Optimization and Engineering manuscript No.
(will be inserted by the editor)

Gradient-only approaches to avoid spurious local minima in unconstrained optimization

D. N. Wilke · S. Kok · J. A. Snyman · A. A. Groenwold

Received: date / Accepted: date

Abstract We reflect on some theoretical aspects of gradient-only optimization for the unconstrained optimization of objective functions containing non-physical step or jump discontinuities. This kind of discontinuity arises when the optimization problem is based on the solutions of systems of partial differential equations, in combination with variable discretization techniques (e.g. remeshing in spatial domains, and / or variable time stepping in temporal domains). These discontinuities, which may cause local minima, are artifacts of the numerical strategies used and should not influence the solution to the optimization problem. Although the discontinuities imply that the gradient field is not defined everywhere, the gradient field associated with the computational scheme can nevertheless be computed everywhere; this field is denoted the *associated gradient field*. We demonstrate that it is possible to overcome attraction to the local minima if only associated gradient information is used. Various gradient-only algorithmic options are discussed. A salient feature of

D.N. Wilke (✉)
Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria, 0002, South Africa.
Tel.:+27-12-4202861
Fax: +27-12-3625087
E-mail: nico.wilke@up.ac.za

S. Kok
Advanced Mathematical Modelling, CSIR, Modelling and Digital Science, P.O.Box 395, Pretoria, 0001, South Africa.

J.A. Snyman
Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria, 0002, South Africa.

A.A. Groenwold
Department of Mechanical Engineering, University of Stellenbosch, Stellenbosch, 7602, South Africa.

our approach is that variable discretization strategies, so important in the numerical solution of partial differential equations, can be combined with efficient local optimization algorithms.

Keywords Step discontinuous · Gradient-only optimization · Unconstrained optimization · Partial differential equations · Variable discretization strategies · Shape optimization

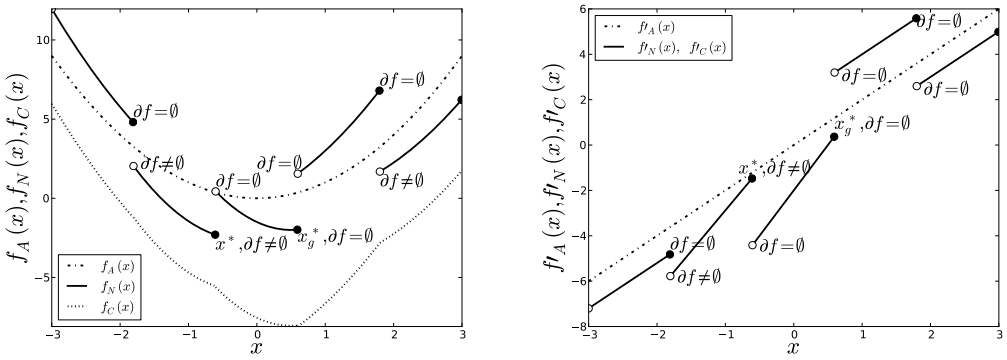
1 Introduction

In this study, we consider some theoretical aspects of gradient-only approaches in unconstrained optimization. Here, gradient-only optimization algorithms refer to optimization strategies that solely considers first order information of a scalar (cost) objective function in computing update directions and update step lengths. We are concerned with piecewise smooth step discontinuous objective functions that contain spurious (local) minima which manifest themselves in the form of step discontinuities. We consider optimization problems that are defined by partial differential equations (PDEs) which are numerically approximated using some discretization strategy, e.g. finite elements or finite differences. However, we assume that the discretization strategies are not constant; this is of crucial importance in many engineering applications, a single example being the requirement for good mesh quality.

The resulting step discontinuous functions are non-differentiable and the gradient field is not defined everywhere [16]. Strategies to allow for optimization of discontinuous functions include smoothing of the discontinuous objective function [25], and more recently by decomposing a discontinuous function into a smooth and non-smooth functions in the neighbourhood of a discontinuity and then by using an active set approach [5]. To the best of our knowledge current approaches to optimize discontinuous functions all act as minimizers, using function values, whereas in this study we propose an alternative by solely focusing on the first order information of discontinuous functions.

Herein, we propose to construct an associated gradient field with the partial derivatives of the gradient vector given by one-sided directional derivatives or partial derivatives when the function is non-differentiable respectively differentiable along partial derivative directions. Such a constructed gradient field follows from the computational scheme since every point has an associated discretization for which (semi) analytical sensitivities [12] of the numerically approximated optimization problem can be calculated. The only requirement is that we use a constant discretization topology when *computing the sensitivities*, i.e. we conduct a consistent sensitivity analysis [15]. We will refer to such a gradient field as an *associated gradient field*. From a computational perspective, an associated gradient field of a discontinuous function is defined everywhere.

During optimization, the domain over which the (P)DEs are solved may remain constant, but the discretization may still be required to change, to ensure convergence or efficiency of the solution. An example is integration over



(a) (b)

Fig. 1.1: (a) Plot depicting a piecewise smooth univariate step discontinuous numerical or approximate objective function $f_N(x)$ of an underlying (unknown) continuous analytical objective function $f_A(x)$ of an optimization problem. We include a projected C^0 objective function $f_C(x)$ which is constructed from $f_N(x)$ by removing all the discontinuities. Also indicated at the discontinuities are the subgradients ∂f, that are defined if $\partial f \neq \emptyset$, but not defined if $\partial f = \emptyset$. (b) Plot depicting the associated derivative of $f_N(x)$ and derivative of $f_A(x)$. The derivative field $f'_A(x)$ is, as indicated, defined everywhere; likewise the associated derivative field for $f'^A_N(x)$ and $f'^A_C(x)$.

a fixed time domain using variable time steps. Alternatively, the design variables may describe the domain over which the (P)DEs are solved. A change in design variables therefore changes the solution domain, which in turn requires the discretization to change. An illustrative example is shape optimization.

