# MUTIPLE-GRADIENT DESCENT ALGORITHM FOR MULTIOBJECTIVE OPTIMIZATION

Jean-Antoine Désidéri

European Congress on Computational Methods
in Applied Sciences and Engineering (ECCOMAS 2012)
J. Eberhardsteiner et.al. (eds.)
Vienna, Austria, September 10-14, 2012

# MUTIPLE-GRADIENT DESCENT ALGORITHM FOR MULTIOBJECTIVE OPTIMIZATION

**Jean-Antoine Désidéri**[1]

[1]INRIA Centre de Sophia Antipolis Mediterranee
2004 Route des Lucioles, BP 93, F-06902 Sophia Antipolis Cedex (France)
e-mail: Jean-Antoine.Desideri@inria.fr

**Keywords:** multiobjective optimization, multidisciplinary, cooperative algorithms, Pareto optimality

**Abstract.** *The steepest-descent method is a well-known and effective* single-objective descent algorithm *when the gradient of the objective function is known. Here, we propose a particular generalization of this method to multi-objective optimization by considering the concurrent minimization of $n$ smooth criteria $\{J_i\}$ ($i = 1, \ldots, n$). The novel algorithm is based on the following observation: consider a finite set of vectors $\{u_i\}$ ($u_i \in \mathbb{R}^N$, $n \leq N$); in the convex hull of this family, there exists a unique element of minimal norm, say $\omega \in \mathbb{R}^N$; then, the scalar product of $\omega$ with any vector in the convex hull, and in particular, with any $u_i$, is at least equal to $\|\omega\|^2 \geq 0$. Applying this to the objective-function gradients ($u_i = \nabla J_i$), we conclude that either $\omega = 0$, and the current design point belongs to the Pareto set, or $-\omega$ is a descent direction common to all objective functions. We propose to construct a fixed-point iteration in which updates of the element $\omega$ are used as successive directions of search. This method converges to a point on the Pareto set. This result applies to both finite-dimensional and functional design spaces. Numerical illustrations have been provided in both cases using either analytical objective functions, or (discretized) functionals in [9] [5]. Here, following [6], a domain-decomposition framework is used to illustrate the necessity, in a (discretized) functional setting, to scale the gradients appropriately.*

## 1 INTRODUCTION

Classically, in multi-objective optimization, several fundamental concepts are introduced: *dominance in efficiency* between design points, *Pareto set*, made of non-dominated solutions in design space, and *Pareto front*, its image in function space [8]. The Pareto front provides the designer with the system maximum attainable performance. For complex systems, in particular those governed by partial-differential equations, a computational challenge is to devise algorithms permitting to identify numerically the Pareto set, or the most useful portions of it. In this respect, certain evolutionary strategies have been adapted to achieve this goal, and appear to provide the most robust algorithms. *NSGA-II* [2] is certainly one of the most widely-used methods for this purpose.

In the context of differentiable optimization, one would expect adequate strategies based on gradient evaluations to also be capable of capturing Pareto fronts, with less generality or robustness, but often far greater cost efficiency. However classical techniques, such as minimizing an agglomerated criterion, or one criterion at a time under the constraints of the others, are limited by hypotheses on the pattern of the Pareto front w.r.t. convexity and, or continuity. The *Multiple-Gradient Descent Algorithm (*MGDA *)*, originally introduced in [3], and again formalized in [5], is based on a very general principle permitting to define at each iteration, a *descent direction common to all criteria*. This direction is the support of the minimum-norm element in the convex hull of the local gradients. The efficacy of the algorithm to identify the Pareto front has been demonstrated in [9] [5] in a test-case in which the Pareto front was non-convex. The method was compared in cost efficiency with an evolutionary strategy, and was found to offer very promising performance.

More recently, a variant, *MGDA-II* , has been proposed in which the descent direction is calculated by a direct procedure [4], which provides a valuable simplification of implementation.

Here, *MGDA* is tested in the fairly different context of a simulation by domain partitioning, as a technique to match the different interface components concurrently. For this, the very simple test-case of the finite-difference discretization of the Dirichlet problem over a square is considered. Full details have been provided in [6]. The study aims at assessing the performance of *MGDA* in a discretized functional setting. One of the main teachings is the necessity, here found imperative, to scale the gradients appropriately.

## 2 DIRICHLET PROBLEM, DOMAIN PARTITIONING AND MATCHING DEFECTS

We consider the model problem consisting in solving Laplace's equation,

$$-\Delta \mathsf{u} = f \qquad (\Omega) \tag{1}$$

over the square

$$\Omega = [-1,1] \times [-1,1] \tag{2}$$

subject to homogeneous boundary conditions:

$$\mathsf{u} = 0 \qquad (\Gamma = \partial\Omega) \tag{3}$$

For this, the domain $\Omega$ is partitioned in four disjoint sub-domains:

$$\begin{cases} \Omega_1 = [0,1] \times [0,1] \\ \Omega_2 = [-1,0] \times [0,1] \\ \Omega_3 = [-1,0] \times [-1,0] \\ \Omega_4 = [0,1] \times [-1,0] \end{cases} \tag{4}$$

with the following interfaces supported by the coordinate axes:

$$\begin{cases} \gamma_1 = \{\, 0 \le x \le 1\,;\; y = 0\,\} \\ \gamma_2 = \{\, x = 0\,;\; 0 \le y \le 1\} \\ \gamma_3 = \{\, -1 \le x \le 0\,;\; y = 0\,\} \\ \gamma_4 = \{\, x = 0\,;\; -1 \le y \le 0\,\} \end{cases} \tag{5}$$

along which the following Dirichlet controls are applied:

$$\begin{cases} \gamma_1 \;:\; \mathsf{u} = v_1(x) \\ \gamma_2 \;:\; \mathsf{u} = v_2(y) \\ \gamma_3 \;:\; \mathsf{u} = v_3(x) \\ \gamma_4 \;:\; \mathsf{u} = v_4(y) \end{cases} \tag{6}$$

(see Fig. 1).

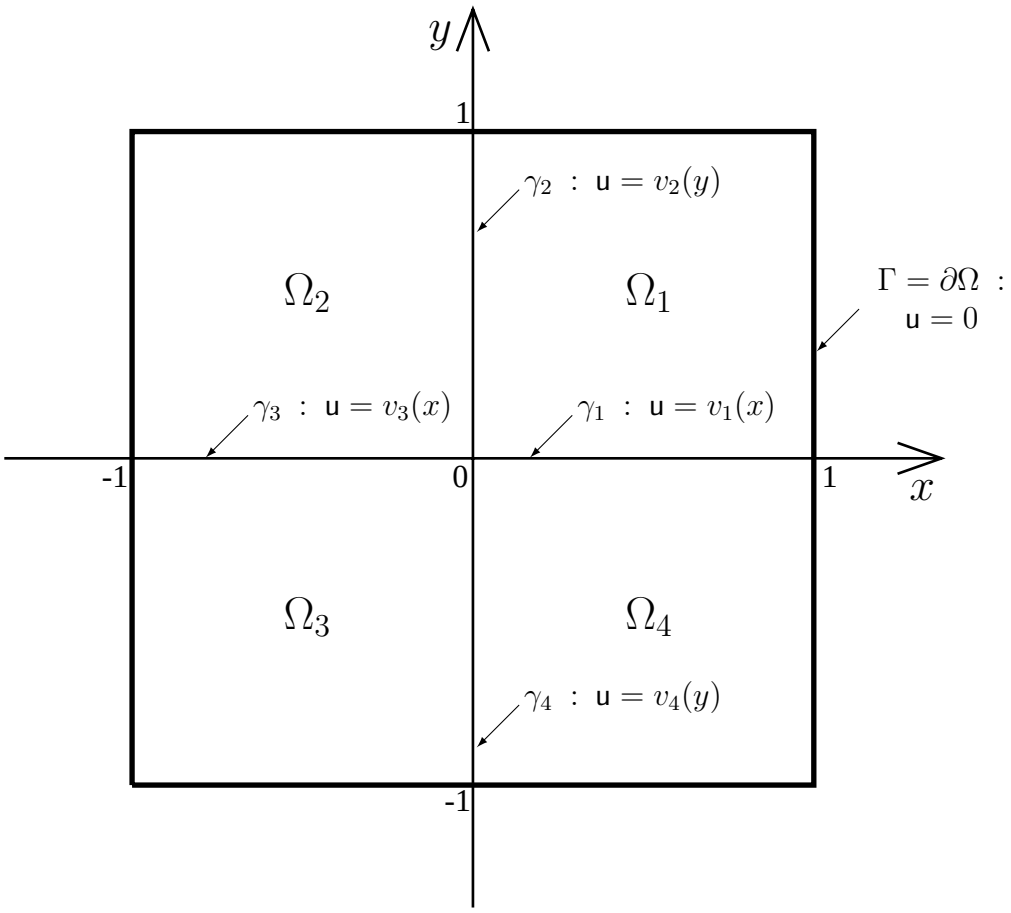


Figure 1: Partition of a square in sub-domains $\{\Omega_i\}_{(i=1,\ldots,4)}$ to solve the Dirichlet problem

