Aircraft fuselage sizing with multilevel optimization

B. Colson*, M. Porcelli †and Ph. L. Toint ‡

3 May 2013

In this technical report we describe the activity research carried out in the years 2010-2012 at Department of Mathematics, University of Namur, Namur (Belgium) in collaboration with LMS Samtech, Angleur (Belgium), in the framework of the project "Méthodes de résolution de problèmes d'optimisation de grande taille pour les structures en matériaux composites" (Acronym LARGO "LARge-scale Optimization problems"). LARGO was granted by the Walloon Region and LMS Samtech in the context of the First Program (convention number 916981).

1 Overview of the LARGO project

We address the optimization problem of sizing an aircraft fuselage and we focus on the problem of computing the dimensions of the different elements constituting a fuselage minimizing the total mass subject to some constraints. These constraints are mechanical stability constraints criteria (e.g. damage tolerance, buckling and post buckling) and they are formulated using Reserve Factors (RF): usually a structure is validated provided all its RFs are greater than one. These functions can be evaluated analytically by dedicated software but practically they have to be considered as black-box functions, i.e. as unknown functions whose corresponding outputs can be obtained from a given list of inputs without knowing its expression or internal structure. Moreover these functions are computationally-expensive since their evaluation is rather costly and plays the major role in the solution of the optimization problem.

In the first stage of the project, we studied the features of the involved functions and analyzed the problem structure. This analysis yielded to confirm that the problem possesses a natural *hierarchical structure* that could be exploited in the design of its numerical solution. In fact, classically, this problem has been addressed using a decomposition approach: individual components of the problems are optimized separately, hence without considering the entire hierarchy, see [6] for a survey on this topic. This approach may lead to designs that are optimal with respect to individual component demands but also to non-optimal structures that are a combination of such individually optimal components. We proposed to tackle the problem by optimizing the problem at the global level exploiting at the same time its *multilevel structure*. To this purpose, in the second stage of the project, we considered the class of multilevel methods and in this class we propose a new algorithm, named AL-RMTR (Augmented Lagrangian Recursive Multilevel Trust Region), that is suitable to solve the addressed aeronautical optimization.

*LMS, A Siemens Business - LMS Samtech - Rue des Chasseurs Ardennais, 8 - B-4031 Angleur (Belgium). Email: benoit.colson@lmsintl.com

†Namur Center for Complex Systems (NAXYS), Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", ISTI-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy. Email: margherita.porcelli@isti.cnr.it

‡Namur Center for Complex Systems (NAXYS), University of Namur, 61, rue de Bruxelles, B-5000 Namur, Belgium. Email: philippe.toint@fundp.ac.be

The last stage of the project was dedicated to the numerical validation of AL-RMTR with particular attention to its application to the solution of the optimization problem under consideration. This comprises the development of an efficient implementation of AL-RMTR, its integration with BOSS quattro and COMBOX (software platforms available at LMS Samtech), numerical experiments on test problems and comparison of its performance with those of the optimization packages available in BOSS quattro.

1.1 Outline of the document

In this report, we introduce the optimization problem under consideration in Section 2; in Section 3 we review the main ideas of the multigrid approach and we present the new AL-RMTR algorithm for solving general nonlinear programming problems. Section 4 is dedicated to the presentation of the numerical experiments. Finally, we give some conclusions in Section 5.

2 The optimization problem

An aircraft fuselage is mainly made of stiffened panels, i.e. thin shells reinforced in the orbital direction by stiffeners called frames or ribs and in the effort direction (i.e. orthogonally to the ribs) by stiffeners of smaller sections called stringers. Two consecutive ribs identify one ring of a fuselage that generally is composed by about twenty rings. Each ring may be considered as decomposed into elementary parts, called super-stringers, which consist in the theoretical union of one stringer and two half panels. Each super-stringer is characterized by some Design Variables (DV) which are firstly local geometry parameters such as panel thickness and stringer height and secondly composite laminate variables such as panel laminate percentages, see Figure 1.

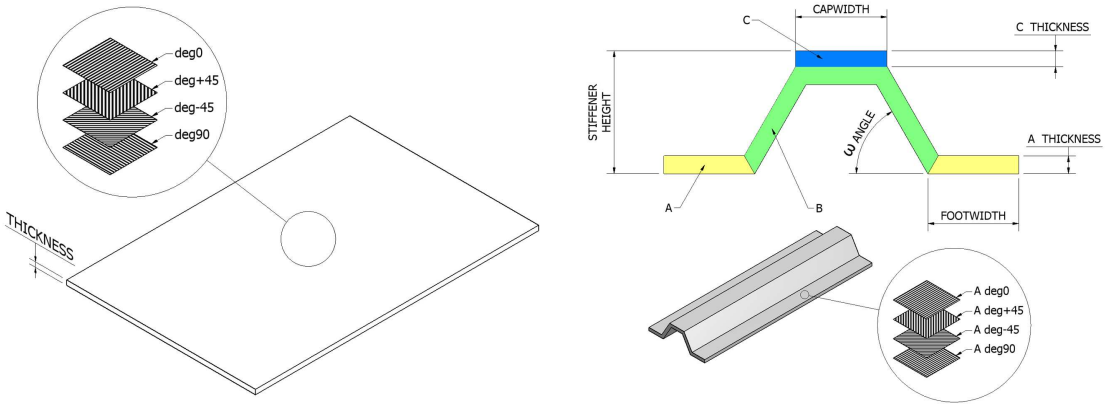

Figure 1: Super-stringer and DVs.

Remarkably, we can reduce the total number of DVs needed to describe the problem by grouping panels and stringers with common features into regions of panels and regions of stringers respectively, so that members of the same region share the same DVs, see Figure 2. Therefore, we can define a hierarchical description of the model by aggregating the elements in different ways. In particular, we may have *coarse* description of the model by defining a small number of

regions, i.e. using a small set of DVs, or we may have a *fine* description of the same model by using a larger number of regions, i.e. a larger set of DVs.

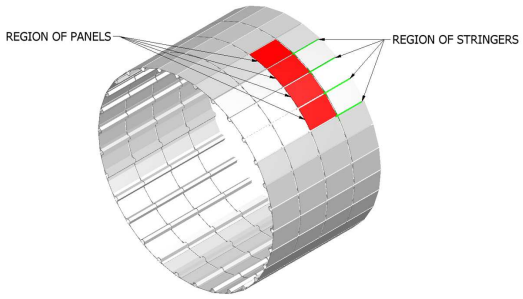

Figure 2: A region with 4 panels and a region made of 4 stringers.

We consider the problem of computing the optimal size (and more precisely minimizing the total mass) of these super-stringers considering different load cases, i.e. different load configurations of the aircraft. Denoting the full list of DVs characterizing a design by a vector x of DVs, we address the following optimization problem

$$\begin{array}{ll} \text{Minimize} & M(x) \\ \text{subject to} & RF(x) \geq 1 \\ & l \leq x \leq u, \end{array} \tag{1}$$

i.e. minimize the overall mass M subject to strength and geometrical constraints formulated by RFs. The vectors l and u define lower and upper bounds on the variables.

The objective function M is computed at the global level, calculating the total weight of the Finite Element model being used for the internal loads analysis, and results to be a piecewise linear function. The management of the RFs constitutes one of the most difficult task in the solution of problem (1). In fact, their computation is performed running a set of Fortran codes (the "black-box" functions mentioned above) which evaluate the function analytically. Input of these codes includes both local geometry and local internal loads. The codes commonly used for computing the RFs are Fortran implementations called *skill tools*. These tools are generally time-consuming and therefore interpolation algorithms have been proposed to save some time in the computation of the RFs (*rapid sizing* techniques).

Since we do not have any a priori knowledge of the regularity of the involved functions, we performed a local parametric analysis of the skill tools to investigate more their geometry. From this analysis, it turned out that most of the RFs are "smooth" but there are also RFs which present serious discontinuities and poles. This "non-smoothness" might be ascribed to "if-then-else" procedures in the skill tools or to some physical reasons. In Figure 3, we plot 2D section of two RFs (the damage tolerance RF and the strain RF) obtained by selecting two DVs (0° and the 90° ply percentage).

3 The multilevel (multigrid) approach

Techniques have been developed in the literature to solve linear systems to exploit the case where the problem hierarchy arises from the multilevel discretization of an underlying continuous

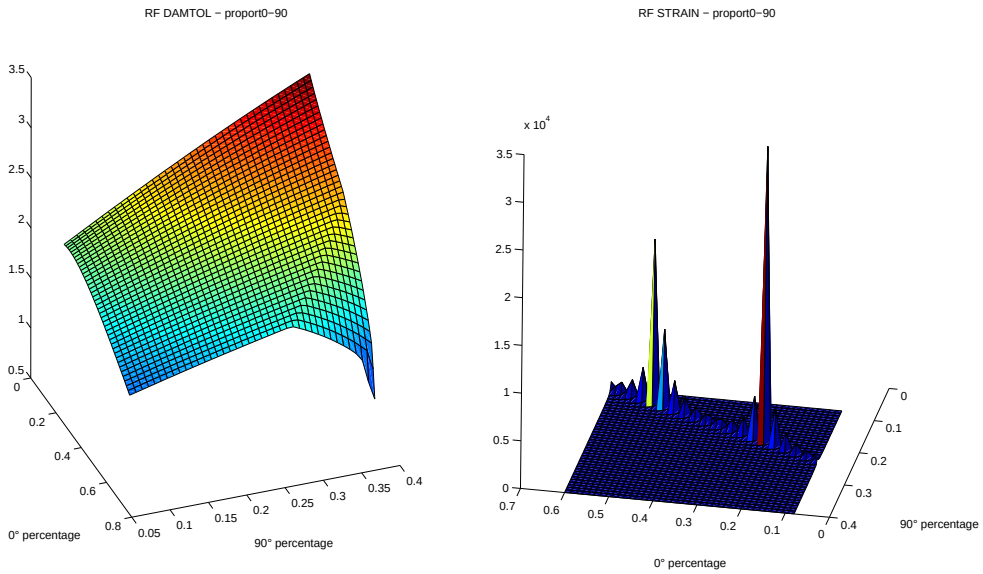

Figure 3: 2D sections of two RFs.

problem, see [2]. These methods are known as multigrid methods. This well-researched field, is based on a double observation: on one hand there exist iterative solution methods (called smoothers) which are very efficient at reducing the high-frequency, oscillatory components of the error while being possibly very inefficient at reducing their smooth, low-frequency part (the Jacobi and Gauss-Seidel methods are preeminent examples); on the other hand, the definition of a high frequency component is intrinsically tied to the discretization grid since the finer the grid, the higher the frequency representable on this grid. Multigrid methods then proceed by using smoothers to reduce the oscillatory error components on a fine grid, and then consider the remaining smooth components on this fine grid as oscillatory ones on a coarser grid. Broadly speaking, these can again be eliminated using smoothers on the coarser grid, and this technique may be applied recursively. One of the main attractions of multigrid methods for linear systems is that their workload increases only linearly with problem size, a feature crucial to the solution of very large instances.