We distinguish between two classes of discretization strategies. First, constant discretization strategies continuously adjust some reference discretization when the solution domains change (and generate a fixed discretization if the solution domain remains constant). Secondly, variable discretization strategies generate new independent discretizations irrespective of whether or not the solution domain changes. Temporal (P)DEs may for example be solved using fixed or variable time steps. For spatial PDEs, the equivalents are fixed and mesh movement strategies versus remeshing. Fixed time steps and mesh movement strategies however may imply serious difficulties, e.g. impaired convergence rates and highly distorted grids and meshes, which may even result in failure of the computational procedures used. The variable discretization strategies are preferable by far.

Thus, a consequence of using variable discretization strategies while solving an optimization problem is that the resulting objective functions contain discontinuities. Consider Figure 1.1(a), which depicts three functions that describe an optimization problem. The functions are an unknown analytical func-

tion $f_A(x)$, a numerically computed approximate piecewise smooth step discontinuous objective function $f_N(x)$, and a projected C^0 continuous objective function constructed by removing the discontinuities from $f_N(x)$. All three functions describe an (approximate) objective function to the same optimization problem, in turn based on some system of partial differential equations. Usually, $f_N(x)$ is the objective function used when searching for the solution to the optimization problem, since $f_A(x)$ is unknown, and to construct $f_C(x)$ from $f_N(x)$ would be computationally expensive. Accordingly, the optimum x^* and the positive associated gradient projection point x_g^* are based on $f_N(x)$.

Here, we refer to x_g^* as a positive associated gradient projection point since the *associated directional derivatives* in all directions around this point are positive. For the sake of brevity however, we will simply refer to x_g^* as a positive projection point, unless we specifically want to distinguish between types of positive projection points. In addition, we indicate whether the subgradients ∂f at the discontinuities are defined ($\partial f \neq \emptyset$) or not defined ($\partial f = \emptyset$). Since $f_N(x)$ is not convex, the subderivatives over parts of the piecewise smooth sections of the functions are *strictly* speaking not defined, since a line (hyperplane for multidimensional functions) constructed from a subderivative is required to support the epigraph of $f_N(x)$.

Gradient-only optimization solves an optimization problem by finding the positive projection point $\boldsymbol{x}_g^*$, as opposed to mathematical programming which aims to find the optimum $\boldsymbol{x}^*$ to solve an optimization problem. It is important to distinguish between $\boldsymbol{x}^*$ and $\boldsymbol{x}_g^*$, since they may be distinctly different. Whether the optimum $\boldsymbol{x}^*$ or the positive projection point $\boldsymbol{x}_g^*$ is a more suitable solution to $f_N(x)$ will ultimately depend on which one best describes the optimum of $f_A(x)$. In addition, a positive projection point allows for an alternative formulation to an optimization problem than the minimization formulation of mathematical programming, which may allow for easier and more flexible formulations of an optimization problem. An important spatial example is structural shape optimization in which fixed or mesh movement strategies are almost always used; the very motivation for this being that remeshing strategies cannot be used efficiently, due to the induced non-physical local minima during optimization, e.g. see References [1,4,9,22]. However, we have shown [23] that remeshing can be successfully handled by reformulating the solution of the shape optimization problem to be a positive projection point instead of the minimum.

An important observation is that the positive projection point in $f_N(x)$ coincides with the minimum of $f_C(x)$, as is depicted in Figure 1.1. Essentially, gradient-only optimization allows for the minimization of $f_C(x)$ directly from $f_N(x)$.

For the sake of simplicity, we will in the following omit the subscripts of $f_N(x)$ and $f_A(x)$, and merely refer to $f_N(x)$ as $f(x)$, which is the function for which we aim to find the *positive projection point* $\boldsymbol{x}_g^*$.

Let us first present two illustrative examples of non-physical step discontinuities, to set the tone for our paper. The first is rather trivial, the second not quite.

1.1 Univariate example problem: Newton's cooling law

Consider Newton's law of cooling, which states that the rate of heat loss of a body is proportional to the difference in temperature between a body and the surroundings of that body, given by the linear first order DE:

$$\frac{dT}{dt} = -\kappa(T(t) - T_{\text{env}}), \tag{1.1}$$

with the well known analytical solution

$$T(t) = T_{\text{env}} + (T_{\text{init}} - T_{\text{env}})e^{-\kappa t}. \tag{1.2}$$

Here κ is a positive proportionality constant, $T(t)$ the temperature of the body at time t, T_{init} its initial temperature, and T_{env} the temperature of the surroundings of the body.

We consider the temperature $T(t)$ of a body after 1s, for $0.5 \leq \kappa \leq 2$, with $T(0) = 100°\text{C}$ at $t = 0$, and $T_{\text{env}} = 10°\text{C}$ for all t. The analytical solution of the temperature $T(1)$ of the body is depicted in Figure 1.2(a) and the *associated derivative* of $T(1)$ w.r.t. κ is depicted in Figure 1.2(b).

Solving Eq. (1.1) for $0.5 \leq \kappa \leq 2$ with a forward Euler method using a variable time stepping strategy introduces step discontinuities in the temperature response; this is shown in Figure 1.2(a). For the variable time step strategy we decrease the time step whenever an allowed temperature increment is exceeded, otherwise we gradually increase the time step. The corresponding discontinuous derivatives are plotted in Figure 1.2(b). Again note that although discontinuous, the *associated derivatives* are uniquely defined everywhere, and can be computed.

