints. But the second derivatives (for the hessian) are also null.

For the equilibrium equation we can see that derivatives of $\mathbf{e}^1$ with respect to $\mathbf{x}^1$ are not zero, but those with respect to $\mathbf{x}^2$ are zero. Similarly for derivatives with respect to $\mathbf{u}$. Finally second derivatives with respect to the same variable are zero, but $\nabla_{x_1 u_1}\, \mathbf{e}^1$ are not zero, and the same for subdomain 2.

In this way the global system reads:

$$\begin{bmatrix}
0 & 0 & \mathbf{S}_{x_1u_1} & 0 & 0 & 0 & 0 & \nabla_{x_1}\mathbf{e}^1 & 0 \\
0 & 0 & 0 & \mathbf{S}_{x_2u_2} & 0 & 0 & 0 & 0 & \nabla_{x_2}\mathbf{e}^2 \\
\mathbf{S}_{u_1x_1} & 0 & 0 & 0 & \nabla_{u_1}\mathbf{g}^1 & 0 & \nabla_{u_1}\mathbf{h} & \nabla_{u_1}\mathbf{e}^1 & 0 \\
0 & \mathbf{S}_{u_2x_2} & 0 & 0 & 0 & \nabla_{u_2}\mathbf{g}^2 & \nabla_{u_2}\mathbf{h} & 0 & \nabla_{u_2}\mathbf{e}^2 \\
0 & 0 & \Lambda\nabla_{u_1}\mathbf{g}_1^T & 0 & \mathbf{G}^1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \Lambda\nabla_{u_2}\mathbf{g}_2^T & 0 & \mathbf{G}^2 & 0 & 0 & 0 \\
0 & 0 & \nabla_{u_1}\mathbf{h}^T & \nabla_{u_2}\mathbf{h}^T & 0 & 0 & 0 & 0 & 0 \\
\nabla_{x_1}\mathbf{e}^{1T} & 0 & \nabla_{u_1}\mathbf{e}^{1T} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \nabla_{x_2}\mathbf{e}^{2T} & 0 & \nabla_{u_2}\mathbf{e}^{2T} & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{d}^k_{x_1} \\ \mathbf{d}^k_{x_2} \\ \mathbf{d}^k_{u_1} \\ \mathbf{d}^k_{u_2} \\ \lambda_1^{k+1} \\ \lambda_2^{k+1} \\ \mu^{k+1} \\ \nu_1^{k+1} \\ \nu_2^{k+1}
\end{bmatrix}
= -
\begin{bmatrix}
\nabla_{x_1} f \\ \nabla_{x_2} f \\ 0 \\ 0 \\ 0 \\ 0 \\ \mathbf{h} \\ \mathbf{e}^1 \\ \mathbf{e}^2
\end{bmatrix}
\tag{55}$$

where it has also taken into account that $\nabla_{u_s} f = 0$ for subdomain $s$ in this particular problem.

We can therefore perform the solution of the global system by solving at each subdomain 1 the following:

$$\begin{bmatrix}
0 & \mathbf{S}_{x_1u_1} & 0 & 0 & \nabla_{x_1}\mathbf{e}^1 \\
\mathbf{S}_{u_1x_1} & 0 & \nabla_{u_1}\mathbf{g}^1 & \nabla_{u_1}\mathbf{h} & \nabla_{u_1}\mathbf{e}^1 \\
0 & \Lambda\nabla_{u_1}\mathbf{g}^{1T} & \mathbf{G}^1 & 0 & 0 \\
0 & \nabla_{u_1}\mathbf{h}^T & 0 & 0 & 0 \\
\nabla_{x_1}\mathbf{e}^{1T} & \nabla_{u_1}\mathbf{e}^{1T} & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\mathbf{d}^k_{x_1} \\ \mathbf{d}^k_{u_1} \\ \lambda_1^{k+1} \\ \mu^{k+1} \\ \nu_1^{k+1}
\end{bmatrix}
= -
\begin{bmatrix}
\nabla_{x_1} f \\ 0 \\ 0 \\ \mathbf{h} + \nabla_{u_2}\, h^T \mathbf{d}^k_{u_2} \\ \mathbf{e}^1
\end{bmatrix}
\tag{56}$$

and similarly for subdomain 2. One can see that $\mathbf{d}_{u_2}$ and $\nu_2^{k+1}$ from subdomain 2 are needed to solve the above system for subdomain 1, and viceversa. The interface forces $\mathbf{z}$ are also required to compute the interface residual forces:

$$\mathbf{e}^1 = \mathbf{K}^1 \mathbf{u}^1 - \mathbf{f}^1 - \mathbf{B}^1 \mathbf{z}$$