A first calculation of the compound solution

$$\mathsf{u} = (\mathsf{u}_1, \mathsf{u}_2, \mathsf{u}_3, \mathsf{u}_4)^t \tag{7}$$

(where the superscript $^t$ stands for transposition) is made based on a certain setting of the interface controls:

$$v = (v_1, v_2, v_3, v_4)^t \tag{8}$$

by solving, possibly in parallel, the partial problems for $i = 1, ..., 4$:

$$\begin{cases} -\Delta \mathsf{u}_i = f & (\Omega_i) \\ \mathsf{u}_i = 0 & (\Gamma \cap \partial\Omega_i) \\ \mathsf{u}_i = v_i & (\gamma_i) \\ \mathsf{u}_i = v_{i+1} & (\gamma_{i+1}) \end{cases} \tag{9}$$

In the above, and all throughout thereafter, by periodicity, the index $i$ is understood modulo 4, i.e. $\mathsf{u}_5 = \mathsf{u}_1$, $\gamma_5 = \gamma_1$, etc.

Since the interface controls are of Dirichlet type, the resulting compound solution $\mathsf{u}$ is continuous, and its derivative along each interface is also continuous. However, in general, unless the specified controls $v_i$'s are equal to the restrictions of the global solution, the normal derivatives exhibit jump discontinuities, $s_i$'s. Here, each interface is supported by a coordinate axis, and we adopt the following sign convention: *on the interface $\gamma_i$ which is supported by the $x$ (resp. $y$) axis for $i = 1$ and 3 (resp. 2 and 4), the jump, $s_i(x)$ (resp. $s_i(y)$), is defined as the increment of the partial derivative $\partial\mathsf{u}/\partial y$ (resp. $\partial\mathsf{u}/\partial x$) as $y$ (resp. $x$) goes from $0^-$ to $0^+$.* Thus:

- over $\gamma_1$ ($0 \le x \le 1\,;\ y = 0$): $s_1(x) = \dfrac{\partial \mathsf{u}}{\partial y}(x, 0^+) - \dfrac{\partial \mathsf{u}}{\partial y}(x, 0^-) = \left[\dfrac{\partial \mathsf{u}_1}{\partial y} - \dfrac{\partial \mathsf{u}_4}{\partial y}\right](x, 0)$;
- over $\gamma_2$ ($x = 0\,;\ 0 \le y \le 1$): $s_2(y) = \dfrac{\partial \mathsf{u}}{\partial x}(0^+, y) - \dfrac{\partial \mathsf{u}}{\partial x}(0^-, y) = \left[\dfrac{\partial \mathsf{u}_1}{\partial x} - \dfrac{\partial \mathsf{u}_2}{\partial x}\right](0, y)$;
- over $\gamma_3$ ($-1 \le x \le 0\,;\ y = 0$): $s_3(x) = \dfrac{\partial \mathsf{u}}{\partial y}(x, 0^+) - \dfrac{\partial \mathsf{u}}{\partial y}(x, 0^-) = \left[\dfrac{\partial \mathsf{u}_2}{\partial y} - \dfrac{\partial \mathsf{u}_3}{\partial y}\right](x, 0)$;
- over $\gamma_4$ ($x = 0\,;\ -1 \le y \le 0$): $s_4(y) = \dfrac{\partial \mathsf{u}}{\partial x}(0^+, y) - \dfrac{\partial \mathsf{u}}{\partial x}(0^-, y) = \left[\dfrac{\partial \mathsf{u}_4}{\partial x} - \dfrac{\partial \mathsf{u}_3}{\partial x}\right](0, y)$.

The above local measures of the defect in matching conditions can be associated with global measures defined as separate functionals defined over the interfaces:

$$J_i = \int_{\gamma_i} \tfrac{1}{2} s_i^2 \, w \, d\gamma_i \tag{10}$$

that is, explicitly:

$$J_1 = \int_0^1 \tfrac{1}{2} s_1(x)^2 \, w(x) \, dx \qquad J_2 = \int_0^1 \tfrac{1}{2} s_2(y)^2 \, w(y) \, dy \tag{11}$$

$$J_3 = \int_{-1}^0 \tfrac{1}{2} s_3(x)^2 \, w(x) \, dx \qquad J_4 = \int_{-1}^0 \tfrac{1}{2} s_4(y)^2 \, w(y) \, dy \tag{12}$$

Here, $w(t)$ ($t \in [0, 1]$) is an optional weighting function, and $w(-t) = w(t)$.

The jump $s_i$ depends on the partial solutions $\mathsf{u}_{i-1}$ and $\mathsf{u}_i$, which themselves, depend on $(v_{i-1}, v_i)$ and $(v_i, v_{i+1})$ respectively. Hence, the integral $J_i$ depends on all four sub-controls except $v_{i+2}$. Nevertheless, these four integrals are thereafter considered as functionals of $v$.

The coordination problem is to devise a convergent iteration on the control $v$ to satisfy in the limit the matching conditions

$$J_1 = J_2 = J_3 = J_4 = 0 \tag{13}$$

To achieve this, the functional gradients are firstly established using the classical adjoint-equation approach, and several strategies are proposed and tested numerically.

## 3 ADJOINT PROBLEMS AND FUNCTIONAL GRADIENTS

A first calculation is made based on the four-component control $v = (v_1, v_2, v_3, v_4)^t$, resulting in the compound solution $\mathsf{u} = (\mathsf{u}_1, \mathsf{u}_2, \mathsf{u}_3, \mathsf{u}_4)^t$, and the multi-component criterion $J = (J_1, J_2, J_3, J_4)^t$.

Then, one perturbs the control $v$ of

$$v' = (v'_1, v'_2, v'_3, v'_4)^t = \Big(\delta v_1(x), \delta v_2(y), \delta v_3(x), \delta v_4(y)\Big)^t \tag{14}$$

Consequently, the compound solution $u$ is perturbed of

$$\mathsf{u}' = (\mathsf{u}'_1, \mathsf{u}'_2, \mathsf{u}'_3, \mathsf{u}'_4)^t = \Big(\delta \mathsf{u}_1(x,y), \delta \mathsf{u}_2(x,y), \delta \mathsf{u}_3(x,y), \delta \mathsf{u}_4(x,y)\Big)^t \tag{15}$$

in which $\mathsf{u}'_i$ is the solution of the following linearized system posed on sub-domain $\Omega_i$:

$$\begin{cases} \Delta \mathsf{u}'_i = 0 & (\Omega_i) \\ \mathsf{u}'_i = 0 & (\Gamma \cap \partial\Omega_i) \\ \mathsf{u}'_i = v'_i & (\gamma_i) \\ \mathsf{u}'_i = v'_{i+1} & (\gamma_{i+1}) \end{cases} \tag{16}$$

These state perturbations induce the following functional perturbations:

$$J'_i = \delta J_i = \int_{\gamma_i} s_i s'_i \, w \, d\gamma_i \tag{17}$$

in which $s'_i = \delta s_i$. But:

$$\begin{cases} s_1 s'_1 = \left[\dfrac{\partial \mathsf{u}_1}{\partial y} - \dfrac{\partial \mathsf{u}_4}{\partial y}\right]\left[\dfrac{\partial \mathsf{u}'_1}{\partial y} - \dfrac{\partial \mathsf{u}'_4}{\partial y}\right](x,0) \\ s_2 s'_2 = \left[\dfrac{\partial \mathsf{u}_1}{\partial x} - \dfrac{\partial \mathsf{u}_2}{\partial x}\right]\left[\dfrac{\partial \mathsf{u}'_1}{\partial x} - \dfrac{\partial \mathsf{u}'_2}{\partial x}\right](0,y) \\ s_3 s'_3 = \left[\dfrac{\partial \mathsf{u}_2}{\partial y} - \dfrac{\partial \mathsf{u}_3}{\partial y}\right]\left[\dfrac{\partial \mathsf{u}'_2}{\partial y} - \dfrac{\partial \mathsf{u}'_3}{\partial y}\right](x,0) \\ s_4 s'_4 = \left[\dfrac{\partial \mathsf{u}_4}{\partial x} - \dfrac{\partial \mathsf{u}_3}{\partial x}\right]\left[\dfrac{\partial \mathsf{u}'_4}{\partial x} - \dfrac{\partial \mathsf{u}'_3}{\partial x}\right](0,y) \end{cases} \tag{18}$$