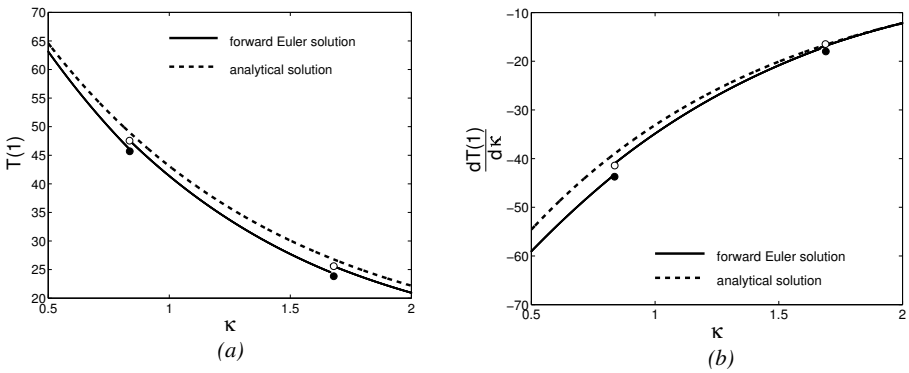


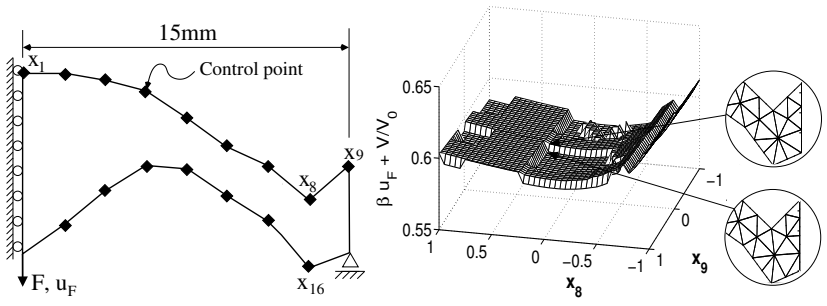
Fig. 1.2: Numerical and analytical solutions for Newton's cooling law. (a) Temperature T after 1 second for $0.5 \leq \kappa \leq 2$, and (b) the corresponding *associated derivative* $\frac{dT(1)}{d\kappa}$.

1.2 Multivariate example problem: Shape optimization

Next, we consider a non-trivial benchmark problem in structural shape optimization, namely the so-called Michell structure [7] depicted in Figure 1.3(a). The geometry is represented using 16 control variables with only vertical degrees of freedom, and with piecewise linear interpolation between the control points. The objective of this shape optimization problem is to minimize the sum of the vertical displacement βu_{F} at the point of load application and the normalized volume $\frac{V}{V_0}$ for a unit thickness structure with $\mathrm{F} = 1\mathrm{N}$, $V_0 = 150\mathrm{mm}^3$ and $\beta = 1$.

The displacement u_{F} is computed using a linear elastic finite element method with linear strain triangular elements (e.g. see [6]). For the material properties we use Young's modulus $E = 200\mathrm{GPa}$ and Poisson's ratio $\nu = 0.3$. The meshes required for the finite element analyses are generated using a quadratic convergent remeshing strategy [24] with ideal element length $h_0 = 1\mathrm{mm}$. To illustrate the discontinuous nature of the objective function, the two control variables x_8 and x_9 are perturbed around the reference configuration depicted in Figure 1.3(a) over the range -1.0 through 1.0, using constant intervals of 0.05.

The resulting objective function values are shown in Figure 1.3(b). The step discontinuities due to remeshing are clearly evident; they result since the number of nodes, and the nodal connectivity, changes. This is evident from Figure 1.3(b): a small decrease in x_9 results in 3 elements (top insert in Figure 1.3(b)) as opposed to 4 elements (bottom insert Figure 1.3(b)) on the rightmost edge of the structure.



(a) *(b)*

Fig. 1.3: (a) Structure, boundary conditions and control variables and (b) the sum of the vertical displacement u_{F} and the normalized volume $\frac{V}{V_0}$ for variations of the two rightmost upper control variables (x_8, x_9) for the Michell shape optimization problem.

1.3 Introductory comments

Clearly, the introduced non-physical discontinuities cannot be accommodated in optimization methods developed for C^0 or C^1 continuous objective functions. However, again note that the *associated gradients* of the piecewise smooth step-discontinuous functions considered in this study are uniquely defined everywhere. Consider the positive projection point x_g^* that occurs over a discontinuity as depicted by $f_N(x)$ in Figure 1.1, with a piecewise smooth part to the left and a piecewise smooth part to the right of x_g^*. Both the left and the right hand limits represent approximations to the analytical value of the objective function; the left and right hand limits differ only as a result of the *discretization technique* used, and these values approach each other in the limit of mesh refinement anyway. Hence, the value of the objective function being reported is not unique.

In this study, we consider the unconstrained optimization of objective functions containing non-physical step or jump discontinuities with accurate gradients that are everywhere defined. For the sake of brevity, we restrict our efforts to finding positive projective points (but the equivalents for negative projection points are clear). Our paper is organized as follows: we present definitions for optimality that are solely based on the gradient of a function in Section 2. In Section 3, we introduce the gradient-only optimization problem and in Appendix A we offer proofs of convergence of descent sequences defined in the previous section. We give practical considerations regarding gradient-only optimization algorithms in Section 4, and present a brief comparative discussion of classical mathematical programming and gradient-only optimization in Section 5. In Section 6 we present a shape optimization problem of practical importance, and a number of analytical test functions. Concluding remarks then follow.

2 Definitions

Not all step discontinuities are necessarily problematic for classical optimization, and we distinguish between two step discontinuity types, namely those that are *inconsistent* with the function *trend*, and those that are *consistent* with the function *trend*, as shown in Figure 2.1. (All other discontinuities may be taken to be representable of either a local minimum or a local maximum.) To represent semi-continuity of f we introduce a double empty/filled circle convention as depicted in Figure 2.1(a), where a filled circle indicates $F(\lambda_0)$. Upper semi-continuity is represented by the filled/empty circle pair indicated by 1's in Figure 2.1(a) i.e. the filled/empty circles lie *above* f. Lower semi-continuity in turn is represented by the empty/filled circle pair indicated by 2's, i.e. the empty/filled circles lie *below* f, in Figure 2.1(a).