Exploiting hierarchical problem structure in optimization is much more recent and, in particular, to our knowledge, the algorithms proposed for constrained problems are very few [8, 9, 11, 14]. We focused on the Recursive Multilevel Trust Region (RMTR) method proposed in [7] for unconstrained problems and extended to bound-constrained problems in [8, 9]. In fact, RMTR possesses good global convergence properties and resulted very efficient and reliable in solving very large problems for which a hierarchy of description exists. Typical cases are infinite-dimensional problems for which the levels of the hierarchy correspond to discretization levels, from coarse to fine. Moreover, the method has been efficiently implemented into a Fortran 90 package made available by the authors that can be used as a basis for a new multilevel code to solve generally constrained problems.

We refer the reader to [9, 13] for a full description of RMTR for solving bound-constrained problems and in the next section we just give the main idea of the procedure.

3.1 The Recursive Multilevel Trust Region method

RMTR belongs to the class of trust-region methods and then proceed iteratively by minimizing a model of the objective function in a region where the model can be trusted and which is defined in a specific norm. Moreover, the method is able to exploit the problem structure when the problem at hand can be decomposed into hierarchical levels. If fact, it is assumed that an appropriate hierarchy of descriptions is known for the problem under consideration. Suppose that a collection of functions $\{f_i\}_{i=0}^r$, is known, where $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, is twice-continuously differentiable, $n_i \geq n_{i-1}$ and f_i is "more costly" to minimize than f_{i-1}, for each $i = 1, \ldots, r$. This may be because f_i has more variables than f_{i-1} (as would typically be the case if the f_i represent increasingly finer discretizations of the same infinite-dimensional objective), or because the structure (in terms of partial separability, sparsity, or eigenstructure) of f_i is more complex than that of f_{i-1}, or for any other reason. To fix terminology, we will refer to a particular i as a level. Then, we assume that $n_r = n$ and $f_r(x) = f(x)$ for all $x \in \mathbb{R}^n$ giving back to the original problem. The number of levels r is strongly problem-dependent (typical values vary from 3 to 10) and, in general, the only requirement is that the problem at coarsest level is meaningful in the context of the problem at hand.

Moreover, some relationship between the variables of f_{i-1} and f_i must exist. So, we assume that there exist two full-rank linear operators $R_i : \mathbb{R}^{n_i} \to \mathbb{R}^{n_{i-1}}$ (the restriction) and $P_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$ (the prolongation) such that

$$\sigma_i P_i = R_i^T, \tag{2}$$

for some constant $\sigma_i > 0$, $i = 1, \ldots, r$. Moreover, the restriction operator is normalized to ensure $\|R_i\|_\infty = 1$ and the entries of R_i and P_i are nonnegative. We consider linear operators since they work well in practice and may represent interpolation operators. From a theoretical point of view, we might consider more general operators granted that they have bounded condition number. Figure 4 represents a typical multilevel scheme.

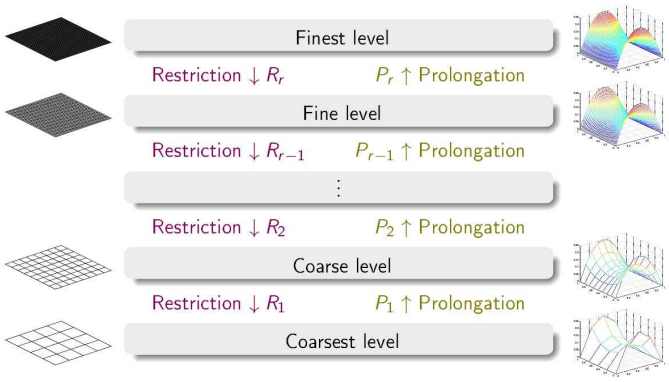

Figure 4: Multilevel scheme.

Being based on the trust-region framework, the RMTR method computes its steps by (approximately) minimizing some models inside some trust regions. However it uses a two-pronged strategy to define the model at a given iteration k. First, it may use (as many practical trust-

region algorithms) the quasi-Newton quadratic model

$$m_k(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s,$$

where $B_k \in \mathbb{R}^{n \times n}$ is a symmetric approximation of the objective Hessian $\nabla^2 f(x_k)$. A sufficient decrease in this model inside the trust region is then obtained by (approximately) solving the problem

$$\min \ m_k(x_k + s), \quad \text{s.t. } \|s\|_\infty \leq \Delta_k, \ l \leq x + s \leq u,$$

given some trust-region radius Δ_k. The philosophy of the RMTR method is however to use the multilevel hierarchy to efficiently construct minimization steps. This yields the second strategy to compute an appropriate step. More precisely, considering $x_{i,k}$, the k-th iterate at level $i > 0$, we first build a local coarse model $h_{i-1}(x_{i-1,0}+s_{i-1})$ around the restricted point $x_{i-1,0} = R_i x_{i,k}$. We then minimize this model (using a trust-region method) inside a coarse set of bound constraints $\mathcal{L}_i$, which represent both the feasibility with respect to the original problem bound constraints, and the constraints on the step size inherited from the trust regions of the finer levels. Let $x_{i-1,*}$ thus be the (approximate) solution of this local coarse subproblem at level $i-1$, given by

$$\min \ h_{i-1}(x_{i-1,0} + s_{i-1}), \quad \text{s.t. } x_{i-1,0} + s_{i-1} \in \mathcal{L}_i. \tag{3}$$

The coarse step $s_{i-1} = x_{i-1,*} - x_{i-1,0}$ is finally prolongated (using P_i) into a trust-region step s_i at level i.

Although other strategies are possible and available in the RMTR package, the code uses by default the Galerkin approximation to define the local coarse model, i.e. a restricted version of the current level quadratic Taylor's model:

$$h_{i-1}(x_{i-1,0} + s_{i-1}) = s_{i-1}^T R_i g_{i,k} + \frac{1}{2} s_{i-1}^T R_i \nabla^2 h_i(x_{i,k}) P_i s_{i-1}.$$

This choice is covered by the theory and it has better performance compared to other tested models. Interestingly, the Galerkin approximation does not require evaluation of f_{i-1} or its derivatives.

Except at the coarsest level where the subproblem is solved with a projected truncated conjugate gradient method (PTCG) [5], the subproblem is solved with a method inspired by the Gauss-Seidel method. Indeed, the model is successively minimized along the coordinate axis, yielding the Sequential Coordinate Minimization (SCM). This process has been shown to act as a smoothing procedure [7].

Moreover, the RMTR procedure leaves flexibility in the way to choose between the coarse model h_{i-1} and the Taylor's model $m_{i,k}$ and in the recursion pattern, allowing for free or fixed (V-cycles) recursion. In a V-cycle, minimization at lower levels (above the coarsest) consists of one successful smoothing iteration followed by a successful recursive iteration, itself followed by a second successful smoothing iteration. In particular, we considered three implementations: the full multilevel (FM) strategy where the choice to use the coarse model may be associated with that to start from a good initial point (Figure 5), the multilevel on finest (MF) strategy, where we apply the RMTR algorithm on the finest level (Figure 6) and the all on finest (AF) strategy which is a standard Newton trust-region algorithm (with PTCG as subproblem solver) applied at the finest level, without recourse to coarse-level computations.

3.2 The bound-constrained Augmented Lagrangian approach

Clearly, the application of the RMTR algorithm to solution of the general constrained problem (1) is not trivial since we need to take into account the presence of the nonlinear inequality

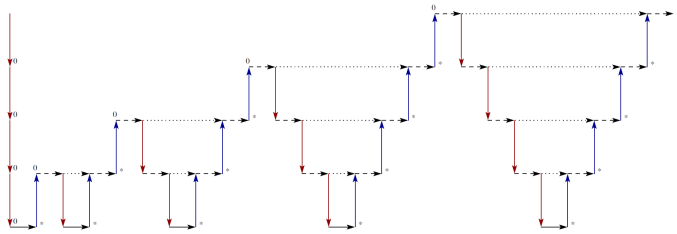

Figure 5: Scheme of the Full Multilevel strategy.

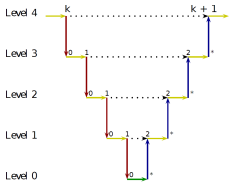

Figure 6: Scheme of the Multilevel on Finest strategy.

constraints. The existing class of methods to solve problem (1) comprises Augmented Lagrangian methods, Sequential Quadratic Programming methods and Interior Point methods, [12]. We choose to consider a bound-constrained Augmented Lagrangian approach [4, 5] which handles a general constrained problem by solving a sequence of bound-constrained subproblems.

The bound-constrained Augmented Lagrangian approach is proposed to solve the equality constrained problem

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & f(x) \\ \text{subject to} & h(x) = 0, \\ & l \leq x \leq u, \end{array} \tag{4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are smooth function. The basic idea of this strategy is to incorporate only the equality constraints in (4) in the Augmented Lagrangian, that to is

$$\mathcal{L}_A(x, \lambda; \mu) = f(x) - \lambda^T h(x) + \frac{\mu}{2} h(x)^T h(x), \tag{5}$$

where $\lambda \in \mathbb{R}^m$ is the vector of the Lagrangian multipliers and the positive scalar μ denotes the penalty parameter. The kth iteration of the algorithm consists in fixing the penalty parameter to some value $\mu_k > 0$ fixing λ at the current estimate λ_k, and performing the minimization with respect to x, i.e. compute x_{k+1} that solves the subproblem

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & \mathcal{L}_A(x, \lambda_k; \mu_k) \\ \text{subject to} & l \leq x \leq u. \end{array} \tag{6}$$

Note that the first-order optimality condition for x to be a solution of problem (6) is that (for fixed λ and μ)

$$x - \mathcal{P}_{[l,u]}(x - \nabla_x \mathcal{L}_A(x, \lambda; \mu)) = 0,$$

where $\mathcal{P}_{[l,u]}(v)$ is the projection onto the box $[l, u]$ defined as $\mathcal{P}_{[l,u]}(v) = \max\{l, \min\{v, u\}\}$.