We now recall Green's formula for two functions $\phi$ and $\psi \in H^2(\varpi)$, for a simply-connected planar domain $\varpi$ with smooth enough boundary. Since

$$\iint_{\varpi} \phi \, \Delta\psi = \iint_{\varpi} \phi \, \nabla.(\nabla\psi) = \int_{\partial\varpi} \phi \, \psi_n - \iint_{\varpi} \nabla\phi.\nabla\psi \tag{19}$$

where $\vec{n}$ is outward unit vector normal to the boundary $\partial\varpi$, and $\psi_n = \partial\psi/\partial n$ the normal derivative, and symmetrically,

$$\iint_{\varpi} \psi\,\Delta\phi = \iint_{\varpi} \psi\,\nabla.(\nabla\phi) = \int_{\partial\varpi} \psi\,\phi_n - \iint_{\varpi} \nabla\psi.\nabla\phi \tag{20}$$

where $\phi_n = \partial\phi/\partial n$ is the normal derivative of $\phi$, one has:

$$\iint_{\varpi} (\phi\,\Delta\psi - \psi\,\Delta\phi) = \int_{\partial\varpi} (\phi\,\psi_n - \psi\,\phi_n) \tag{21}$$

Consider the following eight adjoint systems (two per sub-domain):

$$\begin{cases} \Delta p_i = 0 & (\Omega_i) \\ p_i = 0 & (\partial\Omega_i\backslash\gamma_i) \\ p_i = s_i w & (\gamma_i) \end{cases} \qquad \begin{cases} \Delta q_i = 0 & (\Omega_i) \\ q_i = 0 & (\partial\Omega_i\backslash\gamma_{i+1}) \\ q_i = s_{i+1} w & (\gamma_{i+1}) \end{cases} \tag{22}$$

Then apply Green's formula, (21), to the eight cases corresponding to

$$\varpi = \Omega_i\ (i = 1, 2, 3, 4), \quad \phi = p_i \text{ or } q_i, \quad \psi = \mathsf{u}_i' \tag{23}$$

so that $\Delta\phi = \Delta\psi = 0$ in $\varpi$ giving:

$$\int_{\partial\Omega_i} \phi\,\psi_n = \int_{\partial\Omega_i} \psi\,\phi_n \tag{24}$$

On the boundary $\partial\varpi = \partial\Omega_i$:

- $\phi = 0$ except for $\phi = p_i = s_i w$ along $\gamma_i$, and $\phi = q_i = s_{i+1} w$ along $\gamma_{i+1}$;
- $\psi = \mathsf{u}_i' = \begin{cases} v_i' & \text{along } \gamma_i \\ v_{i+1}' & \text{along } \gamma_{i+1} \\ 0 & \text{along } \Gamma \cap \Omega_i \end{cases}$

Hence (24) reduces to:

$$\int_{\gamma_i} s_i\,\mathsf{u}_{in}' w = \int_{\gamma_i} p_{in}\,v_i' + \int_{\gamma_{i+1}} p_{in}\,v_{i+1}' \tag{25}$$

for $\phi = p_i$, and to:

$$\int_{\gamma_{i+1}} s_{i+1}\,\mathsf{u}_{in}' w = \int_{\gamma_i} q_{in}\,v_i' + \int_{\gamma_{i+1}} q_{in}\,v_{i+1}' \tag{26}$$

for $\phi = q_i$.

These two equations are particularized to sub-domains $\Omega_i$ as follows.

**Sub-domain $\Omega_1$:** on $\gamma_1$: $\vec{n} = \vec{n}_{14} = -\vec{\jmath}$; on $\gamma_2$, $\vec{n} = \vec{n}_{12} = -\vec{\imath}$. Thus (25)-(26) write:

$$\begin{cases} \displaystyle\int_0^1 s_1(x)\,\left(-\frac{\partial \mathsf{u}_1'}{\partial y}(x,0)\right) w(x)\,dx = \int_0^1 \left(-\frac{\partial p_1}{\partial y}(x,0)\right) v_1'(x)\,dx + \int_0^1 \left(-\frac{\partial p_1}{\partial x}(0,y)\right) v_2'(y)\,dy \\ \displaystyle\int_0^1 s_2(y)\,\left(-\frac{\partial \mathsf{u}_1'}{\partial x}(0,y)\right) w(y)\,dy = \int_0^1 \left(-\frac{\partial q_1}{\partial y}(x,0)\right) v_1'(x)\,dx + \int_0^1 \left(-\frac{\partial q_1}{\partial x}(0,y)\right) v_2'(y)\,dy \end{cases} \tag{27}$$

**Sub-domain $\Omega_2$:** on $\gamma_2$: $\vec{n}=\vec{n}_{21}=+\vec{\imath}$; on $\gamma_3$, $\vec{n}=\vec{n}_{23}=-\vec{\jmath}$. Thus (25)-(26) write:

$$\left\{\begin{array}{l} \displaystyle\int_0^1 s_2(y)\,\left(\frac{\partial \mathsf{u}_2'}{\partial x}(0,y)\right)\,w(y)\,dy=\int_0^1\left(\frac{\partial p_2}{\partial x}(0,y)\right)\,v_2'(y)\,dy+\int_{-1}^0\left(-\frac{\partial p_2}{\partial y}(x,0)\right)\,v_3'(x)\,dx \\ \displaystyle\int_{-1}^0 s_3(x)\,\left(-\frac{\partial \mathsf{u}_2'}{\partial y}(x,0)\right)\,w(x)\,dx=\int_0^1\left(\frac{\partial q_2}{\partial x}(0,y)\right)\,v_2'(y)\,dy+\int_{-1}^0\left(-\frac{\partial q_2}{\partial y}(x,0)\right)\,v_3'(x)\,dx \end{array}\right. \tag{28}$$

**Sub-domain $\Omega_3$:** on $\gamma_3$: $\vec{n}=\vec{n}_{32}=+\vec{\jmath}$; on $\gamma_4$, $\vec{n}=\vec{n}_{34}=+\vec{\imath}$. Thus (25)-(26) write:

$$\left\{\begin{array}{l} \displaystyle\int_{-1}^0 s_3(x)\,\left(\frac{\partial \mathsf{u}_3'}{\partial y}(x,0)\right)\,w(x)\,dx=\int_{-1}^0\left(\frac{\partial p_3}{\partial y}(x,0)\right)\,v_3'(x)\,dx+\int_{-1}^0\left(\frac{\partial p_3}{\partial x}(0,y)\right)\,v_4'(y)\,dy \\ \displaystyle\int_{-1}^0 s_4(y)\,\left(\frac{\partial \mathsf{u}_3'}{\partial x}(0,y)\right)\,w(y)\,dy=\int_{-1}^0\left(\frac{\partial q_3}{\partial y}(x,0)\right)\,v_3'(x)\,dx+\int_{-1}^0\left(\frac{\partial q_3}{\partial x}(0,y)\right)\,v_4'(y)\,dy \end{array}\right. \tag{29}$$

**Sub-domain $\Omega_4$:** on $\gamma_4$: $\vec{n}=\vec{n}_{43}=-\vec{\imath}$; on $\gamma_1$, $\vec{n}=\vec{n}_{41}=+\vec{\jmath}$. Thus (25)-(26) write:

$$\left\{\begin{array}{l} \displaystyle\int_{-1}^0 s_4(y)\,\left(-\frac{\partial \mathsf{u}_4'}{\partial x}(0,y)\right)\,w(y)\,dy=\int_{-1}^0\left(-\frac{\partial p_4}{\partial x}(0,y)\right)\,v_4'(y)\,dy+\int_0^1\left(\frac{\partial p_4}{\partial y}(x,0)\right)\,v_1'(x)\,dx \\ \displaystyle\int_0^1 s_1(x)\,\left(\frac{\partial \mathsf{u}_4'}{\partial y}(x,0)\right)\,w(x)\,dx=\int_{-1}^0\left(-\frac{\partial q_4}{\partial x}(0,y)\right)\,v_4'(y)\,dy+\int_0^1\left(\frac{\partial q_4}{\partial y}(x,0)\right)\,v_1'(x)\,dx \end{array}\right. \tag{30}$$