Figure 2.1(a) depicts an inconsistent step discontinuity; the function decreases as λ increases, but the step discontinuity results in an increase of the

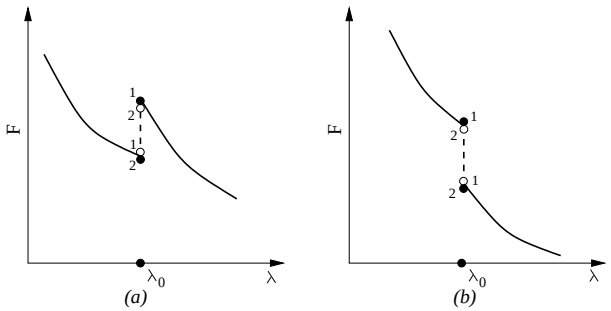


Fig. 2.1: Upper and lower semi-continuous univariate functions with (a) an inconsistent step discontinuity, and (b) a consistent step discontinuity.

function over the step discontinuity. Similarly, Figure 2.1(b) depicts a consistent step discontinuity.

The functions we consider in this study are step-discontinuous and therefore not everywhere differentiable. However, computationally the derivatives and gradients are everywhere computable since the analysis is per se restricted to the part of the objective function before, or after a discontinuity. We therefore define an *associated derivative* $f'^{A}(x)$ and *associated gradient* $\nabla_A f(\boldsymbol{x})$ which follows computationally when the sensitivity analysis is consistent [15]. Firstly, we define the *associated derivative*

Definition 2.1 Let $f : X \subset \mathbb{R} \to \mathbb{R}$ be a piecewise smooth real univariate step-discontinuous function that is everywhere defined. The *associated derivative* $f'^{A}(x)$ for $f(x)$ at a point x is given by the derivative of $f(x)$ at x when $f(x)$ is differentiable at x. The *associated derivative* f'^{A} for $f(x)$ non-differentiable at x, is given by the left-sided derivative of $f(x)$ when x is associated with the piecewise continuous section of the function to the left of the discontinuity, otherwise it is given by the right-sided derivative.

Secondly, the *associated gradient* is defined as follows:

Definition 2.2 Let $f : X \subset \mathbb{R}^n \to \mathbb{R}$ be a piecewise continuous function that is everywhere defined. The *associated gradient* $\nabla_A f(\boldsymbol{x})$ for $f(\boldsymbol{x})$ at a point $\boldsymbol{x}$ is given by the gradient of $f(\boldsymbol{x})$ at $\boldsymbol{x}$ when $f(\boldsymbol{x})$ is differentiable at $\boldsymbol{x}$. The *associated gradient* $\nabla_A f(\boldsymbol{x})$ for $f(\boldsymbol{x})$ non-differentiable at $\boldsymbol{x}$ is defined as the vector of partial derivatives with each partial derivative an *associated derivative* (see Definition 2.1).

It follows from Definitions 2.1 and 2.2 that the *associated gradient* reduces to the gradient of a function that is everywhere differentiable.

We now proceed to develop a self-contained theoretical framework for gradient-only optimization; what follows is a rather straightforward extension of classical concepts.

Definition 2.3 Let $f : (a, b) \subset \mathbb{R} \to \mathbb{R}$ be a real univariate function that is not necessarily continuous in both function value $f(\lambda)$ and *associated derivative* $f'^{A}(\lambda)$ but for which $f(\lambda)$ and $f'^{A}(\lambda)$ are uniquely defined for every $\lambda \in (a, b)$. Then, $f(\lambda)$ is said to have a (resp., strictly) negative *associated derivative* on (a, b) if $f'^{A}(\lambda)$ (resp., $<$) ≤ 0, $\forall\ \lambda \in (a, b)$, e.g. see Figure 2.1. Conversely, $f(\lambda)$ is said to have a (resp., strictly) *positive associated derivative* on (a, b) if $f'^{A}(\lambda)$ (resp., $>$) ≥ 0, $\forall\ \lambda \in (a, b)$.

Next, we define lower and upper semi-continuity of the *associated gradient*.

Definition 2.4 Let $f : X \subset \mathbb{R}^n \to \mathbb{R}$ be a real valued function with an *associated gradient* field $\nabla_A f(\boldsymbol{x})$ that is uniquely defined for every $\boldsymbol{x} \in X$.

- Then the *associated directional derivative* along a normalized direction $\boldsymbol{u} \in \mathbb{R}^n$ is lower semi-continuous at $\boldsymbol{y} \in X$ if

$$\nabla_A{}^{\mathrm{T}} f(\boldsymbol{y})\boldsymbol{u} \leq \liminf_{\lambda \to 0^{\pm}} \nabla_A{}^{\mathrm{T}} f(\boldsymbol{y} + \lambda \boldsymbol{u})\boldsymbol{u},\ \lambda \in \mathbb{R}.$$

- The *associated directional derivative* along a normalized direction $\boldsymbol{u} \in \mathbb{R}^n$ is upper semi-continuous at $\boldsymbol{y} \in X$ if

$$\nabla_A{}^{\mathrm{T}} f(\boldsymbol{y})\boldsymbol{u} \geq \limsup_{\lambda \to 0^{\pm}} \nabla_A{}^{\mathrm{T}} f(\boldsymbol{y} + \lambda \boldsymbol{u})\boldsymbol{u},\ \lambda \in \mathbb{R}.$$

- The *associated directional derivative* along a normalized direction $\boldsymbol{u} \in \mathbb{R}^n$ is pseudo-continuous at $\boldsymbol{y} \in \mathbb{R}^n$ if it is both upper and lower semi-continuous at $\boldsymbol{y}$.

We note that a univariate function $f(\lambda)$ may be step-discontinuous at a point $\bar{\lambda} \in (a, b)$, but the *associated derivative* may still be pseudo-continuous at $\bar{\lambda}$, e.g. the function

$$f(\lambda) = \begin{cases} \lambda^2, & \lambda < -1 \\ \lambda^2 - 2, & \lambda \geq -1 \end{cases},$$

is not pseudo-continuous at $\bar{\lambda} = 1$. However, the *associated derivative*

$$f'^{A}(\lambda) = \begin{cases} 2\lambda, & \lambda < -1 \\ 2\lambda, & \lambda \geq -1 \end{cases},$$

is pseudo-continuous at $\bar{\lambda} = -1$, where we defined the *associated derivative* at $\bar{\lambda} = -1$ by the right-hand limit.

3 Gradient-only optimization problem

We now present the general unconstrained gradient-only optimization problem that is equivalent to the classical minimization problem for smooth convex cost functions [18].