At each iteration, the subproblem (6) is solved with an increasing accuracy. The parameters λ_k and μ_k are updated taking into account the constraint violation measure $\|h(x_k)\|_\infty$: if it is

sufficiently "small", μ_k is left unchanged and λ_k is modified as $\lambda_{k+1} = \lambda_k - \mu_k h(x_k)$, otherwise λ_k is left unchanged and μ_k is increased by a fixed factor, see [5]. These updates ensure convergence of the scheme and the penalty parameter is guaranteed to remain bounded away from zero.

This scheme has been implemented in the well-known LANCELOT software package [5] which has been intensively tested giving excellent numerical results showing that the bound-constrained Augmented Lagrangian approach is reliable and efficient.

3.3 The Augmented Lagrangian RMTR (AL-RMTR) method

The problem (1) under consideration, can be classified as an optimization problem with simple bound and general nonlinear inequality constraints of the form

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & f(x) \\ \text{subject to} & c(x) \leq 0, \\ & l \leq x \leq u, \end{array} \tag{7}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$. For the solution of this problem, we propose an algorithm called AL-RMTR that combines the bound-constrained Augmented Lagrangian approach and the RMTR algorithm presented above. From these methods, AL-RMTR inherits the convergence properties and, at the same time, is able to exploit an available multilevel structure of problem (7).

In fact, we assume not only the availability of multilevel structure in the variable space, i.e. a collection of objective functions $\{f_i\}_{i=0}^r$, where $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, but also the availability of such structure in the constraint (i.e. Lagrange multiplier) space. Then we assume the existence of a collection of coarse constraint functions $\{c_i\}_{i=0}^r$, with $c_i : \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}$ where m_i is the number of constraints at level i ($m_i \geq m_{i-1}$) and $c_r(x) = c(x), m_r = m$.

Therefore, in addition to the transfer operators (2) for variables, we assume the existence of a relationship between the constraints c_{i-1} and c_i, i.e. the existence of a restriction operator $R_i^c : \mathbb{R}^{m_i} \to \mathbb{R}^{m_{i-1}}$ and a prolongation operator $P_i^c : \mathbb{R}^{m_{i-1}} \to \mathbb{R}^{m_i}$ such that

$$\sigma_i^c P_i^c = R_i^{cT}, \tag{8}$$

for some constant $\sigma_i^c > 0$, $i = 1, \ldots, r$. Moreover, to each constraint c_i we can associate the corresponding coarse approximation of the Lagrange multipliers $\lambda_i \in \mathbb{R}^{m_i}$ and we can relate λ_i with the coarser λ_{i-1} with the operators P_i^c and R_i^c.

It is important to remark that the lagrangian multipliers associated with continuous constraints are not necessarily continuous and may exhibit δ-function-like behaviour. Therefore, when the operators R_i^c and P_i^c are linear interpolation operators (as it is commonly the case), the multipliers are approximated by a piece-wise linear function and this may not fully capture their behaviour. We followed a strategy proposed in [1] where the multipliers are "smoothed" before applying the Prolongation-Restriction operator and the inverse of the Laplacian operator Δ^{-1} is used as a smoother. For instance, the computation of the lagrangian multiplier at the lower level $i-1$ is made from the value at the upper level i as follows:

$$\lambda_{i-1} = \Delta R_i^c \Delta^{-1} \lambda_i,$$

see Figures 7 and 8.

The fist step to use the Augmented Lagrangian function in (3.1), is to reformulate problem (7) as problem (4), that is to convert the nonlinear inequalities $c(x) \leq 0$ into the equalities

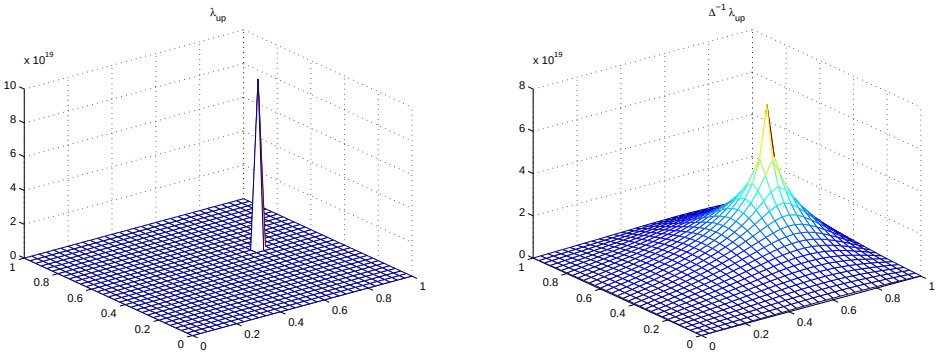

Figure 7: Smoothing of the Lagrange multplier: $\lambda_i \overset{\text{smooth}}{\longrightarrow} \Delta^{-1}\lambda_{up}$.

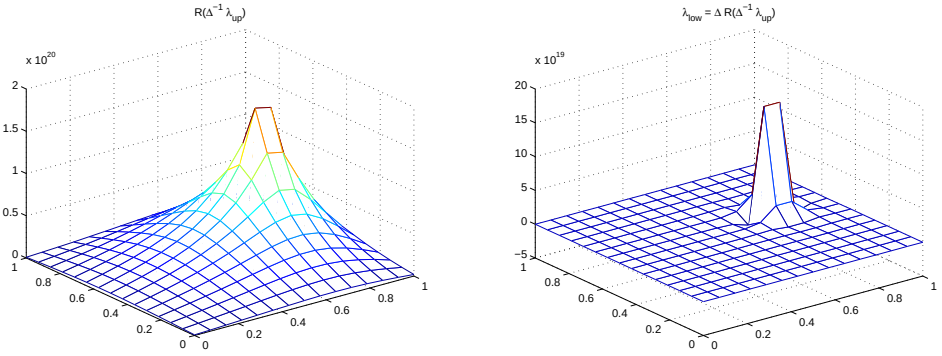

Figure 8: Computation of the restricted smoothed lagrangian multiplier: $R(\Delta^{-1}\lambda_{up}) \longrightarrow \lambda_{low} = \Delta R \Delta^{-1}\lambda_{up}$.

$h(x) = 0$. We propose two reformulations: the first (`R-max`^{2}) is obtained using the continuously differentiable function $[t]_+ = \max\{t, 0\}^2/2$ so that h takes the form

$$h(x) = \max\{c(x), 0\}^2/2; \tag{9}$$

the second (`R-slack`) is obtained adding a nonnegative slack variable $s \in \mathbb{R}^m$ so that h is defined on $\mathbb{R}^{n+m}$ and takes the form

$$h(x, s) = c(x) + s, \tag{10}$$

with the additional bound constraint $s \geq 0$ which must be added to the simple bounds $l \leq x \leq u$. We denote the new bounds as $\hat{l} = [l, 0]$ and $\hat{u} = [u, \infty]$.

The choice of using `R-max`^{2} or `R-slack` yields to different exploitable structures for the gradient and the Hessian of the Augmented Lagrangian function. Concerning the gradient, let J_c and J_h denote the Jacobian matrices of c and h in (7) and (4) respectively. Then, if `R-max`^{2} is used, the gradient of the Augmented Lagrangian function takes the form

$$\nabla \mathcal{L}_A(x, \lambda; \mu) = \nabla f(x) - J_h^T \lambda + \mu J_h^T h(x),$$

where

$$(J_h)_j = \max\{((c(x))_j, 0\}(J_c)_j, \quad j = 1, \ldots, m$$

and $(J)_j$ denotes the jth row of a matrix J and $(c(x))_j$ is the jth component of the vector $c(x)$. On the other hand, by using `R-slack` we have the following structure for $\nabla\mathcal{L}_A$:

$$\nabla\mathcal{L}_A((x,s),\lambda;\mu) = \begin{pmatrix} \nabla f(x) \\ 0 \end{pmatrix} - \begin{pmatrix} J_c^T \\ I_m \end{pmatrix} \lambda + \mu \begin{pmatrix} J_c^T \\ I_m \end{pmatrix} h(x,s).$$

It is important to remark that `R-slack` involves an increase in the number of variables from n to $n+m$. In this case, we define the prolongation/restriction operator in the variable space $\mathbb{R}^{n+m}$ blockwise, as e.g for the restriction,

$$\begin{pmatrix} R_i & \\ & R_i^c \end{pmatrix}, \tag{11}$$

so that R_i acts on the original variables x and R_i^c is applied on the slack variables s.

Once problem (7) has been reformulated as (4), one can straightforwardly apply the bound-constrained Augmented Lagrangian scheme and use the RMTR algorithm to solve the subproblems (6). In particular, fixed μ, if h_i is computed from c_i by (9), then at each level i the coarse Augmented Lagrangian $\mathcal{L}_{A,i} : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}$ is given by

$$\mathcal{L}_{A,i}(x,\lambda;\mu) = f_i(x) - \lambda_i^T h_i(x) + \frac{\mu}{2} h_i(x)^T h_i(x),$$

whereas if h_i is computed from c_i by (10), then $\mathcal{L}_{A,i} : \mathbb{R}^{n_i+m_i} \times \mathbb{R}^{m_i} \to \mathbb{R}$ has the form

$$\mathcal{L}_{A,i}((x,s),\lambda;\mu) = f_i(x) - \lambda_i^T h_i(x,s) + \frac{\mu}{2} h_i(x,s)^T h_i(x,s).$$

The resulting AL-RMTR algorithm in sketched in Algorithm 3.1: the string `R-choice` denotes the reformulation used to convert inequalities into equalities, i.e. it is either `R-max`^{2} or `R-slack`, and τ_k is the following criticality measure

$$\tau_k = \|x_k - \mathcal{P}_{[l,u]}(x_k - \nabla_x\mathcal{L}_A(x_k,\lambda_k;\mu_k))\|_2, \tag{12}$$

for `R-max`^{2} and

$$\tau_k = \|(x_k,s_k) - \mathcal{P}_{[\hat{l},\hat{u}]}((x_k,s_k) - \nabla_x\mathcal{L}_A((x_k,s_k),\lambda_k;\mu_k))\|_2, \tag{13}$$

otherwise.