Then, (27)-(30) are injected in (18) and (17) to get:

$$\left\{\begin{array}{l} \displaystyle J_1'=\int_0^1 s_1(x)\,s_1'(x)\,w(x)\,dx=\int_0^1 s_1(x)\,\left[\frac{\partial \mathsf{u}_1'}{\partial y}-\frac{\partial \mathsf{u}_4'}{\partial y}\right](x,0)\,w(x)\,dx \\ \displaystyle\quad=\int_0^1\frac{\partial(p_1-q_4)}{\partial y}(x,0)\,v_1'(x)\,dx+\int_0^1\frac{\partial p_1}{\partial x}(0,y)\,v_2'(y)\,dy+\int_{-1}^0\frac{\partial q_4}{\partial x}(0,y)\,v_4'(y)\,dy \\ \displaystyle J_2'=\int_0^1 s_2(y)\,s_2'(y)\,w(y)\,dy=\int_0^1 s_2(y)\,\left[\frac{\partial \mathsf{u}_1'}{\partial x}-\frac{\partial \mathsf{u}_2'}{\partial x}\right](0,y)\,w(y)\,dy \\ \displaystyle\quad=\int_0^1\frac{\partial q_1}{\partial y}(x,0)\,v_1'(x)\,dx+\int_0^1\frac{\partial(q_1-p_2)}{\partial x}(0,y)\,v_2'(y)\,dy+\int_{-1}^0\frac{\partial p_2}{\partial y}(x,0)\,v_3'(x)\,dx \\ \displaystyle J_3'=\int_{-1}^0 s_3(x)\,s_3'(x)\,w(x)\,dx=\int_{-1}^0 s_3(x)\,\left[\frac{\partial \mathsf{u}_2'}{\partial y}-\frac{\partial \mathsf{u}_3'}{\partial y}\right](x,0)\,w(x)\,dx \\ \displaystyle\quad=-\int_0^1\frac{\partial q_2}{\partial x}(0,y)\,v_2'(y)\,dy+\int_{-1}^0\frac{\partial(q_2-p_3)}{\partial y}(x,0)\,v_3'(x)\,dx-\int_{-1}^0\frac{\partial p_3}{\partial x}(0,y)\,v_4'(y)\,dy \\ \displaystyle J_4'=\int_{-1}^0 s_4(y)\,s_4'(y)\,w(y)\,dy=\int_{-1}^0 s_4(y)\,\left[\frac{\partial \mathsf{u}_4'}{\partial x}-\frac{\partial \mathsf{u}_3'}{\partial x}\right](0,y)\,w(y)\,dy \\ \displaystyle\quad=-\int_0^1\frac{\partial p_4}{\partial y}(x,0)\,v_1'(x)\,dx-\int_{-1}^0\frac{\partial q_3}{\partial y}(x,0)\,v_3'(x)\,dx+\int_{-1}^0\frac{\partial(p_4-q_3)}{\partial x}(0,y)\,v_4'(y)\,dy \end{array}\right. \tag{31}$$

These formulas are of the form:

$$J_i' = \sum_{j=1}^{4} \int_{\gamma_j} G_{i,j}\, v_j'\, d\gamma_j \qquad (i = 1, ..., 4) \tag{32}$$

in which the kernels, $\{G_{i,j}\}$, are partial gradients given in terms of the partial derivatives of the eight adjoint states $\{p_i, q_i\}_{(i=1,...,4)}$.

## 4 DISCRETIZATION

For purpose of numerical treatment, we assume that each sub-problem is discretized by standard centered finite-differences over a uniform (sub-)mesh of dimension $N_X \times N_Y$ rectangular cells. This permits a fast direct inversion by discrete separation of variables:

$$\mathsf{u}_h = (\Omega_X \otimes \Omega_Y)\ (\Lambda_X \oplus \Lambda_Y)^{-1}\ (\Omega_X \otimes \Omega_Y)\ f_h \tag{33}$$

$\Omega_X$ and $\Omega_Y$ are respectively the $N_X \times N_X$ and $N_Y \times N_Y$ orthogonal matrices associated with the discrete sine transform. For Dirichlet boundary conditions these matrices are also symmetric. The matrices $\Lambda_X$ and $\Lambda_Y$ are diagonal matrices of the known eigenvalues of the second-order difference operators in $x$ and $y$ respectively (see [6] for details).

For each sub-domain $\Omega_i$, derivatives normal to a given interface are calculated by one-sided second-order finite differences, and tangential derivatives, by central differencing.

The matching-defect integrals are then approximated by the trapezoidal rule:

$$J_1 \doteq \frac{h_Z}{2} \sum_{j=1}^{N_X-1} s_{1,j}^2\, w_j\ ,\ J_2 \doteq \frac{h_Y}{2} \sum_{k=1}^{N_Y-1} s_{2,k}^2\, w_k\ ,\ J_3 \doteq \frac{h_X}{2} \sum_{j=1}^{N_X-1} s_{3,j}^2\, w_j\ ,\ J_4 \doteq \frac{h_Y}{2} \sum_{k=1}^{N_Y-1} s_{4,k}^2\, w_k\ . \tag{34}$$

Two adjoint problems are solved on each sub-domain $\Omega_i$, again by direct inversions, to get the functions $p_i$ and $q_i$ whose derivatives are then approximated along interfaces and injected in (31). Then [6]:

$$J_i' \doteq \sum_{j=1}^{4} \sum_{k=1}^{N_Z} G_{i,j\,.k}\, v_{j,k}'\, h_Z \qquad (i = 1, ..., 4) \tag{35}$$

The discrete gradient of the criterion $J_i$ w.r.t. the nodal values of the control $v_j$ is given by:

$$\frac{\partial J_i}{\partial v_{j\,,k}} \doteq G_{i,j\,.k}\, h_Z \tag{36}$$

For each criterion $J_i$, four such discrete gradients are calculated (one per control $v_j$), except that one of them is equal to 0. These four vectors are assembled in one, thereafter denoted $\nabla J_i$, of dimension $2(N_X + N_Y - 2)$.

Now, knowing (second-order approximations of) the criteria $\{J_i\}_{(i=1,...4)}$ and their gradients $\{\nabla J_i\}_{(i=1,...,4)}$ w.r.t. the $2(N_X + N_Y - 2)$ nodal controls, we need to set up a strategy to iterate on these controls to satisfy the matching conditions at convergence.

## 5 GRADIENT-BASED COORDINATION ITERATIONS

Our main objective is to compare the standard steepest-descent method with the *Multiple-Gradient Descent Algorithm (*MGDA*)* as potential iterative methods to satisfy the matching conditions by driving the defect functionals to 0.

### 5.1 Conventional steepest-descent method

In the conventional approach, one considers a *single matching defect measure*, treating all interfaces as one:

$$\mathsf{J} = \sum_{i=1}^{4} J_i \tag{37}$$

The discrete gradient is then simply equal to the sum of the individual contributions of the interfaces:

$$\nabla \mathsf{J} = \sum_{i=1}^{4} \nabla J_i \tag{38}$$

The above global criterion can then be driven to 0 by the classical *steepest-descent method* [1] [7]: at iteration $\ell$, the control $v$ is updated proportionally to (the opposite of) the discrete gradient:

$$v^{(\ell+1)} = v^{(\ell)} - \rho_\ell \nabla \mathsf{J}^{(\ell)} \tag{39}$$

for some appropriate positive step-size $\rho_\ell$ (see below), and a new compound solution $u^{(\ell+1)}$ is calculated, the defect-functional and its gradient reevaluated, and so on until a satisfactory convergence is achieved.

Strictly speaking, in the standard steepest-descent method, once the direction of search is identified, by the calculation of the gradient $\nabla \mathsf{J}^{(\ell)}$, the step-size $\rho_\ell$ is often defined via a one-dimensional minimization:

$$\rho_\ell = \mathsf{Argmin}_\rho \, j(\rho) \, ; \; j(\rho) := \mathsf{J}\left(v^{(\ell)} - \rho \nabla \mathsf{J}^{(\ell)}\right) \tag{40}$$

This minimization is usually carried out by a numerical procedure. However here, we know of an additional information: the targeted value of $\mathsf{J}$ is known: $\mathsf{J} = 0$. An estimation of the variation of $\mathsf{J}$ is given by the differential:

$$\delta \mathsf{J} = \nabla \mathsf{J}^{(\ell)} . \delta v^{(\ell)} = -\rho_\ell \left\| \nabla \mathsf{J}^{(\ell)} \right\|^2 \tag{41}$$

Hence, the step-size expected to diminish $\mathsf{J}^{(\ell)}$ of the amount $\delta \mathsf{J} = -\varepsilon \mathsf{J}^{(\ell)}$ is estimated to be:

$$\rho_\ell = \frac{\varepsilon \mathsf{J}^{(\ell)}}{\left\| \nabla \mathsf{J}^{(\ell)} \right\|^2} \tag{42}$$

In particular for $\varepsilon = 1$, we get the quasi-Newton method since the employed discrete gradient is only approximately equal to the gradient of the discrete $\mathsf{J}$.

### 5.2 Multiple-gradient descent algorithm (*MGDA* )

In this subsection, we propose an alternative coordination algorithm in which the matching of the sub-solutions is treated as a multi-objective optimization problem, considering that all defect-functionals $J_i$'s should be driven to 0 concurrently.