Problem 3.1 Given a real-valued function $f : X \subset \mathbb{R}^n \to \mathbb{R}$, find a non-negative associated gradient projection point $\boldsymbol{x}_g^* \in X$ such that for every $\boldsymbol{u} \in \{\boldsymbol{y} \in \mathbb{R}^n \,/\, \|\boldsymbol{y}\| = 1\}$ there exists a real number $r_u > 0$, and the following holds:

$$\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}_g^* + \lambda \boldsymbol{u})\boldsymbol{u} \geq 0 \; \forall \; \lambda \in (0, r_u].$$

Accordingly, we define non-negative generalized associated gradient projection point that characterizes a minimum according to the *associated gradient* field of a scalar function, be it local or global, as follows:

Definition 3.2 Suppose that $f : X \subset \mathbb{R}^n \to \mathbb{R}$ is a real-valued function for which the *associated gradient* field $\nabla_A f(\boldsymbol{x})$ is uniquely defined for every $\boldsymbol{x} \in X$.

Then, a point $\boldsymbol{x}_g^* \in X$ is a generalized non-negative associated gradient projection point (G-NN-GPP) if there exists a real number $r_u > 0$ for every $\boldsymbol{u} \in \{\boldsymbol{y} \in \mathbb{R}^n \,/\, \|\boldsymbol{y}\| = 1\}$ such that

$$\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}_g^* + \lambda \boldsymbol{u})\boldsymbol{u} \geq 0, \; \forall \; \lambda \in (0, r_u].$$

A special case of Problem 3.1 is given below which we refer to as the strict unconstrained gradient-only optimization problem.

Problem 3.3 Given a real-valued function $f : X \subset \mathbb{R}^n \to \mathbb{R}$, find a $\boldsymbol{x}_g^* \in X$ such that for every $\boldsymbol{u} \in \{\boldsymbol{y} \in \mathbb{R}^n \,/\, \|\boldsymbol{y}\| = 1\}$ there exists a real number $r_u > 0$, and the following holds:

$$\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}_g^* + \lambda \boldsymbol{u})\boldsymbol{u} > 0 \; \forall \; \lambda \in (0, r_u].$$

Accordingly, we define a strict non-negative *associated gradient projection point* to imply a minimum according to the *associated gradient* field of a scalar function, be it local or global, as follows:

Definition 3.4 Suppose that $f : X \subset \mathbb{R}^n \to \mathbb{R}$ is a real-valued function for which the *associated gradient* field $\nabla_A f(\boldsymbol{x})$ is uniquely defined for every $\boldsymbol{x} \in X$.

Then, a point $\boldsymbol{x}_g^* \in X$ is a strict non-negative associated gradient projection point (S-NN-GPP) if there exists a real number $r_u > 0$ for every $\boldsymbol{u} \in \{\boldsymbol{y} \in \mathbb{R}^n \,/\, \|\boldsymbol{y}\| = 1\}$ such that

$$\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}_g^* + \lambda \boldsymbol{u})\boldsymbol{u} > 0, \; \forall \; \lambda \in (0, r_u].$$

It follows that the strict unconstrained gradient-only optimization problem is included in the generalized unconstrained gradient-only optimization problem. Note that our definition of a G-NN-GPP is consistent with the classical mathematical programming definition of a minimum for smooth functions.

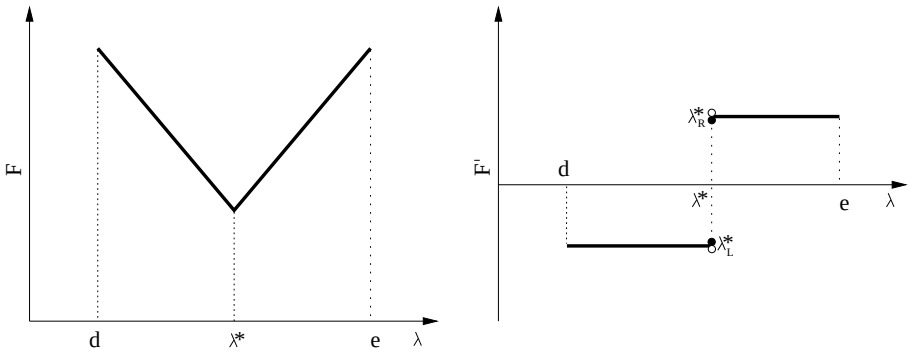


Fig. 3.1: An illustration of (a) the function value and (b) the corresponding *associated derivative* that is either upper or lower semi-continuous, with a step-discontinuous gradient projection point (GPP) in $\in (d, e)$.

3.1 Discontinuous gradient projection points (GPP)

Our newly introduced definitions for a non-negative associated gradient projection point (NN-GPP) or a non-positive associated gradient projection point (NP-GPP) of a function only require that the *associated gradient* field be uniquely defined everywhere; no assumptions regarding the continuity of the function are required. Hereafter associated gradient projection point (GPP) or associated gradient projection set (GPS) is used to imply either a non-negative or non-positive associated gradient projection (point / set). We therefore omit the conventional inclusion of a saddle (point / set). In addition the function may be discontinuous at a GPP. We first consider discontinuous GPPs for univariate functions and then for multivariate functions. An example of a function with a discontinuous NN-GPP is the absolute value function with the *associated derivative* at the minimum point λ^* defined by either the left or right limit as depicted in Figure 3.1, as opposed to the conventional undefined derivative at λ^*. The *associated derivative* at λ^* is therefore either upper or lower semi-continuous as indicated by the double empty/filled notation.