Concerning the implementation of Algorithm 3.1, we have to remark that it depends on the recurrence scheme chosen in the calling of RMTR and on the reformulation used to convert inequalities into equalities. If the MF recurrence is chosen (AL-RMTR-MF) or the AF version is used (AL-RMTR-AF), the Augmented Lagrangian and its derivatives needs to be evaluated only at the finest level whereas if the FM is employed (AL-RMTR-FM), coarser values of the Augmented Lagrangian (i.e. coarser values of the objective and constraint functions) and its derivatives have to be provided.

Moreover, if AL-RMTR-MF or AL-RMTR-AF are used with `R-max`^{2}, then the transfer operators R_i^c and P_i^c are not employed in practice. On the other hand, if `R-slack` is used, these operators R_i^c and P_i^c in the constraints space are always activated, at least in the variable space when using (11).

4 Experiments

In this section we report on the numerical experiments carried out in the third stage of LARGO. We first give a description of the test problem set pointing out the multilevel properties of the problems.

Algorithm 3.1: The AL-RMTR **algorithm:** k**th iteration**

Given `R-choice`, x_k, λ_k, μ_k and ω_k, η_k (and tolerances η_* and ω_*)

Step 1: Solve the bound-constrained subproblem with RMTR. If `R-choice=R-max`[2] , find an approximate solution x_{k+1} of

$$\begin{array}{ll} \min_x & \mathcal{L}_A(x, \lambda_k; \mu_k) \\ \text{subject to} & l \leq x \leq u. \end{array}$$

such that $\tau_{k+1} \leq \omega_k$.
Else, find an approximate solution x_{k+1} of

$$\begin{array}{ll} \min_{x,s} & \mathcal{L}_A((x, s), \lambda_k; \mu_k) \\ \text{subject to} & l \leq x \leq u, \\ & 0 \leq s, \end{array}$$

such that $\tau_{k+1} \leq \omega_k$.

Step 2: Check for convergence. If

$$\|h(x_{k+1})\|_\infty \leq \eta_* \text{ and } \tau_{k+1} \leq \omega_*$$

stop with approximate solution x_{k+1};

Step 3: Updates. If $\|h(x_{k+1})\|_\infty \leq \eta_k$, set $\lambda_{k+1} = \lambda_k - \mu_k h(x_k)$.
Else, set $\mu_{k+1} \geq \mu_k$. Decrease η_k and ω_k.

Two academic problems: MSP and vBRATU

We consider two optimization problems of the form

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & f(x) \\ \text{subject to} & c(x) \leq 0, \end{array} \tag{14}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is given by the discretization of two very well known PDE problems that will be described further on, and the constraint function $c : \mathbb{R}^n \to \mathbb{R}^m$ is constructed as follows. Let $x \in \mathbb{R}^n$ be a given surface discretized on a $N \times N$ grid ($n = N^2$) and for each internal node of the grid, let $(\chi(x))_j$ be the local curvature of x around the node j, for $j = 1, \ldots, m$ with $m = (N-2)^2$. Then we define the function $c : \mathbb{R}^n \to \mathbb{R}^m$ such that

$$(c(x))_j = ((\chi(x))_j - \bar{\chi})/m, \tag{15}$$

with $j = 1, \ldots, m$ and $\bar{\chi} \in \mathbb{R}$, and we impose the inequality $c(x) \leq 0$, i.e. that the scaled curvature of the surface x is upper bounded by some value $\bar{\chi}$. Note that the constraint function is locally defined on the grid and that both its dimension and its value depend on the dimension of the grid so that coarser versions of the constraints can be trivially defined varying N.

Concerning the objective function of problem (14), we consider the following function

$$f_{ms}(v) = \int_{[0,1]\times[0,1]} \sqrt{1 + \|\nabla_t v\|_2^2},$$

with $v \in H^1([0,1] \times [0,1])$ that satisfies the condition $v(t) = v_0(t)$ on $\partial([0,1] \times [0,1])$ with

$$v_0(x) = \begin{cases} t_1(1-t_1), & t_2 \in \{0,1\} \quad t_1 \in (0,1) \\ 0, & t_2 \in (0,1) \quad t_1 \in \{0,1\} \end{cases}$$

which is classically referred to the very well-known Minimal Surface problem, and

$$f_{vb}(v) = \int_{[0,1]\times[0,1]} \|\nabla_t v\|_2^2 + \beta v e^v,$$

with $\beta = 6.8$ and $v \in H^1([0,1] \times [0,1])$ has to satisfies the condition $v(t) = 0$ on $\partial([0,1] \times [0,1])$, which is related to a variant of the BRATU problem. Here, $H^1([0,1] \times [0,1])$ denotes the Hilbert space of all functions with compact support in the domain $[0,1] \times [0,1]$ such that if $v \in H^1([0,1] \times [0,1])$ then v and $\|\nabla v\|_2^2$ belong to $\mathcal{L}^2([0,1] \times [0,1])$.

For both problems, the surfaces are defined on the square $[0,1] \times [0,1]$ and we discretize them by a finite-element basis (P1) defined using a uniform triangulation of $[0,1] \times [0,1]$, with the same grid spacing along the two coordinate axes. The discretization of the function f_{ms} and f_{vb} constitutes the objective function f in (14) and we call MSP and vBRATU the resulting problems, respectively. Analogously to the constraints, coarser and finer description of f are obtained by using few or many nodes in the discretization grids.

In Figure 9 and 10, the unconstrained solutions of MSP and vBRATU (i.e. without bounding their curvature) and their curvature are plotted.

The industrial problem: RFUSE

We considered problem (1) applied to a small model consisting of a rectangular piece of an aircraft fuselage made of 6×8 panels and 7×8 stringers, see Figure 11 (model provided by LMS Samtech). For each panel and each stringer we take into account one DV, the panel thickness t

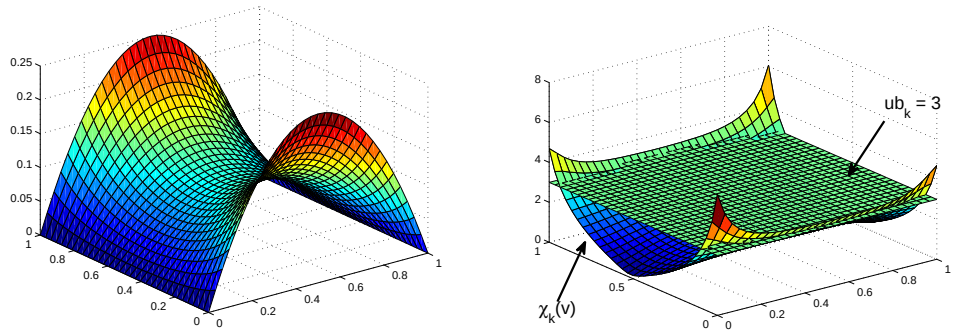

Figure 9: The unconstrained solution of MPS, its curvature and the upper bound $\bar{\chi} = 3$.

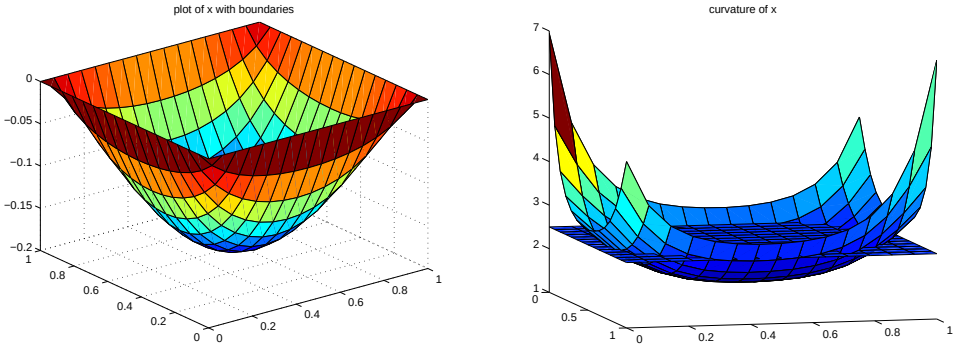

Figure 10: The unconstrained solution of vBRATU, its curvature and the upper bound $\bar{\chi} = 2.5$.

and the stiffener section s, respectively, so that the total number of DVs (and regions) is 104, i.e. $M : \mathbb{R}^{104} \to \mathbb{R}$. Concerning the constraints, we consider 3 RF constraints for each Calculation Point (CP) (combination of one stringer and its two adjacent panels) which are approximated by using the Rapid Sizing analysis tool. This way, we have 3 RF values per internal stringer, i.e. $RF : \mathbb{R}^{104} \to \mathbb{R}^{120}$. We will refer to this problem with the name RFUSE.

Now we give a possible multilevel description for RFUSE.

Let the panel be modelled as the grid of Figure 12 with $N_l = 8$ lines ad $N_c = 6$ columns where each sub-rectangle represents a panel and the vertical segments represent a stringer. For each panel, let $t_i, i = 1, \ldots, 48$ denote the panel thickness DV and let $a_i, i = 1, \ldots, 48$ be the panel area parameter; for each stringer, let $s_i, i = 1, \ldots, 56$ represent the stiffener section DV and $d_i, i = 1, \ldots, 56$ be the stringer length parameter.

In order to define coarser levels with smaller design variable space, we can suitably aggregate groups of panels and groups of stringers. Figure 13 shows the choice we follow: we group 4 panels into 1 and the corresponding 6 stringers into 2. This way, we have 3 different levels and the number of variables is $n_3 = 104$ at level 3 (the finest), $n_2 = 28$ at level 2 and $n_1 = 10$ at level 1 (the coarsest). Clearly, the border structural elements have to be considered separately: if the number of columns or the number of lines is not even, we only take 2 elements (or 1 if both are

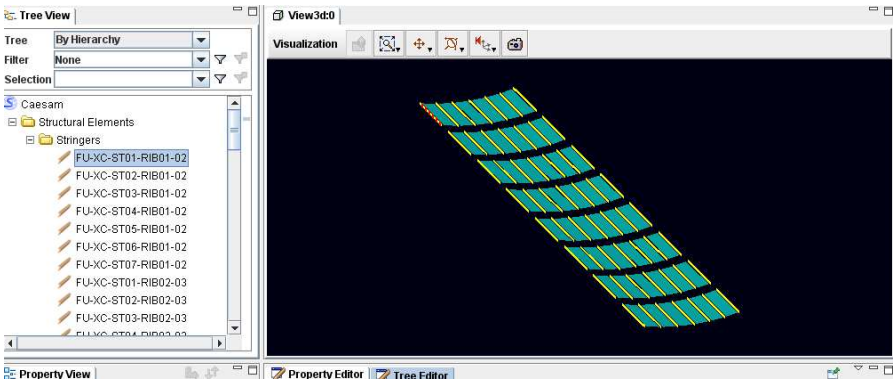

Figure 11: The RFUSE test model.