In the *Multiple Gradient Descent Algorithm MGDA* (see [3] for a detailed definition and convergence proof), once the individual discrete gradients,

$$u_i = \nabla J_i \qquad (i = 1, ..., 4) \qquad (u_i \in \mathbb{R}^N) \tag{43}$$

are known, one considers the convex hull $\overline{\mathsf{U}}$ of these vectors, and identifies its minimum-norm element $\omega$:

$$\begin{cases} \omega = \mathsf{Argmin}_{u\in\overline{\mathsf{U}}} \|u\|^2 \\ \overline{\mathsf{U}} = \left\{ u\in\mathbb{R}^N \,/\, u = \sum_{i=1}^{4} \alpha_i u_i \,;\; \alpha_i \geq 0 \;(\forall i) \,;\; \sum_{i=1}^{4} \alpha_i = 1 \right\} \end{cases} \tag{44}$$

In our problem, the dimension $N$ is the number of nodal controls:

$$N = 2(N_X + N_Y - 2) \tag{45}$$

A special parameterization of the convex hull is proposed to facilitate the determination of the element $\omega$ by a numerical optimization procedure (see Appendix A).

Then, once the element $\omega$ is determined, if $\omega = 0$[1], the current iterate is, or is treated as Pareto stationary. But here, the Pareto front is made of only one point corresponding to $J_i = 0$ for all $i$. This situation corresponds to full convergence of the coordination algorithm. Otherwise ($\omega \neq 0$), $-\omega$ is a descent direction for all criteria simultaneously. Thus, (39) is replaced by:

$$v^{(\ell+1)} = v^{(\ell)} - \rho_\ell \omega^{(\ell)} \tag{46}$$

Here again, we propose to adjust the step-size $\rho_\ell$ according to (42). However here, it is not clear *a priori* that the proper scaling corresponds to $\varepsilon \sim 1$.

## 6 NUMERICAL EXPERIMENTATION

### 6.1 Test-case

Let $a$ and $b$ be two adjustable constants, and:

$$\begin{cases} \psi = (x+a)^2 + (y+b)^2 \qquad (-1 \leq x \leq 1) \qquad (-1 \leq y \leq 1) \\ \tau = \dfrac{1}{\ln(a^2+b^2)} \\ \phi = \tau \ln \psi \\ f^{(x)} = 1 - x^2 \qquad f^{(y)} = 1 - y^2 \\ \mathsf{u}_e = f^{(x)} f^{(y)} \phi \end{cases} \tag{47}$$

As a result, $\phi$ is a harmonic function:

$$\Delta\phi = 0 \tag{48}$$

and this permits us to simplify somewhat the expression of the Laplacian of $\mathsf{u}_e$:

$$\begin{aligned} \Delta \mathsf{u}_e &= \Delta\left(f^{(x)} f^{(y)}\right)\phi + 2\nabla\left(f^{(x)} f^{(y)}\right).\nabla\phi + \left(f^{(x)} f^{(y)}\right)\Delta\phi \\ &= -2\left(f^{(x)} f^{(y)}\right)\phi + 2\nabla\left(f^{(x)} f^{(y)}\right).\nabla\phi \qquad (49) \\ &= -f_e \qquad (50) \end{aligned}$$

where

$$f_e = 2\left(f^{(x)} f^{(y)}\right)\phi + \frac{8\tau}{\psi}\left[x(x+a)f^{(y)} + y(y+b)f^{(x)}\right] \tag{51}$$

[1] In the numerical implementation, this condition is relaxed to be: $||\omega|| < TOL$, for a given tolerance $TOL$.

Hence, for $f = f_e$, the exact solution of the continuous problem is $\mathsf{u} = \mathsf{u}_e$.

The constants $a$ and $b$ have been introduced to destroy the symmetry in the solution. More specifically, the following settings were made: $a = \frac{5}{4}$ and $b = \frac{3}{4}$. The corresponding problem has been discretized and solved using either one domain to establish a reference, or four to experiment multi-criterion optimization algorithms.

The single-domain discrete solution $\mathsf{u}_h$ is depicted in Fig. 2 as a surface in 3D, and the corresponding contour map is given more precisely in Fig. 3.

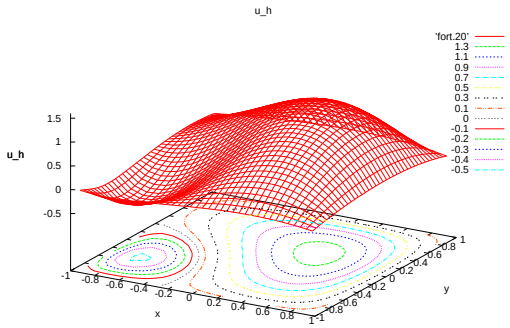


Figure 2: Single-domain discrete solution $\mathsf{u}_h$

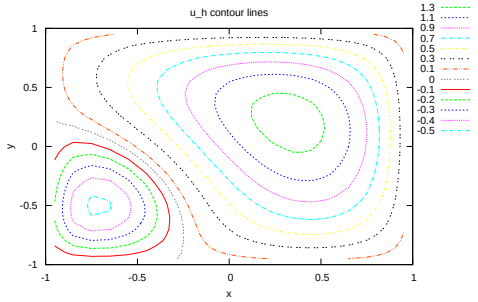


Figure 3: Single-domain discrete solution $\mathsf{u}_h$; contour map

### 6.2 Quasi-Newton steepest descent

In a first series of numerical experiments, the steepest-descent method was applied to drive to 0 the global criterion J. After a few trials, it appeared that best convergence was achieved by setting $\varepsilon$ to 1 in (42), which corresponds to the quasi-Newton method.

Two experiments are reported presently. They differ in the setting of the initial interface conditions. In the first, the $v_i$'s are initially assigned the restriction to the corresponding interface of the exact solution $\mathsf{u}_e$, which differs from the discrete solution by truncation errors. In this case, iterative errors are initially very small, which permits the asymptotic convergence to be assessed. In the second experiment, the controls are initially set to 0 in order to assess the global convergence.

**Asymptotic convergence.** The convergence history of the global criterion J as well as its individual parts, $\{J_i\}_{(i=1,\ldots,4)}$ is represented in Fig. 4. The criterion J, in 20 iterations, goes from $2.9 \times 10^{-2}$ to a level below $10^{-4}$. Note that $J_4$ is somewhat smaller in magnitude and more subject to oscillations.

**Global convergence.** The convergence history of the global criterion J as well as its individual parts, $\{J_i\}_{(i=1,\ldots,4)}$ is represented in Fig. 5. The criterion J, in 200 iterations, is reduced by 8 orders of magnitude. The different criteria, apart from small oscillations, converge at essentially the same rate. In a linear convergence process, this rate is imposed by the most persistent mode, present in all criteria when the initial condition is arbitrary.

**Convergence of the gradients.** The evolution of the four gradients $\{\partial \mathsf{J}/\partial v_i\}_{(i=1,\ldots,4)}$ over 200 iterations is given on Fig. 6-Fig. 9. They appear as high-frequency modes. Each one vanishes asymptotically over the corresponding interface.

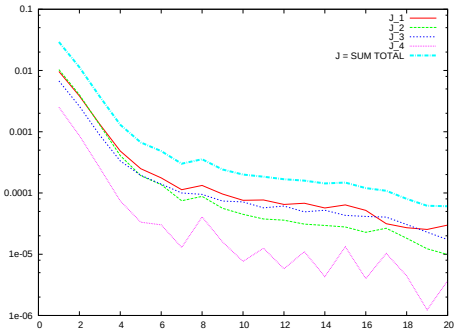


Figure 4: Quasi-Newton steepest descent - convergence history of criteria (discretized continuous solution imposed initially at interfaces)

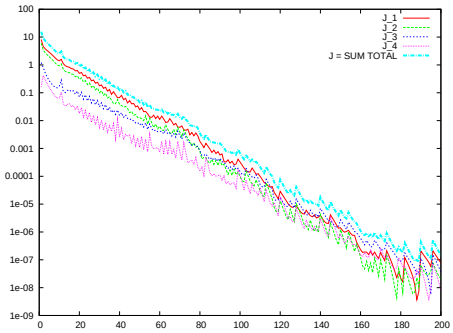


Figure 5: Quasi-Newton steepest descent - convergence history of criteria (zero initial interface conditions)

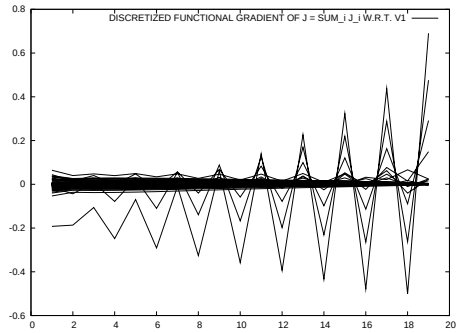


Figure 6: Quasi-Newton steepest descent - 200 iterations of $\partial \mathsf{J}/\partial v_1$