Proposition 3.5 *Let $f : [d, e] \subset \mathbb{R} \to \mathbb{R}$ be a real univariate function that is not necessarily continuous in both function value $f(\lambda)$ and* associated derivative *$f'^{A}(\lambda)$ but for which $f(\lambda)$ and $f'^{A}(\lambda)$ are uniquely defined for every $\lambda \in [d, e]$. In addition, let $f'^{A}(\lambda)$ be step-discontinuous (upper or lower* associated derivative *semi-continuous) at a gradient projection point (GPP) $\lambda^* \in (d, e)$ according to Definition (resp. 3.2 / 3.4). Let λ^*_L be the left limit and λ^*_R the right limit of λ^*.*

Then in addition to the GPP λ^*, either λ^*_L is a GPP if

$$\lim_{\lambda \to \lambda^{*-}} f'^{A}(\lambda) \neq f'^{A}(\lambda^*),$$

or λ_R^* *is a GPP if*

$$\lim_{\lambda \to \lambda^{*+}} f'^A(\lambda) \neq f'^A(\lambda^*).$$

Proof This is immediate from Definition 3.2.

For multivariate functions we can state a similar proposition.

Proposition 3.6 *Let* $f : X \subset \mathbb{R}^n \to \mathbb{R}$ *be a real valued function with* associated gradient *field* $\nabla_A f(\boldsymbol{x})$ *that is uniquely defined for every* $\boldsymbol{x} \in X$*. In addition let* $\nabla_A f(\boldsymbol{x})$ *be step-discontinuous at a GPP* $\boldsymbol{x}_g^* \in X$ *according to Definition 3.2. Then* $\boldsymbol{y} = \lim_{\boldsymbol{z} \to \boldsymbol{x}_g^*} \boldsymbol{z}$ *with* $\boldsymbol{z} \in X$ *is also a GPP if*

Proof This is immediate from Proposition 3.5.

We now introduce a gradient projection set (GPS) to accommodate all the gradient projection points (GPPs) in the compact neighbourhood of a GPP $\boldsymbol{x}_g^*$.

Definition 3.7 Let $f : X \subset \mathbb{R}^n \to \mathbb{R}$ be a real valued function with *associated gradient* field $\nabla_A f(\boldsymbol{x})$ that is uniquely defined for every $\boldsymbol{x} \in X$. In addition let $\boldsymbol{x}_g^* \in X$ be a GPP according to Definition 3.2.

We define the set S as follows:

$$S = \left\{ \boldsymbol{x}_g^*, \boldsymbol{y} : \lim_{\boldsymbol{y} \to \boldsymbol{x}_g^*} \nabla_A f(\boldsymbol{y}) \neq \nabla_A f(\boldsymbol{x}_g^*),\ \forall\, \boldsymbol{y} \in \mathbb{R}^n \right\}$$

The set S is then a GPS of $\boldsymbol{x}_g^*$ if every $\boldsymbol{x} \in S$ is a GPP according to Definition 3.2.

It is straightforward to show that our definition of a GPS reduces to a singleton $\nabla f(\boldsymbol{x}_g^*) = \boldsymbol{0}$ for smooth functions, being consistent with the mathematical programming definition of a minimum point.

3.2 Derivative descent sequences

Having defined GPPs and GPSs solely based on the *associated gradient* field of a function, we proceed to define descent sequences that only consider the *associated gradient* field of a function.

Definition 3.8 For a given sequence $\{\boldsymbol{x}^{\{k\}} \in X \subset \mathbb{R}^n : k \in \mathbb{P}\}$ suppose $\nabla_A f(\boldsymbol{x}^{\{k\}}) \neq \boldsymbol{0}$ for some k and $\boldsymbol{x}^{\{k\}} \notin S$ with S defined in Definition 3.7. Then the sequence $\{\boldsymbol{x}^{\{k\}}\}$ is an associated derivative descent sequence for $f : X \to \mathbb{R}$, if an associated sequence $\{\boldsymbol{u}^{\{k\}} \in \mathbb{R}^n : k \in \mathbb{P}\}$ may be generated such that if $\boldsymbol{u}^{\{k\}}$ is a descent direction from the set of all possible descent directions at $\boldsymbol{x}^{\{k\}}$, i.e. $\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}^{\{k\}}) \boldsymbol{u}^{\{k\}} < 0$ then

$$\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}^{\{k+1\}}) \boldsymbol{u}^{\{k\}} < 0, \text{ for } \boldsymbol{x}^{\{k\}} \neq \boldsymbol{x}^{\{k+1\}} \tag{3.1}$$

We also include the definition of a stricter class of associated derivative descent sequences which we require for convergence proofs of multimodal functions of dimension two and higher in order to exclude oscillating sequences. Oscillating sequences may occur when the sequence defined in Definition 3.8 is considered.

Definition 3.9 For a given sequence $\{\boldsymbol{x}^{\{k\}} \in X \subset \mathbb{R}^n : k \in \mathbb{P}\}$ suppose $\nabla_A f(\boldsymbol{x}^{\{k\}}) \neq \mathbf{0}$ for some k and $\boldsymbol{x}^{\{k\}} \notin S$ with S defined in Definition 3.7. Then the sequence $\{\boldsymbol{x}^{\{k\}}\}$ is a conservative associated derivative descent sequence for $f : X \to \mathbb{R}$, if an associated sequence $\{\boldsymbol{u}^{\{k\}} \in \mathbb{R}^n : k \in \mathbb{P}\}$ may be generated such that if $\boldsymbol{u}^{\{k\}}$ is a descent direction from the set of all possible descent directions at $\boldsymbol{x}^{\{k\}}$ then

$$\nabla_A{}^{\mathrm{T}} f\left(\boldsymbol{x}^{\{k\}} + \lambda(\boldsymbol{x}^{\{k+1\}} - \boldsymbol{x}^{\{k\}})\right) \boldsymbol{u}^{\{k\}} < 0, \forall\, \lambda \in [0,1] \text{ for } \boldsymbol{x}^{\{k\}} \neq \boldsymbol{x}^{\{k+1\}}. \tag{3.2}$$

We offer convergence proofs for univariate and multidimensional derivative descent sequences in Appendix A.

4 Practical algorithmic considerations

We now consider some practical algorithmic implications of the foregoing, relying in particular on the new definitions for a GPP presented in Definition 3.2.

We aim to give a fairly general outline for modifying classical gradient based optimization algorithms to become gradient-only optimization algorithms; often this merely requires subtle modifications to conventional gradient based algorithms. We consider two classes of optimization algorithms, namely line search descent methods, and approximation methods; both are prevalent in practical optimization. In addition, we also consider the non-smooth gradient-only r-algorithm for optimization of C^0 continuous functions by Shor [16].