On the other hand the Lagrange multiplier $\mu$ is computed at both subdomains. An iterative scheme is proposed.

## 6 EXAMPLES

Some examples are shown of analysis and optimization of a linear elastic plane stress problem. The aim is to show how domain decomposition methods may be used in this context. The examples correspond to small problems as they are preliminary results of these techniques.

### 6.1 Structural analysis by using the Schur complement matrix

A plane stress problem as shown in figure 4 has been solved by domain decomposition, computing the Schur complement matrix and solving a primal interface problem. It has been decomposed into four subdomains as shown in figure 6. The deformed structure is given in figure 5, the maximum horizontal displacement at the upper right corner is 6.8924, which coincides with the result of the standard finite element solution.

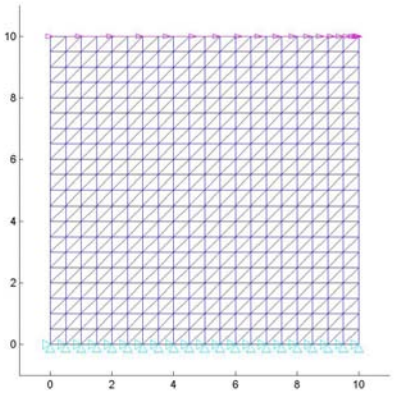


Figure 4: Plane stress problem: mesh and boundary conditions

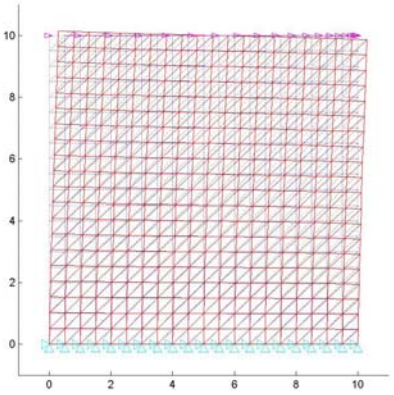


Figure 5: Domain decomposition analysis of the plane stress problem: deformed mesh

### 6.2 Structural analysis by minimization of the interface displacement errors

The same problem as previously described has been solve by minimizing the error in compatibility of the interface displacements. This time the structure has been divided in two subdomains (figure 7). Figure 8 gives the deformed mesh, where the upper right corner displacement was 6.3212. The tolerance for the function evaluation was 1.E-5. Figure 9 shows the evolution of a norm of the error in the interface displacements, for the different iterations.

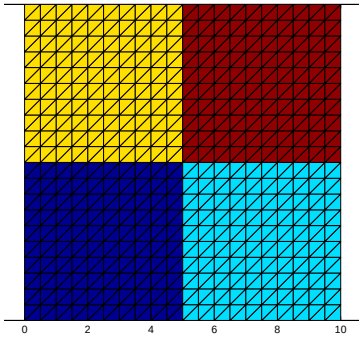


Figure 6: Domain decomposition analysis for the plane stress problem of figure 4

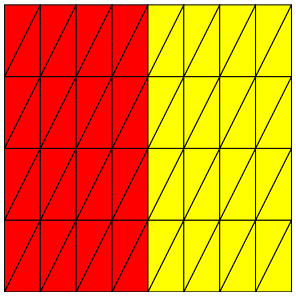

Figure 7: Domain decomposition for analysis with minimization of the interface displacements errors