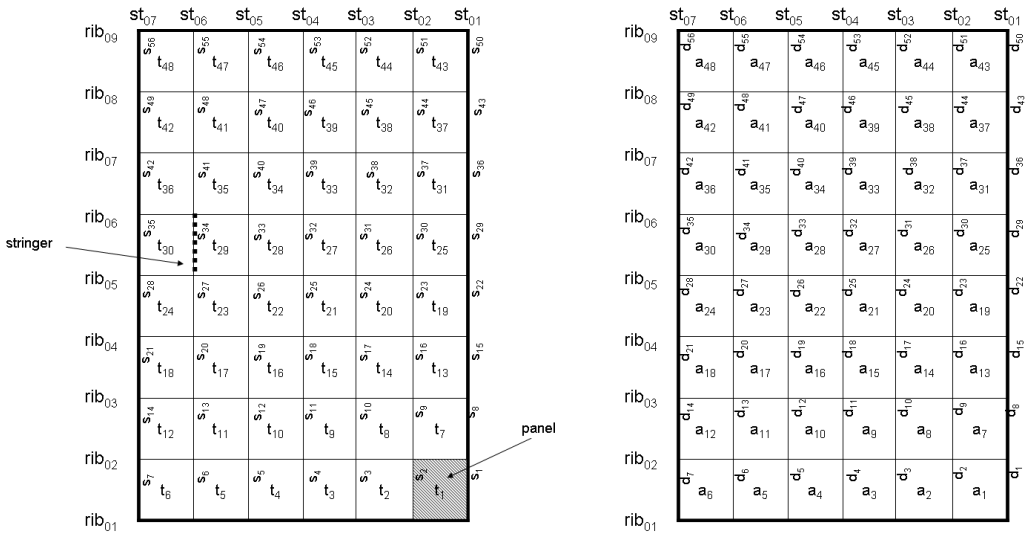

Figure 12: Design variables (left) and parameters (right).

odd).

Consider now the constraint space. For each CP (or internal stringers), we have 3 RFs which are denoted as RF^1, RF^2 and RF^3 and their 40 components are ordered as in Figure 14 (omitting the superscript 1,2,3). Applying the aggregation rule of Figure 13 to the CPs, the number of constraints at each level is: $m_3 = 3 \times 40 = 120$ at level 3, $m_3 = 3 \times 8 = 24$ at level 2 and $m_3 = 3 \times 2 = 6$ at level 1.

The number of DVs (t and s) and constraints (RF^1, RF^2 and RF^3) at each level are summarized in Table 1.

4.1 Numerical Results

In this section we study the the practical behaviour of AL-RMTR in comparison with that of the optimization packages available in BOSS quattro on the test problems described in the previous section. The implementation of AL-RMTR is based on Algorithm 3.1 and uses the

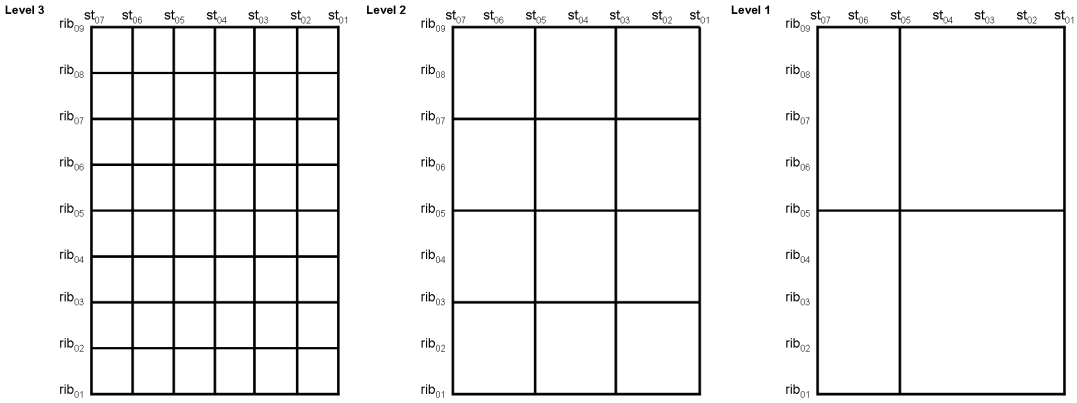

Figure 13: Multilevel structure.

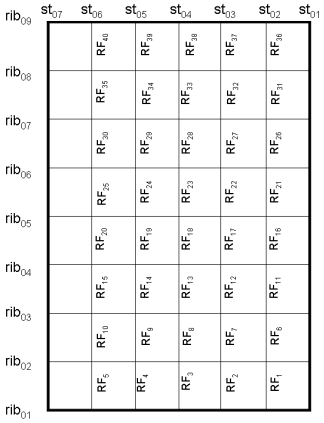

Figure 14: Reserve Factors.

packages RMTR [13] as a solver for Step 1 and LTS [10] for the Hessian approximation.

We performed 3 sets of experiments:

Experiment set 1: AL-RMTR and the BOSS quattro packages are compared in the solution of MSP using `R-max`^{2};

Experiment set 2: the two reformulations `R-max`^{2} and `R-slack` are tested by using AL-RMTR in the solution of vBRATU;

Experiment set 3: RFUSE is solved by AL-RMTR and GCM [3].

To measure the efficiency of the algorithms, we consider the CPU time and the equivalent number of finest evaluations

$$Ffe = \sum_{i=0}^{r} \frac{Ffe_i}{fe_r},$$

where F is a generic function and Ffe_i is the number of F-function evaluations at level i (r corresponds to the finest level), that takes into account that function evaluations at coarser

level	N_l	N_c	# t	# s	# DVs	# CPs	# RFs
3	8	6	48	56	104	40	120
2	4	3	12	16	28	6	24
1	2	2	4	6	10	2	6

Table 1: Multilevel dimensions for RFUSE.

levels are considerably cheaper than those at higher ones.

In the Experiment sets 1 and 2, we chose the finest grid size $N = 17$ for MSP and $N = 15$ for vBRATU. The operators P_i and P_i^c are chosen as linear interpolation operators imposing nonzero Dirichlet conditions on the boundary for MSP and interior conditions on the boundary for VBRATU. Then R_i and R_i^c are given by transposing P_i and P_i^c, respectively. Moreover, we set $\bar{\chi} = 3$ and $\bar{\chi} = 2.5$ in (15) for MSP and vBRATU, respectively. A summary of the multilevel dimensions is given in Table 2.

	MSP		vBRATU	
level	n	m	n	m
3	289	225	225	225
2	81	49	49	49
1	25	9	9	9

Table 2: Multilevel dimensions for MSP and vBRATU.

Concerning the Experiment set 3, we now describe two-block-restriction linear operators for the DVs of RFUSE which make use the panel area a and the stringer length d parameters.

To obtain the DVs at a coarser level, we define a two-block-restriction linear operator: one block acts only on the panel DVs and one block has effect only on the stringer DVs. The operator takes into account some of the given problem parameters, in particular the panel area a and the stringer length d. These parameters are aggregated through the levels in analogy with the corresponding DVs. From now on, assume that the DVs and the parameters are numbered as in Figure 12.

We now discuss the definition of the block of the restriction operator corresponding to the panel thickness t. Assume to aggregate the 4 "internal" variables $t_j^i, t_{j+1}^i, t_{j+N_c}^i, t_{j+N_c+1}^i$ at level i into the variable t_j^{i-1} at the coarser level $i-1$. Then we have

$$t_j^{i-1} = \frac{a_j^i t_j^i + a_{j+1}^i t_{j+1}^i + a_{j+N_c}^i t_{j+N_c}^i + a_{j+N_c+1}^i t_{j+N_c+1}^i}{a_j^i + a_{j+1}^i + a_{j+N_c}^i + a_{j+N_c+1}^i}$$

and for the corresponding parameter we have

$$a_j^{i-1} = a_j^i + a_{j+1}^i + a_{j+N_c}^i + a_{j+N_c+1}^i.$$

The boundary variables may be grouped analogously.

For the stringer section s, we consider different cases depending on the position of the variables in the model. In Figure 15 we use the superscripts lo and up to denote the "lower" and "upper" stringer, and the subscript L, C and R for the "left", "central" and "right" stringer, respectively. Referring to Figure 15, we define the following stringer elements at coarser level:

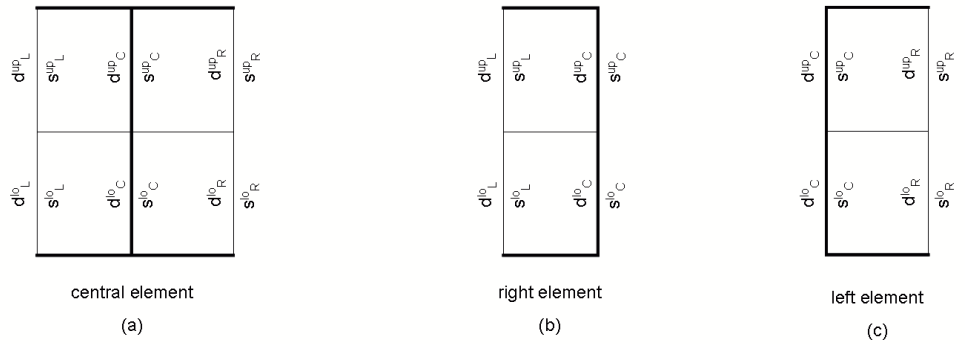

Figure 15: Grouping stringers: central element (a), right bound (b) and left bound (c).