4.1 Line search descent methods

Line search methods are generally present in first order methods (e.g. steepest descent and conjugate gradient methods), and second order methods (e.g. modified Newton methods like Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS)) [18]. In any event, for a given iteration k, the current position is given by $\boldsymbol{x}^{\{k-1\}},\ k = 1, 2, 3, \ldots$ and search direction $\boldsymbol{u}^{\{k\}}$ at $\boldsymbol{x}^{\{k-1\}}$. In general, line search methods use function value and directional derivative information to predict an update step along the search direction $\boldsymbol{u}^{\{k\}}$. In formulating a rudimentary gradient-only algorithm, the line search simply needs to be modified to consider only the directional derivative along the search direction $\boldsymbol{u}^{\{k\}}$. In this study we consider two inexact line search algorithms.

Firstly, we consider the zoom algorithm [11] to satisfy the strong Wolfe conditions which requires that the following two conditions are satisfied: the sufficient decrease condition given by

$$f(\boldsymbol{x}^{\{k\}} + \lambda^{\{k\}}\boldsymbol{u}^{\{k\}}) \leq f(\boldsymbol{x}^{\{k\}}) + c_1\lambda^{\{k\}}\nabla^{\mathrm{T}}f(\boldsymbol{x}^{\{k\}})\boldsymbol{u}^{\{k\}}, \tag{4.1}$$

and the curvature condition given by

$$|\nabla^{\mathrm{T}}f(\boldsymbol{x}^{\{k\}} + \lambda^{\{k\}}\boldsymbol{u}^{\{k\}})\boldsymbol{u}^{\{k\}}| \leq c_2|\nabla^{\mathrm{T}}f(\boldsymbol{x}^{\{k\}})\boldsymbol{u}^{\{k\}}|, \tag{4.2}$$

with the parameters satisfying $0 < c_1 < c_2 < 1$. Clearly, (4.1) uses both function values and directional derivatives whereas (4.2) only requires directional derivatives. The algorithm is outlined in [11] (Algorithm 3.5) which we refer to as BFGS(f) to signify the use of both function values and directional derivatives.

Secondly, we consider a line search that considers only directional derivatives. We start with some maximum step length λ_{max} and stop when either (4.2) or

$$\nabla^{\mathrm{T}}f(\boldsymbol{x}^{\{k\}} + \lambda^{\{k\}}\boldsymbol{u}^{\{k\}})\boldsymbol{u}^{\{k\}} \leq 0 \tag{4.3}$$

is satisfied. Otherwise, we half the step and continue. We refer to this gradient-only BFGS algorithm as BFGS(g). A discussion on gradient-only exact line searches is presented in [23] and gradient-only interpolation methods presented in [17].

4.1.1 Algorithmic implementation

We now consider the algorithmic implementation of the second-order line search BFGS method to solve unconstrained gradient-only optimization problems. Given an initial point $\boldsymbol{x}^{\{0\}}$, the BFGS implementation proceeds as follows:

1. **Initialization:** Select real constants $\epsilon > 0$, $c_1 > 0$ and $c_2 > 0$. Select integer constants k_{max} and l_{max}. Set $\boldsymbol{G}^{\{0\}} = \boldsymbol{I}$ initially then update $\boldsymbol{G}^{\{0\}} = \frac{(\boldsymbol{y}^{\{1\}})^{\mathrm{T}}\boldsymbol{v}^{\{1\}}}{(\boldsymbol{y}^{\{1\}})^{\mathrm{T}}\boldsymbol{y}^{\{1\}}}\boldsymbol{I}$ after first step. Set $k := 0$ and $l := 0$.
2. **Gradient evaluation:** Compute $\boldsymbol{\nabla} f(\boldsymbol{x}^{\{k\}})$.
3. **Update the search direction** $\boldsymbol{u}^{\{k+1\}} = -\boldsymbol{G}^{\{k\}}\boldsymbol{\nabla} f(\boldsymbol{x}^{\{k\}})$.
4. **Initiate an inner loop to conduct line search:** Find $\lambda^{\{k+1\}}$ using either the function value or gradient-only line search strategy described in Section 4.1 with $l := l + 1$, counting the number of required iterations.
5. **Test for re-initialization of** $\boldsymbol{G}^{\{k\}}$**:** if $|\frac{\boldsymbol{\nabla}^{\mathrm{T}}f(\boldsymbol{x}^{\{k\}})\boldsymbol{\nabla}f(\boldsymbol{x}^{\{k-1\}})|}{\|\boldsymbol{\nabla}f(\boldsymbol{x}^{\{k-1\}})\|} < 0.1$ then $\boldsymbol{G}^{\{k\}} = \boldsymbol{I}$ else

$$\boldsymbol{G}^{\{k\}} = \boldsymbol{G}^{\{k-1\}} + \left[1 + \frac{(\boldsymbol{y}^{\{k\}})^{\mathrm{T}}\boldsymbol{G}^{\{k-1\}}\boldsymbol{y}^{\{k\}}}{(\boldsymbol{v}^{\{k\}})^{\mathrm{T}}\boldsymbol{y}^{\{k\}}}\right]\left[\frac{\boldsymbol{v}^{\{k\}}(\boldsymbol{v}^{\{k\}})^{\mathrm{T}}}{(\boldsymbol{v}^{\{k\}})^{\mathrm{T}}\boldsymbol{y}^{\{k\}}}\right] - \left[\frac{\boldsymbol{v}^{\{k\}}(\boldsymbol{y}^{\{k\}})^{\mathrm{T}}\boldsymbol{G}^{\{k-1\}} + \boldsymbol{G}^{\{k-1\}}\boldsymbol{y}^{\{k\}}(\boldsymbol{v}^{\{k\}})^{\mathrm{T}}}{(\boldsymbol{v}^{\{k\}})^{\mathrm{T}}\boldsymbol{y}^{\{k\}}}\right],$$

with $\boldsymbol{v}^{\{k\}} = \lambda^{\{k\}}\boldsymbol{u}^{\{k\}}$ and $\boldsymbol{y}^{\{k\}} = \left(\boldsymbol{\nabla} f(\boldsymbol{x}^{\{k\}}) - \boldsymbol{\nabla} f(\boldsymbol{x}^{\{k-1\}})\right)$.