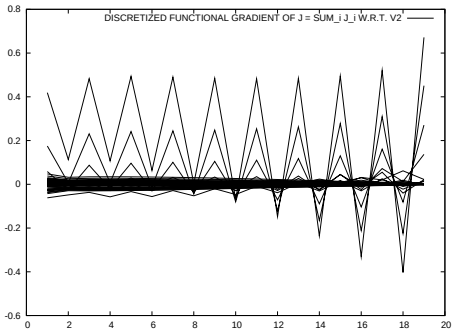


Figure 7: Quasi-Newton steepest descent - 200 iterations of $\partial \mathsf{J}/\partial v_2$

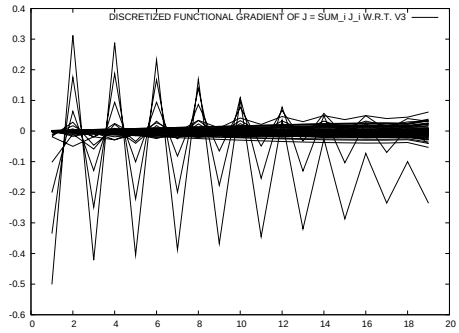


Figure 8: Quasi-Newton steepest descent - 200 iterations of $\partial \mathsf{J}/\partial v_3$

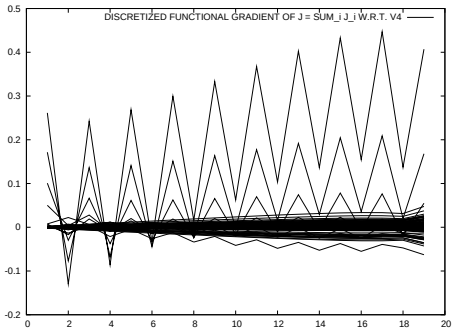


Figure 9: Quasi-Newton steepest descent - 200 iterations of $\partial \mathsf{J}/\partial v_4$

**Discrete solution.** The four-domain discrete solution is found perfectly smooth, in fact even smoother than the above single-domain discrete solution (Figs. 10 and 11). This is due to a higher degree of iterative convergence.

### 6.3 Basic *MGDA*

**Practical determination of the minimum-norm element $\omega$.** In the experiments of this section, at each iteration, the 3 parameters $c_1$, $c_2$ and $c_3$ of (76) have been discretized uniformly by step of 0.01, and $\omega$ was set equal to the vector of minimum norm among the essentially $10^6$ corresponding candidates. The resulting vector was therefore a very coarse approximation of the actual vector $\omega$.

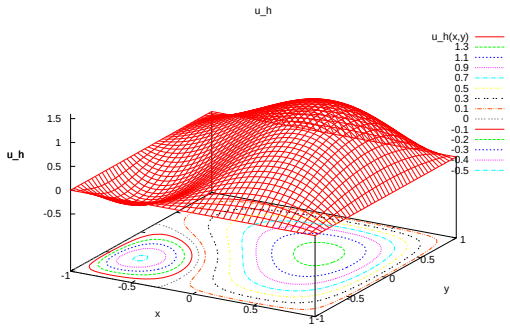


Figure 10: Four-domain discrete solution $u_h$

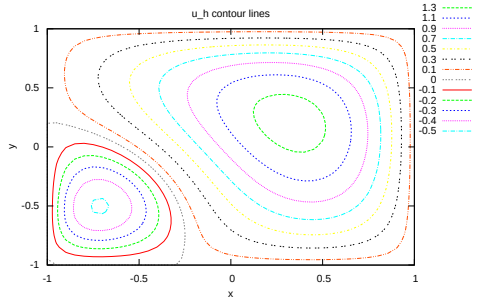


Figure 11: Four-domain discrete solution $u_h$; contour map

**Asymptotic convergence** For this experiment, the discretized continuous solution is again imposed initially at the interfaces. The convergence history of the above basic algorithm is indicated in Fig. 12 for 20 iterations. After some initial adjustment, the trend is towards decaying, but at a deceiving slow rate.

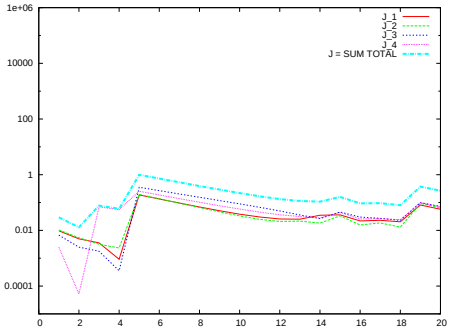


Figure 12: Basic *MGDA* - asymptotic convergence history of criteria

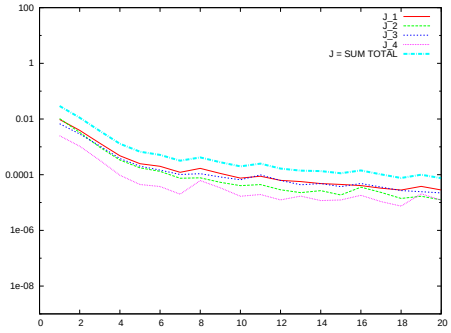


Figure 13: *MGDA* based on logarithmic gradients - asymptotic convergence history of criteria

In an attempt to explain this poor convergence, the following observation was made: suppose the gradients of the individual criteria, $\{\partial J_i/\partial v\}_{(i=1,\ldots,4)}$, are very different in magnitude. Remember that in a linear iterative process, unless initial conditions are very special, all quantities converge at the same rate, say $C\rho^{(iter)}$, where $\rho$ is the spectral radius, and $C$ a constant which depends on the quantity considered. For example, in the previous experiment, $J_4$ itself was observed to be somewhat smaller than the other criteria, and so was its gradient. Then, the convex-hull minimum-norm element $\omega$ is paradoxically dominated by the gradient of smallest magnitude, since in the convex combination of the gradients, putting the largest weight on the smallest has the effect of reducing the norm of the combination. But this is not efficient, since this gradient corresponds to the already small criterion for which minimization is the least necessary. This observation has led us to calculate the direction $\omega$ as the minimum-norm element in the convex hull of *normalized gradients*. Here, the normalization was made by scaling each gradient to the corresponding value of the individual criterion. In other words, *logarithmic gradients* were considered. In fact, that is exactly what Newton's method does with the global criterion J.

The above experiment was then repeated, using logarithmic gradients to determine the vector $\omega$. The corresponding convergence history is indicated in Fig. 13. This new result is now found

very similar to the analogous result previously achieved by the quasi-Newton method (Fig. 4). The importance of scaling is therefore confirmed.

**Global convergence.** All interface controls $v_i$'s are initially set to 0. The resulting convergence history over 200 iterations is indicated in Fig. 14 for the basic algorithm, and in Fig. 15 for the scaled algorithm based on logarithmic gradients. The first algorithm seems to convergence, but at a very slow rate. The second seems to subject to accidents and to experience a difficulty to enter the asymptotic convergence phase. The criteria stagnate. This may be caused by many factors on which current investigation is focused:

- the insufficiently accurate determination of $\omega$;
- the non-optimality of the scaling of gradients;
- the non-optimality of the step-size, the parameter $\varepsilon$ in (42) being maintained equal to 1 throughout;
- the large dimension of the design space, here 76 (4 interfaces associated with 19 d.o.f.'s).

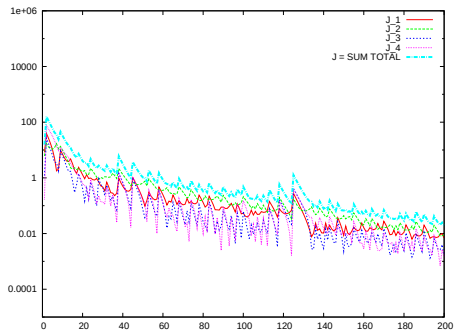


Figure 14: Basic *MGDA* : global convergence

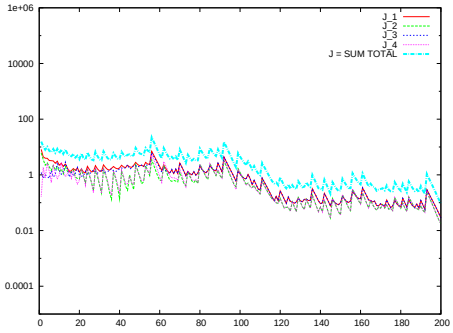


Figure 15: Basic *MGDA* based on logarithmic gradients: global convergence

### 6.4 *MGDA-II*

Recently, a variant, *MGDA-II* , has been proposed in which the descent direction is calculated by a direct procedure, which provides a valuable simplification of implementation, as well as more accurate definition of the direction of search [4]. This new algorithm is now presented again along with a new variant (*MGDA-II b*), and tested on the DDM problem.