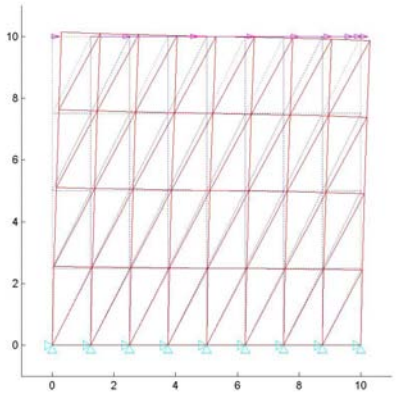


Figure 8: Analysis with minimization of the interface displacements errors: deformed mesh

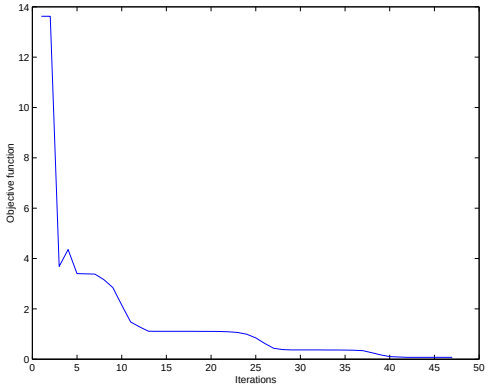


Figure 9: Domain decomposition for analysis with minimization of the interface displacements errors, norm of the displacement errors at interface

### 6.3 Structural optimization using interior point algorithms

Structural optimization using interior point algorithms is illustrated with an elastic plane stress cantilever, subject to concentrated loads as shown in figure 10, where the finite element mesh is also drawn. The beam dimensions are 10x10 units and the initial thickness is uniform with unit value. The initial volume, which is the objective function is 100.

The finite element mesh is composed by 128 triangular (linear) elements and 81 nodes. The total variables for the optimization problem is 272, corresponding 128 to the element thickness and 144 to the nodal displacements (state variables).

The initial values for the state variables correspond to an equilibrated configuration.

Inequality constraints are defined so as to limit the element stress under a maximum value and the element thickness are bounded by a minimum value 0.01.

A feasible directions interior point strategy has been followed to solved this problem.

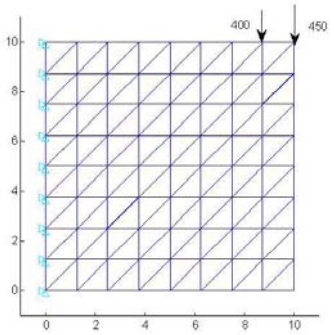


Figure 10: 2D cantilever beam

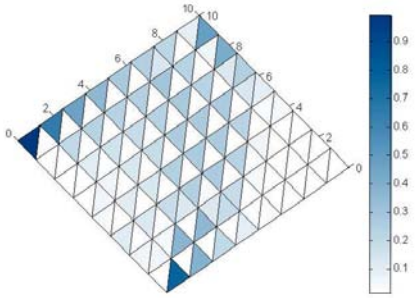

Figure 11: Optimization of a cantilever beam: final element thickness

After 67 iterations the beam volume is 14.65, the gradient of the lagrangian reached a value of 3.62359e-005 and the residual of the equilibrium equation, 4.22405e-014.

Figures 12 and 13 show the evolution, with iterations, of the objective function and a norm of the residual of the equilibrium equations.

The final thickness map is shown in figure 11. It can be seen that it roughly (with the limitations of the finite element size, and with element uniform thickness) corresponds to what is expected for a beam like the example.

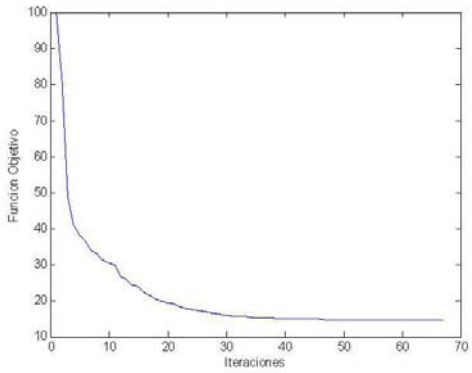


Figure 12: Optimization of a cantilever beam: evolution of the beam volume with iterations