- Central element (a):

$$s^{lo} = \frac{d_L^{lo}s_L^{lo}/2+d_C^{lo}s_C^{lo}+d_R^{lo}s_R^{lo}/2}{d_L^{lo}/2+d_C^{lo}+d_R^{lo}/2}, \quad d^{lo} = \frac{d_L^{lo}/2+d_C^{lo}+d_R^{lo}/2}{2},$$
$$s^{up} = \frac{d_L^{up}s_L^{up}/2+d_C^{up}s_C^{up}+d_R^{up}s_R^{up}/2}{d_L^{up}/2+d_C^{up}+d_R^{up}/2}, \quad d^{up} = \frac{d_L^{up}/2+d_C^{up}+d_R^{up}/2}{2},$$

$$s_C^* = \frac{d^{lo}s^{lo}+d^{up}s^{up}}{d^{lo}+d^{up}} = \frac{\left(d_L^{lo}s_L^{lo}/2+d_C^{lo}s_C^{lo}+d_R^{lo}s_R^{lo}/2\right)+\left(d_L^{up}s_L^{up}/2+d_C^{up}s_C^{up}+d_R^{up}s_R^{up}/2\right)}{\left(d_L^{lo}/2+d_C^{lo}+d_R^{lo}/2\right)+\left(d_L^{up}/2+d_C^{up}+d_R^{up}/2\right)}$$

- Right element (b):

$$s^{lo} = \frac{d_L^{lo}s_L^{lo}/2+d_C^{lo}s_C^{lo}}{d_L^{lo}/2+d_C^{lo}}, \quad d^{lo} = \frac{d_L^{lo}/2+d_C^{lo}}{3/2},$$
$$s^{up} = \frac{d_L^{up}s_L^{up}/2+d_C^{up}s_C^{up}}{d_L^{up}/2+d_C^{up}}, \quad d^{up} = \frac{d_L^{up}/2+d_C^{up}}{3/2},$$

$$s_R^* = \frac{d^{lo}s^{lo}+d^{up}s^{up}}{d^{lo}+d^{up}} = \frac{\left(d_L^{lo}s_L^{lo}/2+d_C^{lo}s_C^{lo}\right)+\left(d_L^{up}s_L^{up}/2+d_C^{up}s_C^{up}\right)}{\left(d_L^{lo}/2+d_C^{lo}\right)+\left(d_L^{up}/2+d_C^{up}\right)},$$

- Left element (c):

$$s^{lo} = \frac{d_C^{lo}s_C^{lo}+d_R^{lo}s_R^{lo}/2}{d_C^{lo}+d_R^{lo}/2}, \quad d^{lo} = \frac{d_C^{lo}+d_R^{lo}/2}{3/2},$$
$$s^{up} = \frac{d_C^{up}s_C^{up}+d_R^{up}s_R^{up}/2}{d_C^{up}+d_R^{up}/2}, \quad d^{up} = \frac{d_C^{up}+d_R^{up}/2}{3/2},$$

$$s_L^* = \frac{d^{lo}s^{lo}+d^{up}s^{up}}{d^{lo}+d^{up}} = \frac{\left(d_C^{lo}s_C^{lo}+d_R^{lo}s_R^{lo}/2\right)+\left(d_C^{up}s_C^{up}+d_R^{up}s_R^{up}/2\right)}{\left(d_C^{lo}+d_R^{lo}/2\right)+\left(d_C^{up}+d_R^{up}/2\right)}.$$

Note that right and left elements correspond to boundary stringers of the model. Analogously, for the stringer length parameter d we have

$$d_C^* = \frac{\left(d_L^{lo}/2+d_C^{lo}+d_R^{lo}/2\right)+\left(d_L^{up}/2+d_C^{up}+d_R^{up}/2\right)}{4}$$

for the central element, and for the right and left bound we have

$$d_R^* = \frac{\left(d_L^{lo}/2 + d_C^{lo}\right) + \left(d_L^{up}/2 + d_C^{up}\right)}{3}, \quad d_L^* = \frac{\left(d_C^{lo} + d_R^{lo}/2\right) + \left(d_C^{up} + d_R^{up}/2\right)}{3}.$$

The resulting restriction operator is very sparse and its infinity norm is always equal to 1 so that we can define the prolongation operator just as its transpose ($\sigma_i = 1$ in (2)). It is important to remark that the multilevel structure presented above may be generalized to any rectangular model, see Table 3.

level	`no. of lines`	`no. of columns`	`dim. of` t	`dim. of` s
i	N_l	N_c	$N_l \times N_c$	$N_l \times (N_c + 1)$
$i-1$	$Q_l + R_l$	$Q_c + R_c$	$(Q_l + R_l) \times (Q_c + R_c)$	$(Q_l + R_l) \times (N_c + R_c + 1)$

Table 3: $N_l = 2 * Q_l + R_l$ and $N_c = 2 * Q_c + R_c$.

Concerning the constraint aggregation, we designed the following three-block-restriction linear operator and for the sake of simplicity, we will give the details of R_3^c from level 3 to level 2. Assume, up to permutation, that the constraints are ordered so that

$$RF = (RF_1^1, \ldots, RF_{40}^1, RF_1^2, \ldots, RF_{40}^2, RF_1^3, \ldots, RF_{40}^3)$$

Then we define R_3^c of dimension 24×120 of the form

$$R_3^c = \begin{pmatrix} Q_3^c & & \\ & Q_3^c & \\ & & Q_3^c \end{pmatrix},$$

with blocks Q_3^c of dimension 8×40; each block Q_3^c can be written as

$$Q_3^c = \begin{pmatrix} T_3^c & & & \\ & T_3^c & & \\ & & T_3^c & \\ & & & T_3^c \end{pmatrix},$$

with T_3^c of dimension 2×10 defined as

$$T_3^c = \begin{pmatrix} 1 & 1 & 1/2 & 0 & 0 & 1 & 1 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1 & 1 & 0 & 0 & 1/2 & 1 & 1 \end{pmatrix}.$$

We conclude this section remarking that Tables 4, 6 and 7 refer to Algorithm 3.1 and to performance of AL-RMTR. They give information on: the iteration number k, the constraint violation $\|h(x_k)\|_\infty$ and the criticality measure τ_k given in (12) or (13) at at the solution computed at Step 1 of Algorithm 3.1, the equivalent number of finest evaluations $\mathcal{L}fe_k$ performed by RMTR at the kth iteration and the total elapsed CPU time (in seconds).

Experiment set 1

In this section we address the solution of MSP and we choose to convert the inequalities into equalities by using `R-max`[2].

In Algorithm 3.1 we set $\eta^* = \omega^* = 10^{-4}$ and the following initial parameters $\eta_0 = 0.79, \omega_0 = 10^{-1}, \mu_0 = 10, \lambda_0 = 0, x_0 = 0$. The gradient of the Augmented Lagrangian function is approximated by finite differences and its Hessian is approximated by a 7-diagonal finite difference matrix, see [10].

In Table 4 we report the results obtained with Algorithm 3.1 and the three variants of RMTR: AL-RMTR-MF, AL-RMTR-FM and AL-RMTR-AF. The three procedures all converge to the solution and the CPU time is favorable to AL-RMTR-FM which exploits the multilevel structure in both variable and constraint space.

We consider now the optimization solvers GCM, CONLIN and SQP available in BOSS quattro. Since BOSS quattro requires the definition of lower and upper bound in the model, we set $l = -10, u = 10$ so that they should not be involved in the minimization process. With this setting, GCM and CONLIN failed and the BOSS-SQP declares convergence to a solution in 18 iterations starting from $x_0 = 0$ and in 45 iterations from $x_0 = 1$. In Table 5, we summarize the results obtained by BOSS-SQP: f^* and $\|h^*\|_\infty$ denote the objective function value and the constraint violation attained at the computed solution and $\#h_v^*$ is the number of violated constraints. Taking into account that the value of f^* computed at the solution given by AL-RMTR-MF is $f^* = 1.08$, BOSS-SQP seems to find good value for the objective function but the number of violated constraints is high and the constraint violation as well. This fact indicates that the computed solution is nonoptimal and this is evident in Figures 17 and 18 where we plot the solution x^* computed by AL-RMTR-MF and BOSS-SQP with $x_0 = 1$ and their curvature $\chi(x^*)$. Clearly, the multilevel procedures find a better solution than BOSS-SQP.

As an example, we now focus on the plot of the curvature of the solution x^* computed by AL-RMTR-MF. The constraint violation is tiny ($\|h^*\|_\infty = 6.9E-05$) indicating that the equality constraints $h(x) = 0$ are well satisfied. On the contrary, if we measure the violation in terms of inequalities, we have $\|c^*\|_\infty = 8.3E-03$ and $|\chi^* - \bar{\chi}| = 2.4$, that is very large, see Figure 18. This is due to the use of `R-max`^{2} which amplifies the discrepancy between the equality and non scaled inequality violation.

Experiment set 2

The aim of the Experiment set 2 is to reduce the inequality violation observed in the Experiment set 1 due to the use of `R-max`^{2}. Here we consider `R-slack` as an alternative to `R-max`^{2} to transform inequalities into equalities and we compare the two reformulations by solving vBRATU with AL-RMTR.

In Algorithm 3.1 we set $\eta^* = \omega^* = 10^{-4}$ for `R-max`^{2} and $\eta^* = 10^{-4}, \omega^* = 10^{-3}$ for `R-slack`. Moreover we set: $\eta_0 = 0.79, \omega_0 = 10^{-1}, \mu_0 = 100, \lambda_0 = 0, x_0 = 0$ and $s_0 = 0$ in `R-slack`. The gradient of the Augmented Lagrangian function is approximated by finite differences and its Hessian is approximated by a 13-diagonal finite difference matrix for `R-max`^{2} and for `R-slack` the Hessian has the following block structure [10]

$$\begin{pmatrix} \text{13-diagonal} & \text{5-diagonal} \\ \text{5-diagonal} & \text{diagonal} \end{pmatrix}.$$

In Tables 6 and 7 we report the results obtained with `R-max`^{2} and `R-slack`, respectively. Both versions succeed in finding a solution and the MF scheme is the most efficient. The CPU time is in favor of AL-RMTR with `R-max`^{2} and this may be ascribed to the fact that the problem slack structure is not fully exploited in the implementation of AL-RMTR-MF with `R-slack`. Both plots of the solutions computed by the two versions of AL-RMTR-MF look similar to the ones in Figures 9 and 10 and are not reported. On the other hand, in Figure 19 we plot the corresponding curvatures. From this figure, we notice that the inequality violation is lower if

`R-slack` is employed. In fact, despite the value of the equality violation is smaller in both cases, we have $|\chi^* - \bar{\chi}| \approx 10^{-1}$ and $|\chi^* - \bar{\chi}| \approx 10^{-6}$ for `R-max`^{2} and `R-slack`, respectively.