**Basic definition, scaling.** In *MGDA-II* , the possibility to prescribe scales for the gradients, $\{S_i\}_{(i=1,...,n)}$ $(S_i > 0 \ (\forall i))$ is offered. In the following experiments, at a given iteration, these scales are either set all equal to 1 ("no scaling prescribed"), or equal to the current values of the criteria ("prescribed scaling"):

$$S_i = J_i \qquad (i = 1, ..., 4)\,; \tag{52}$$

the latter implies that the descent direction is based on *logarithmic gradients*. These scales being supplied by the user, we define the following "scaled gradients":

$$\mathrm{J}'_i = \frac{\nabla J_i}{S_i} \tag{53}$$

Assuming that the gradients form a linearly-independent family, an assumption never contradicted in the numerical experiments, a family of orthogonal, but usually not orthonormal vectors are formed $\{u_i\}_{(i=1,...,n)}$ according to the following:

$$u_1 = \mathrm{J}'_1 \tag{54}$$

and, for $i = 2, 3, ..., n$:

$$u_i = \frac{\mathrm{J}'_i - \sum_{k<i} c_{i,k} u_k}{A_i} \tag{55}$$

where:

$$\forall k < i \ : \ c_{i,k} = \frac{\left(\mathrm{J}'_i, u_k\right)}{\left(u_k, u_k\right)} \tag{56}$$

and

$$A_i = \begin{cases} 1 - \displaystyle\sum_{k<i} c_{i,k} & \text{if nonzero} \\ \\ \varepsilon_i & \text{otherwise} \end{cases} \tag{57}$$

for some arbitrary, but small $\varepsilon_i$ ($0 < |\varepsilon_i| \ll 1$).

The minimum-norm element in the convex of the family $\{u_i\}_{(i=1,...,n)}$ is given by:

$$\omega = \sum_{i=1}^{n} \alpha_i u_i \tag{58}$$

and one finds:

$$\alpha_i = \frac{1}{\|u_i\|^2 \sum_{j=1}^{n} \frac{1}{\|u_j\|^2}} = \frac{1}{1 + \sum_{j \neq i} \frac{\|u_i\|^2}{\|u_j\|^2}} < 1 \tag{59}$$

which confirms that $\omega$ does belong to the interior of the convex hull, so that:

$$\forall i \ : \ \alpha_i \|u_i\|^2 = \frac{\lambda}{2} \qquad \text{(a given Lagrange multiplier)}, \tag{60}$$

and:

$$\forall k \ : \ \left(u_k, \omega\right) = \alpha_k \|u_k\|^2 = \frac{\lambda}{2} = \|\omega\|^2 \ . \tag{61}$$

Convening that $\varepsilon_i = 0$ in the regular case ($\sum_{k<i} c_{i,k} \neq 1$), and otherwise by modifying slightly the definition of the scaling factor to be

$$S'_i = (1 + \varepsilon_i) S_i \, , \tag{62}$$

and redefining the "scaled gradient" accordingly ($\mathrm{J}'_i = \nabla J_i / S'_i$), the following holds:

$$\boxed{\left(\mathrm{J}'_i, \omega\right) = \|\omega\|^2 \quad (\forall i)} \tag{63}$$

that is, the same positive constant [4].

As a result of this direct, fast and accurate construction, the vector $\omega$ is usually different from the former definition, except in special cases, as for example, when $n = 2$ and the angle between the two gradients is obtuse. Nevertheless, the new $\omega$ also provides a descent direction common to all criteria, scaled essentially as initially prescribed.

**Automatic rescaling: *MGDA-II b*.** The examination of the case $n = 2$ has led us to propose a slightly different handling of the scales. Here, one lets

$$A_i = S_i - \sum_{k=1}^{i-1} c_{i,k} \tag{64}$$

only when this number is strictly-positive. Otherwise, the scale $S_i$ is redefined ("automatic rescaling") as follows:

$$S_i = \sum_{k=1}^{i-1} c_{i,k} \tag{65}$$

and one sets $A_i = \varepsilon_i S_i$, for some small $\varepsilon_i$.

The result in (63) is still valid, and it now provides an information on gradients that have been weighted as prescribed whenever $S_i > \sum_{k=1}^{i-1} c_{i,k}$, and otherwise by the procedure itself. This rescaling procedure is certainly perfectible.

**Convergence experiments and discussion.** *MGDA-II* has been tested on the partitioning problem in the four possible options corresponding to "no scaling prescribed" vs "prescribed scaling", and "automatic rescale off" vs "on". In *MGDA-II b*, when $\sum_{k<i} c_{i,k}$ was found greater or equal the prescribed $S_i$ (=1 or $J_i$), $\varepsilon_i$ was set to 0.01 (and maintained to 0 otherwise).

A first observation was made: the new procedure for determining $\omega$ is much faster, and *MGDA-II* seems to be less sensitive to round-off errors.

In Fig. 16 and Fig. 17, the automatic rescale is off, and the effect of scaling alone is evaluated. Over the first 200 iterations, the result is about the same. However the scaled version indicates a trend to convergence acceleration to be confirmed.

In Fig. 18 and Fig. 19, the automatic rescale is on, and the option of prescribed scaling is off/on. Again a better convergence is achieved when scales are prescribed.

In order to confirm these results, the best option "prescribed scales and automatic rescale" is compared with the basic method in Fig. 20 and Fig. 21 over 500 iterations. The trends indicate a linear convergence for the first method, and a seemingly-quadratic convergence for the second. Compared to the quasi-Newton method of Fig. 5, *MGDA-II b* is grossly-speaking twice slower, but it indicates a more definite trend to asymptotic convergence acceleration.

One last remark: in these experiments, we observe that scaling has the effect of making the convergence curves associated with the different criteria closer to one another.

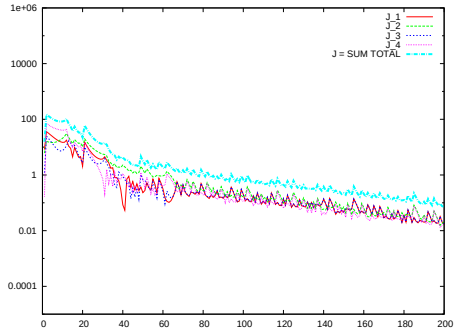


Figure 16: *MGDA-II* , no scaling prescribed

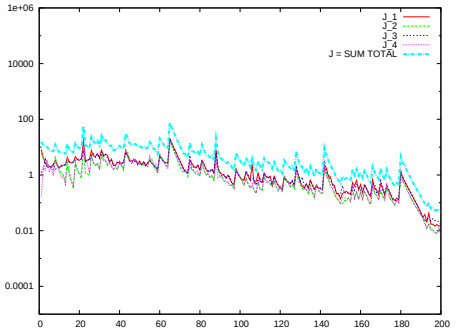


Figure 17: *MGDA-II* , prescribed scaling

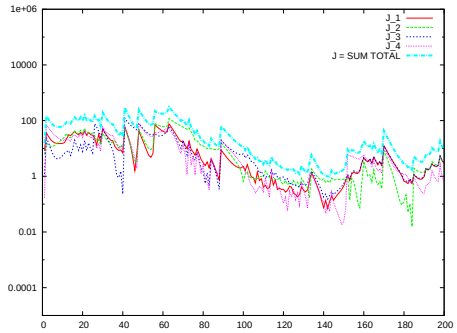


Figure 18: *MGDA-II*, no scaling prescribed, automatic rescale

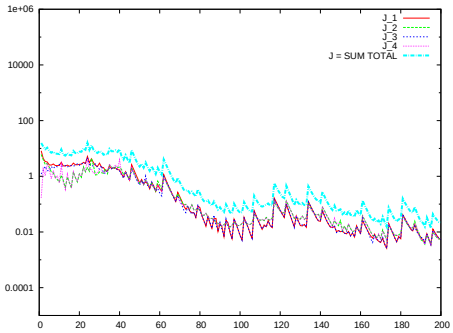


Figure 19: *MGDA-II b*, prescribed scaling, automatic rescale

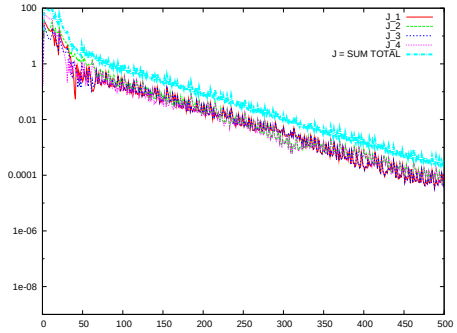


Figure 20: *MGDA-II*, no scaling prescribed, automatic rescale off, 500 iterations

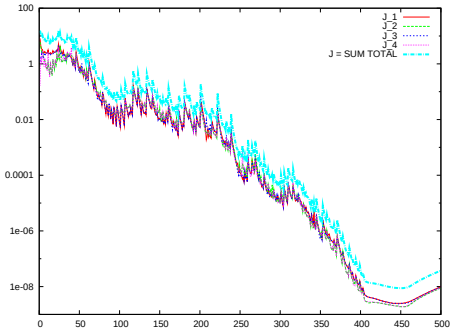


Figure 21: *MGDA-II b*, prescribed scaling, automatic rescale, 500 iterations

## 7 CONCLUSION

In this study, various versions of the *Multiple-Gradient Descent Algorithm* (*MGDA*) have been tested numerically over a domain-partitioning problem treated as a multi-objective problem in which matching defect integrals at the different interfaces are to be minimized concurrently, and in fact, all driven to 0.