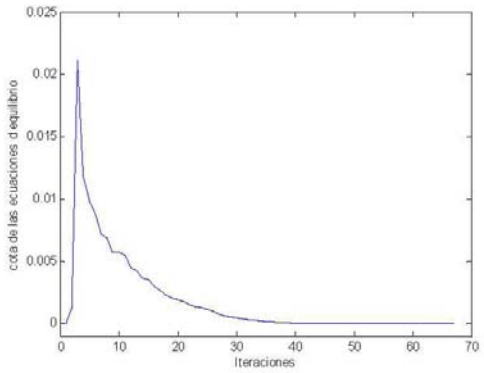


Figure 13: Optimization of a cantilever beam: evolution of a norm of the residual forces with iterations

## 7 CONCLUSIONS

A summary of domain decomposition methods has been given in this paper. Structural problems are treated in the work and in this context both analysis and optimization problems are addressed. Domain decomposition is used to solve these problems. Some examples are shown for analysis of a linear elastic plane structure, solved by direct computing of the interface variables, or by minimization of the interface error in displacement compatibility. On the other

hand an interior point technique has been used for thickness optimization of the same structural type. The use of domain decomposition strategy for this last problem has been outlined. Results are not shown since they are work in progress. The next step will be to make use of these techniques in the frame of parallel computation.

## REFERENCES


[1] Bjørstad, P.E. and Widlund, O.B., "Iterative Methods for the Solution of Elliptic Problems on Regions Partitioned into Substructures", SIAM J.Num.Analysis, Vol. 23, pp. 1097-1120 (1986).

[2] Farhat, Ch., Roux, F.-X., "A Method of Finite Element Tearing and Interconnection and its Parallel Solution Algorithm", Int.J.Numerical Methods in Engineering, Vol.32, pp 1205-1227 (1991).

[3] Herskovits J., *A View On Nonlinear Optimization*, in "Advances in Structural Optimization", vol 25 pp 71-116 Editora Kluwer Academic Publishers - Holland 1995

[4] Herskovits J. and Santos, G., *Feasible Arc Interior Point Algorithms for Nonlinear Optimization*, IV World Congress on Computational Mechanics, Buenos Aires, 1998.

[5] Herskovits J., *A Two-stages Feasible Directions Algorithm for Nonlinear Constrained Optimization*, Mathematical Programming, Vol. 36, pp. 19-38, (1986).

[6] Le Tallec, P. and Laporte, E., *Numerical methods in sensitivity analysis and shape optimization*, Springer Verlag, 2002.

[7] Le Tallec, P., De Roeck, Y.-H., Vidrascu, M. "Domain Decomposition Methods for Large Linearly Elliptic 3D Problems", Tech.Report TR/PA/90/20, Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, Toulouse, France (1990).

[8] Mandel J., "Balancing Domain Decomposition", Communications in Numerical Methods in Engineering, Vol. 9, pp 233-241 (1993).

[9] Smith B.F., "An Optimal Domain Decomposition Preconditioner for the Finite Element Solution of Linear Elasticity Problems", SIAM J. Sci.Stat.Comput., Vol 13 (1991).

[10] Sonzogni, V.E., "Domain Decomposition Techniques in Structural Mechanics", Proc. XV CILAMCE: Congresso Ibero Latino-Americano sobre Metodos Computacionais para Engenharia, pp. 1167-1176. 30 nov- 2 dic, 1994, Belo Horizonte (Brasil) (1994).

[11] Sonzogni, V.E., Sanchez, P. and Storti, M.A. "Resolución de grandes sistemas de ecuaciones en un cluster de computadoras", Proc. ENIEF 2004, 8-11 nov- 2 dic, 2004, Bariloche, Argentina (2004).

[12] Sonzogni, V.E. and Queizan, A.,"Simultaneous analysis and design of large structures by domain decomposition" $6^{th}$ World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, 30 May - 03 June 2005, Brazil, (2005)

[13] Storti M., Dalcín L., Paz R., Yommi A., Sonzogni V., 2002, "An interface strip preconditioner for domain decomposition methods", Mecánica Computacional Vol. XXI, pp. 2991-3007, (2002).