Experiment set 3

The model RFUSE is handled by COMBOX, a software application developed at LMS Samtech. From now on, we restrict the description of the functionalities of COMBOX and BOSS quattro to what concerns our testing experience.

The definition of RFUSE is made in COMBOX in an interactive way and we followed the description of the model made at the beginning of this section. For example, we specified the order of the DVs and RFs in Figure 12 and 14 and selected three RFs per CP. Function values and derivatives of problem (1) are provided by COMBOX.

Concerning the solution of the model, a set of optimization solvers are available (e.g. the mentioned SQP, GCM, CONLIN) and the optimization process is performed by using BOSS quattro within COMBOX. Therefore, in order to apply AL-RMTR to RFUSE, it is crucial to embed it into BOSS quattro. In practice this means to create an interface (a Fortran file called Bossol) that allows BOSS quattro and AL-RMTR to interact by reading and writing files and, at the same time, the realization of a reverse communication interface in the RMTR package. The integration of AL-RMTR in COMBOX allowed to select the code among the optimization solvers included in COMBOX by choosing the BC_LARGO algorithm, see Figure 16.

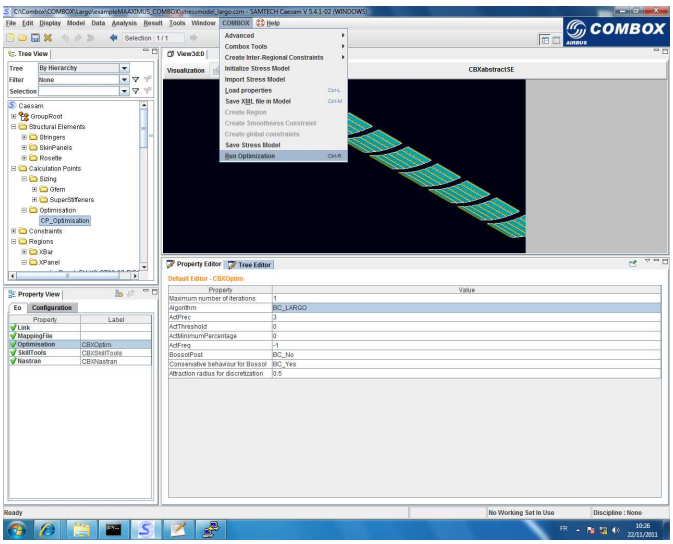

Figure 16: COMBOX graphical interface: BC_LARGO is referred to AL-RMTR-MF.

The RFUSE model data (number of panels and stringers, panel area, stringer length, bounds, initial DV value) are provided by COMBOX and they are collected in a file read at the beginning of AL-RMTR. Then, the multilevel structure of RFUSE is built and the iteration process proceed by writing file with the current DVs and waiting for the corresponding function value (objective, constraints and derivatives) that has to be supplied by BOSS quattro through another file. On its side, BOSS quattro waits for file with the DV value and generates a file with the function values. This way, the optimization process is handled by AL-RMTR (stopping criteria, information printout) and BOSS quattro serves only as a function value supplier. It is important to remark

that BOSS quattro is able to compute function values only at the finest level. Therefore, we had to use the MF scheme in AL-RMTR that does not require coarser function value.

Among the solvers available in BOSS quattro we chose to use GCM as a term of comparison in the solution of RFUSE. BOSS-GCM was run with default parameters and declared convergence to a solution x^*_{gcm} such that $f^*_{gcm} = 27.55$ in 5 iterations. Moreover, at such point the number of violated constraints is 18 but the constraint violation is negligible and, interestingly, 64 components are active, that is equal to the lower or to the upper bound.

Concerning AL-RMTR-MF, in Algorithm 3.1 we set $\eta_0 = 0.79, \omega_0 = 10^{-1}, \mu_0 = 1, \lambda_0 = 0$ and x_0, l, u were supplied by COMBOX.

Consider first the `R-max`^{2} formulation. After the first outer iteration, the procedure computes a point that lies on the lower bound of the problem. At this point, the optimality measure is exactly zero but the inequality constraint violation is quite large (of the order of 10^{-1}). In the following iterations, this violation only slightly decreases and the optimality measure gets worse. Therefore, taking into account the experience developed in the Experiment set 2, we considered `R-slack` in order to get a solution with a smaller inequality violation. In this case, we set the initial slack variables as $s_0 = s_0^f$ with $s_0^f = RF(x_0) - 1$ so that $h(x_0, s_0) = 0$. Unfortunately, the iteration process stagnates at a point where the criticality measure is quite large (of the order of 10^{-1}) as well as the inequality violation. Now, the point does not lie on the boundary of the box since only 48 components are active. Among other attempts, we changed the starting guess setting $x_0 = x^*_{gcm}, s_0 = RF(x^*_{gcm}) - 1$. With this initial choice, the process computes a different point with a lower value of f^* but that it is still not accurate enough. Moreover, using `R-slack`, no V-cycles were performed and, more seriously, we observed a dubious relationship between the step size and the ratio between the decrease in the function value and the decrease in the model value. This seems to indicate an issue in the calculation of the derivatives that deserve more investigation.

All the results are collected in Table 8 where $\#RF^*_v$ is the number of violated inequality constraints, $\max RF^*_v$ the maximum constraint violation and $\#a^*$ the number of active components of x^*. Moreover, in Figures 20 and 21 we plot the value of the DVs (panel thickness and stringer section area) computed by BOSS-GCM and AL-RMTR-MF with `R-slack` and $s_0 = s_f$, respectively.

5 Achievements and conclusions

The first phase of LARGO was dedicated to the study of the aeronautical optimization problem addressed by LMS Samtech. In particular, we performed a systematic exploration of the geometry of the constraint set that provided useful information in the understanding of the problem and highlighted properties of the functions and their derivatives which were unknown. The analysis of these properties constitutes the subject of further research at LMS Samtech.

In the second phase, we studied the structural sizing problem in terms of its multilevel structure and we proposed a new multilevel approach that can take advantage of this structure. Moreover, we made significant modification in the available RMTR package [9] to make it suitable to be employed within the industrial software (reverse communication interface).

The last phase, was focused on the implementation of the new algorithm AL-RMTR, its integration with the platforms available at LMS Samtech and its validation on test problems. The new code widens the applicability of the RMTR package since it can be used in the solution of generally constrained nonlinear programming problems. Moreover, a big effort was made in this phase to link the AL-RMTR code with COMBOX and BOSS quattro and this involved the optimization team at LMS Samtech which gave its technical support.

Concerning the experiments, first we tuned the algorithm on problems available in literature which were modified in order to mimic the behavior of the aeronautical problem under study. Second, we compared AL-RMTR with the optimization solvers available in BOSS quattro and AL-RMTR resulted more robust and computed more accurate solutions. After this preliminary experiments, we considered a small-scale industrial test case supplied by LMS Samtech and we gave its first multilevel description in the DV and constraint space. The results of the experiments on this model were not very satisfying for AL-RMTR and we believe that further work has to be made. In particular, the implementation of AL-RMTR should be improved to take into account the problem structure as well as the interface with COMBOX which provides the model data.

The initial main objective of LARGO was to develop a new approach for large-scale optimization problems (for a full fuselage we expect to deal with around 10^4 DVs and 10^6 constraints). We firmly think that the multilevel approach is the right way to reach this goal and that the work performed within LARGO allowed to compute a first step in this direction. Nonetheless, the practical use of multilevel optimization in this context deserves further research.

Acknowledgments

The authors are grateful to the optimization team at LMS Samtech for its significant contribution to the achievements of the LARGO project. In particular, the second author wish to thank Caroline Raick and Alain Remouchamps for helpful and pleasant discussions and for providing technical support.

References

[1] R.E. Bank, Ph.E. Gill and R.F. Marcia, *Interior methods for a class of elliptic variational inequalities*, in Largescale PDEconstrained optimization (Santa Fe, NM, 2001), vol. 30 of Lect. Notes Comput. Sci. Eng., Springer, Berlin, 218-235, 2003.

[2] W.L. Briggs, V.E. Henson, and S.F. McCormick, *A Multigrid Tutorial*, SIAM, Philadelphia, USA, second edition, 2000.

[3] M. Bruyneel, P. Duysinx, and C. Fleury, *A family of MMA approximations for structural optimization Structural and Multidisciplinary Optimization*, vol. 24(4), 263-276, 2002.

[4] A.R. Conn, N.I.M. Gould and Ph.L. Toint, *A globally convergent Augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numerical Analysis, vol. 28(2), 545-572, 1991.

[5] A.R. Conn, N.I.M. Gould, and Ph.L. Toint, *LANCELOT: a Fortran package for large-scale nonlinear optimization (Release A)*, Number 17 in 'Springer Series in Computational Mathematics'. Springer Verlag, Heidelberg, Berlin, New York, 1992.

[6] A.J. De Wit, *A unified approach towards decomposition and coordination for multi-level optimization*, PhD Thesis, Delft University of Technology, 2009.

[7] S. Gratton, A. Sartenaer, Ph.L. Toint, *Recursive Trust-Region Methods for Multiscale Nonlinear Optimization*, SIAM J. Optim., 19(1), 414-444, 2008b.

[8] S. Gratton, M. Mouffe, Ph.L. Toint, M. Weber-Mendonça, *A recursive trust-region method in infinity norm for bound-constrained nonlinear optimization*, IMA J. Numer. Anal., 28(4), 827-861, 2008a.

	k	$\|h(x_k)\|_\infty$	τ_k	$\mathcal{L}fe_k$	CPU
AL-RMTR-FM	1	6.88E-05	7.80E-02	47.6	199.9
	2	6.92E-05	8.43E-03	100.5	
	3	6.92E-05	9.86E-04	172.9	
	4	6.91E-05	9.11E-05	323.8	
AL-RMTR-MF	1	7.90E-05	8.28E-02	205.0	238.7
	2	7.08E-05	9.97E-03	146.0	
	3	6.94E-05	8.99E-04	144.0	
	4	6.91E-05	8.87E-05	158.0	
AL-RMTR-AF	1	8.93E-05	9.94E-02	149.0	283.3
	2	7.14E-05	9.28E-03	141.0	
	3	6.95E-05	8.49E-04	138.0	
	4	6.90E-05	9.24E-05	127.0	

Table 4: MSP: AL-RMTR results with **R - max**^{2}.

	x_0	f^*	$\|h^*\|_\infty$	$\#h_v^*$
BOSS-SQP	0	1.18	7.63E-01	108
	1	1.12	8.00E-04	81

Table 5: MSP: BOSS-SQP results with **R - max**^{2}.