The major objective of this experimentation was to assess the potential of *MGDA* to handle multi-objective problems in which the finite-dimensional setting was the result of discretization, thus approaching a more general functional setting. In this respect, the demonstration was made. Indeed convergence was achieved by *MGDA*. However the quasi-Newton method applied to the agglomerated criterion was globally found more efficient, but in the most sophisticated version (*MGDA-II b*), the algorithm seems to demonstrate a promising asymptotically-quadratic convergence.

Thus, if the convergence was not always found satisfactory, several observations should temper this somewhat deceiving conclusion, and many promising directions of improvement can be envisaged:

- in the problem under study, the Pareto set was reduced to the single point corresponding to all criteria equal to 0 associated with the unique solution of the discretized Poisson problem; this situation is really atypical of standard multi-objective problems; additionally, the criteria to be minimized were not really antagonistic, since they all converged at almost the same rate with the quasi-Newton method, leaving little possibility of improvement from the start; for these two reasons *MGDA* has been tested in a very straining situation for which it was not devised originally;

- the large dimension of the design space, here 76 (4 interfaces associated with 19 d.o.f.'s), was probably a handicap;
- a robust procedure to define the step-size should be devised; in our experiments, the parameter $\varepsilon$ was not optimized but maintained equal to 1 throughout;
- the determination of $\omega$ in the basic method should be made more accurately by iterative refinement;
- the scaling of gradients was found important; alternatives to the *logarithmic gradients* should be analyzed and rationalized; more generally, preconditioning remains an open question;
- the *MGDA-II* variant was found faster and more robust;
- at present, our most sophisticated algorithm, *MGDA-II b*, also involves an automatic rescaling procedure; it indicates a definite trend to asymptotic convergence acceleration (quadratic convergence).

## REFERENCES


[1] Gradient descent. WikipediA: The Free Encyclopedia . http://en.wikipedia.org/wiki/Gradient_descent.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. In *IEEE Transactions on Evolutionary Computation*, volume 6 (2), pages 182–197. 2002.

[3] J.-A. Désidéri. Multiple-Gradient Descent Algorithm (MGDA). Research Report 6953, INRIA, 2009. http://hal.inria.fr/inria-00389811.

[4] J.-A. Désidéri. MGDA II: A direct method for calculating a descent direction common to several criteria. Research Report 7422, INRIA, April 2012. http://hal.inria.fr/hal-00685762.

[5] J.-A. Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes rendus - Mathématique*, 350(5-6):313–318, March 2012 . http://dx.doi.org/10.1016/j.crma.2012.03.014.

[6] Jean-Antoine Désidéri. Application of MGDA to domain partitioning. Research Report 7968, INRIA, May 2012. http://hal.inria.fr/hal-00694039.

[7] Ph. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, New York London, 1986.

[8] R. Miettinen. *Nonlinear Multiobjective Optimization*. Kluer Academic Publishers, Boston London Dordrecht, 1999.

[9] A. Zerbinati, J.-A. Désidéri, and R. Duvigneau. Comparison between MGDA and PAES for Multi-Objective Optimization. Research Report 7667, INRIA, June 2011 . http://hal.inria.fr/inria-00605423.

## A GENERALITIES ABOUT CONVEX HULLS AND PARAMETERIZATION

**Notations**

$$\begin{cases} u = \displaystyle\sum_{i=1}^{n} \alpha_i u_i \,;\ \{u_i\}_{(i=1,...,n)} \ :\ \text{given family of vectors in } \mathbb{R}^N \\ \text{constraints: } \forall i \ :\ \alpha_i \geq 0 \,;\ \displaystyle\sum_{i=1}^{n} \alpha_i = 1 \,;\ n \leq N \end{cases} \tag{66}$$

**Parameterization -** The convex hull may parameterized by identifying the set of allowable coefficients $\{\alpha_i\}_{(i=1,...,n)}$. To satisfy the positivity condition automatically, one lets:

$$\alpha_i = \sigma_i^2 \qquad (i = 1, ..., n) \tag{67}$$

Then, the equality constraint

$$\sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \sigma_i^2 = 1 \tag{68}$$

states that

$$\sigma = (\sigma_1, \sigma_2, ..., \sigma_n) \in S_n \tag{69}$$

where $S_n$ is the unit sphere of $\mathbb{R}^n$, and precisely not $\mathbb{R}^N$. This sphere is easily parameterized using trigonometric functions of $n-1$ independent arcs $\phi_1, \phi_2, ..., \phi_{n-1}$:

$$\begin{cases} \sigma_1 = & \cos\phi_1 & . & \cos\phi_2 & . & \cos\phi_3 & . & ... & . & \cos\phi_{n-1} \\ \sigma_2 = & \sin\phi_1 & . & \cos\phi_2 & . & \cos\phi_3 & . & ... & . & \cos\phi_{n-1} \\ \sigma_3 = & 1 & . & \sin\phi_2 & . & \cos\phi_3 & . & ... & . & \cos\phi_{n-1} \\ \vdots & & & & & & & \vdots & & \\ \sigma_{n-1} = & 1 & . & 1 & . & ... & . & \sin\phi_{n-2} & . & \cos\phi_{n-1} \\ \sigma_n = & 1 & . & 1 & . & ... & . & 1 & . & \sin\phi_{n-1} \end{cases} \tag{70}$$

that is:

$$\sigma_i = \sin\phi_{i-1} . \prod_{j=i}^{n-1} \cos\phi_j \tag{71}$$

with $\phi_0 = \frac{\pi}{2}$. It is sufficient to consider the portion of the sphere corresponding to $\phi_i \in [0, \frac{\pi}{2}]$ for all $i \geq 1$ since the sign of the $\sigma_i$'s makes no difference.

The usage of trigonometric functions is not really necessary, since one can let:

$$c_i = \cos^2\phi_i \qquad (i = 1, ..., n) \tag{72}$$

and get:

$$\begin{cases} \alpha_1 = & c_1 & . & c_2 & . & c_3 & . & ... & . & c_{n-1} \\ \alpha_2 = & (1-c_1) & . & c_2 & . & c_3 & . & ... & . & c_{n-1} \\ \alpha_3 = & 1 & . & (1-c_2) & . & c_3 & . & ... & . & c_{n-1} \\ \vdots & & & & & & & \vdots & & \\ \alpha_{n-1} = & 1 & . & 1 & . & ... & . & (1-c_{n-2}) & . & c_{n-1} \\ \alpha_n = & 1 & . & 1 & . & ... & . & 1 & . & (1-c_{n-1}) \end{cases} \tag{73}$$

that is:

$$\alpha_i = (1 - c_{i-1}) \cdot \prod_{j=i}^{n-1} c_j \tag{74}$$

with $c_0 = 0$, and $c_i \in [0, 1]$ for all $i \geq 1$.

In this way, the constraints on the coefficients $\{\alpha_i\}$ have been replaced by the bounds 0 and 1 on the new parameters $\{c_i\}$, independently of one another. However, the criterion to be minimized,

$$\|u\|^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \, (u_i, u_j) \tag{75}$$

is now a polynomial of possibly large degree, namely $2(n-1)$, of the new parameters $\{c_i\}$.

In the particular case of the coordination of 4 sub-domains by *MGDA* , and independently of the degree of refinement of the spatial discretization controlled by the integers $N_X$ and $N_Y$, $n = 4$, and once the 10 scalar products $(u_i, u_j)$ ($i, j = 1, ..., 4$) calculated, the determination of the minimum-norm element $\omega$ is equivalent to minimizing a 6th-degree polynomial of $(c_1, c_2, c_3)$ in the $\mathbb{R}^3$ unit cube (limits included):

$$\begin{cases} \alpha_1 = c_1 c_2 c_3 \\ \alpha_2 = (1 - c_1) c_2 c_3 \\ \alpha_3 = (1 - c_2) c_3 \\ \alpha_4 = (1 - c_3) \end{cases} \tag{76}$$