[9] S. Gratton, M. Mouffe, A. Sartenaer, Ph.L. Toint, and D. Tomanos, *Numerical experience with a recursive trust-region method for multilevel nonlinear optimization*, Optim. Method Softw., 25(3), 359-386, 2010b.

[10] V. Malmedy, *Hessian approximation in multilevel nonlinear optimization*, PhD thesis, University of Namur (FUNDP), Namur, 2010.

[11] S.G. Nash, *Convergence and Descent Properties for a Class of Multilevel Optimization Algorithms*, Technical Report, Systems Engineering and Operations Research Dept., George Mason University (April 2009).

[12] J. Nocedal, S.J. Wright, *Numerical optimization*, Springer Series in Operations Research, 1999.

[13] D. Tomanos, *Algorithms and Software for Multilevel Nonlinear Optimization*, PhD thesis, University of Namur (FUNDP), Namur, 2009.

[14] J.C. Ziems, S. Ulbrich, *Adaptive Multilevel Inexact SQP Methods for PDE-Constrained Optimization*, SIAM J. Optim. 21, 1-40, 2011.

	k	$\|h(x_k)\|_\infty$	τ_k	$\mathcal{L}fe_k$	CPU
	1	1.09E-03	4.90E-03	10.3	
	2	8.42E-04	8.92E-05	36.81	
	3	7.31E-04	4.79E-05	48.7	
AL-RMTR-FM	4	1.84E-04	5.58E-05	92.3	22.7
	5	1.17E-04	4.76E-05	79.3	
	6	1.42E-05	9.10E-05	241.9	
	1	1.01E-03	3.69E-03	9.0	
	2	8.49E-04	7.39E-05	26	
AL-RMTR-MF	3	7.36E-04	8.85E-05	14.0	11.4
	4	1.84E-04	4.41E-05	37.0	
	5	1.17E-04	8.09E-05	27.0	
	6	1.42E-05	7.99E-05	143.0	
	1	1.08E-03	9.15E-03	7.0	
	2	8.37E-04	7.76E-05	27.0	
AL-RMTR-AF	3	7.28E-04	7.79E-05	23.0	29.1
	4	1.84E-04	4.70E-05	31.0	
	5	1.17E-04	8.94E-05	22.0	
	6	1.42E-05	6.97E-05	129.0	

Table 6: vBRATU: AL-RMTR results with **R - max**^{2}.

	k	$\|h(x_k)\|_\infty$	τ_k	$\mathcal{L}fe_k$	CPU
	1	2.43E-02	7.29E-03	32.4	
	2	1.79E-02	9.00E-04	63.25	
AL-RMTR-FM	3	1.43E-02	8.42E-04	47.3	421.34
	4	1.18E-03	9.85E-03	2108.7	
	5	1.83E-04	8.77E-04	5450.8	
	6	1.58E-05	9.98E-04	5557.0	
	1	2.39E-02	8.79E-03	44.0	
	2	1.80E-02	7.94E-04	32	
AL-RMTR-MF	3	1.44E-02	7.12E-04	10.0	70.2
	4	1.32E-03	9.95E-03	649.0	
	5	4.79E-05	9.98E-04	2260.0	
	1	2.38E-02	8.24E-03	41.0	
	2	1.83E-02	8.43E-04	19	
AL-RMTR-AF	3	1.45E-02	4.20E-04	19.0	174.16
	4	1.46E-03	8.09E-03	46.0	
	5	1.19E-04	8.75E-04	4490.0	
	6	3.23E-05	8.32E-04	28.0	

Table 7: vBRATU: AL-RMTR results with **R - slack**.

	f^*	$\|h^*\|_\infty$	$\#RF_v^*$	$\max RF_v^*$	τ^*	$\#a^*$
BOSS-GCM	27.55		18	2.3E-05		64
AL-RMTR-MF `R - max`^{2}	25.25	6.5E-02	49	3.6E-01	0.0E+00	104
AL-RMTR-MF `R - slack`, $s_0 = s_f$	39.41	3.4E-01	26	3.3E-01	8.7E-02	48
AL-RMTR- MF `R - slack` $x_0 = x^*_{gcm}, s_0 = RF(x^*_{gcm}) - 1$	26.32	3.3E-01	40	3.3E-01	8.5E-02	64

Table 8: Results for RFUSE.

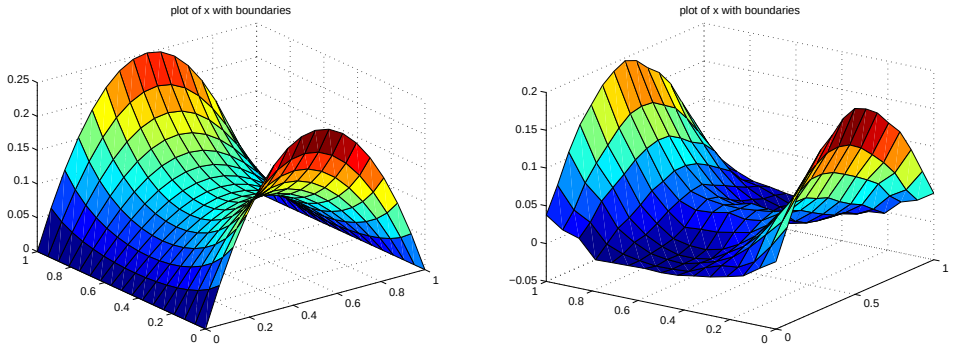

Figure 17: Solution of MPS computed by AL-RMTR-MF (left) and BOSS-SQP (right).

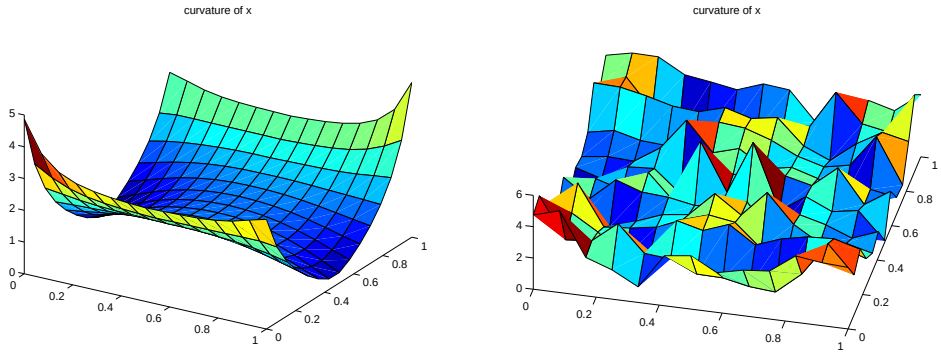

Figure 18: Curvature of the solution of MPS computed by AL-RMTR-MF (left) and BOSS-SQP (right).

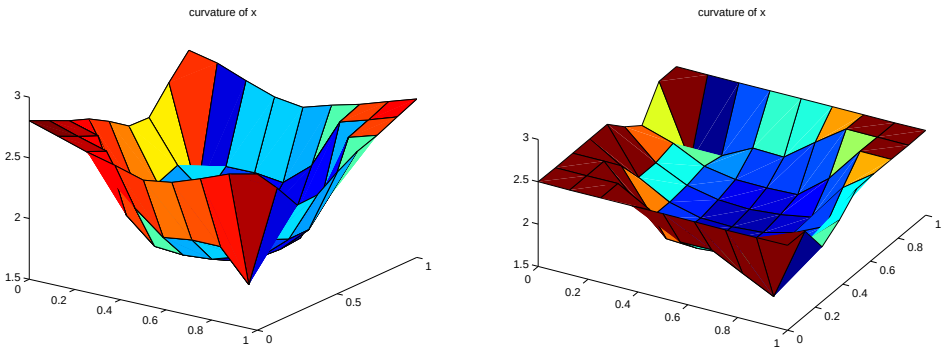

Figure 19: Curvature of the solution of vBRATU computed by AL-RMTR-MF with `R-max`^{2} (left) and with `R-slack` (right).

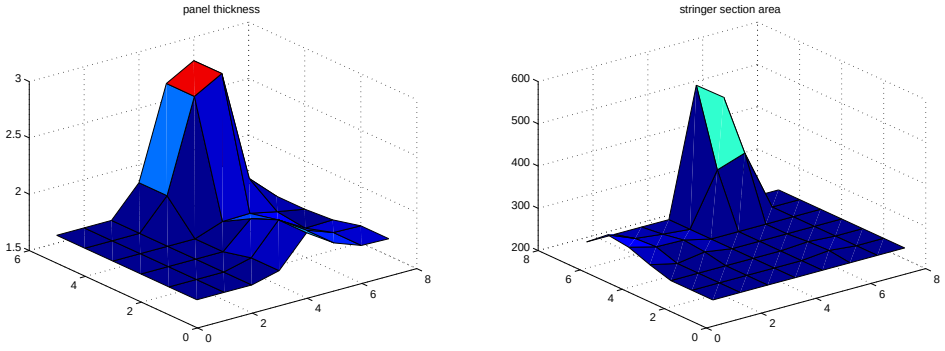

Figure 20: DVs of RFUSE computed by BOSS-GCM.

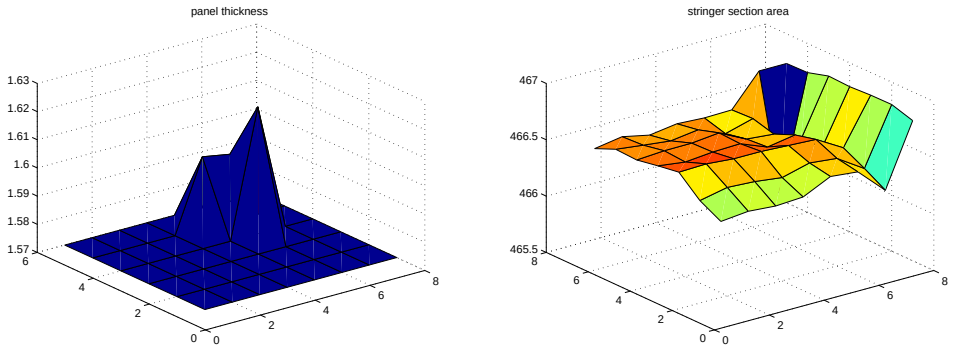

Figure 21: DVs of RFUSE computed AL-RMTR-MF with `R-slack` and $s_0 = s_f$.