6. **Move to the new iterate:** Set $\boldsymbol{x}^{\{k+1\}} := \boldsymbol{x}^{\{k\}} + \lambda^{\{k+1\}}\boldsymbol{u}^{\{k+1\}}$.
7. **Convergence test:** if $\|\boldsymbol{x}^{\{k+1\}} - \boldsymbol{x}^{\{k\}}\| \leq \epsilon$ OR $k = k_{\max}$, stop.
8. **Initiate an additional outer loop:** Set $k := k + 1$ and goto Step 2.

4.2 R-algorithm

In addition, we present results obtained with the r-algorithm developed by Shor [16] for the minimization of C^0 functions. The r-algorithm is a gradient-only optimization algorithm that keeps track of the minimum function value obtained along the search path. Hence, this algorithm converges to positive projection points $\boldsymbol{x}_g^*$ but only reports $\boldsymbol{x}_g^*$ as the optimum if the function value at $\boldsymbol{x}_g^*$ coincides with the minimum function value obtained thus far by the algorithm. The only modification required to make this algorithm completely gradient-only is to report the solution of $\boldsymbol{x}_g^*$ instead of the minimum function value obtained along the search path.

We use the *ralg* implementation of the r-algorithm provided in the *OpenOpt* software library [10], which we denote by *R-ALG*.

4.3 Approximation methods

Approximation methods can also be formulated using only *gradient* information, e.g. see Groenwold *et al.* [8].

Let us consider approximation functions $\tilde{f}$ that use the *second order* Taylor series expansion of a function f around some current iterate $\boldsymbol{x}^{\{k\}}$, given by

$$\tilde{f}^{\{k\}}(\boldsymbol{x}) = f(\boldsymbol{x}^{\{k\}}) + {\nabla_A}^{\mathrm{T}} f(\boldsymbol{x}^{\{k\}})(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}})^{\mathrm{T}}\boldsymbol{H}^{\{k\}}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}),\ k = 0, 1, 2, \ldots \quad (4.4)$$

where superscript k represents an iteration number, $\tilde{f}$ the second order Taylor series approximation to f, ∇_A the *associated gradient* operator and $\boldsymbol{H}^{\{k\}}$ the Hessian. $f(\boldsymbol{x}^{\{k\}})$ and $\nabla_A f(\boldsymbol{x}^{\{k\}})$ respectively represent the function value and *associated gradient* vector at the current iterate $\boldsymbol{x}^{\{k\}}$. Generally speaking, approximation methods use only function value information in constructing $\boldsymbol{H}^{\{k\}}$ (due to the excessive computational effort associated with evaluating and storing $\boldsymbol{H}^{\{k\}}$ in the first place).

Consider for example a diagonal spherical quadratic approximation, with $\boldsymbol{H}^{\{k\}} = c^{\{k\}}\boldsymbol{I}$. The unknown $c^{\{k\}}$ can be obtained by enforcing $\tilde{f}^{\{k\}}(\boldsymbol{x}^{\{k-1\}}) = f(\boldsymbol{x}^{\{k-1\}})$, which results in

$$f(\boldsymbol{x}^{\{k-1\}}) = f(\boldsymbol{x}^{\{k\}}) + {\nabla_A}^{\mathrm{T}} f(\boldsymbol{x}^{\{k\}})(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}) + \frac{c^{\{k\}}}{2}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})^{\mathrm{T}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}), \quad (4.5)$$

e.g. see Snyman and Hay [20]. The scalar $c^{\{k\}}$ is then obtained as

$$c^{\{k\}} = 2\frac{f(\boldsymbol{x}^{\{k-1\}}) - f(\boldsymbol{x}^{\{k\}})}{(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})^{\mathrm{T}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})} - 2\frac{{\nabla_A}^{\mathrm{T}} f(\boldsymbol{x}^{\{k\}})(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})}{(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})^{\mathrm{T}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})}. \quad (4.6)$$

Approximations solely based on gradient information may be constructed by taking the derivative of (4.4), which gives

$$\nabla \tilde{f}^{\{k\}}(\boldsymbol{x}) = \nabla_A f(\boldsymbol{x}^{\{k\}}) + \boldsymbol{H}^{\{k\}}(\boldsymbol{x} - \boldsymbol{x}^{\{k\}}), \quad k = 0, 1, 2, \ldots \quad (4.7)$$

Note that at $\boldsymbol{x} = \boldsymbol{x}^{\{k\}}$, the *associated gradient* of the function $f(\boldsymbol{x})$ exactly matches the gradient of the approximation function $\tilde{f}(\boldsymbol{x})$. We write gradient instead of *associated gradient* of the approximation function to emphasize the differentiability of the approximation function. The Hessian $\boldsymbol{H}^{\{k\}}$ of the approximation $\tilde{f}$ is chosen to match some additional condition. Let us again consider a spherical quadratic approximation, with $\boldsymbol{H}^{\{k\}} = c^{\{k\}}\boldsymbol{I}$. Then, $c^{\{k\}}$ may be obtained by matching the gradient vectors at $\boldsymbol{x}^{\{k-1\}}$. Since only a single free parameter $c^{\{k\}}$ is available, the n components of the respective gradient vectors can (for example) be matched in a least square sense.

The least squares error is given by

$$E^{\{k\}} = (\nabla \tilde{f}^{\{k\}}(\boldsymbol{x}^{\{k-1\}}) - \nabla_A f(\boldsymbol{x}^{\{k-1\}}))^{\mathrm{T}}(\nabla \tilde{f}^{\{k\}}(\boldsymbol{x}^{\{k-1\}}) - \nabla_A f(\boldsymbol{x}^{\{k-1\}})). \quad (4.8)$$

After substitution of $\nabla_A \tilde{f}^{\{k\}}(\boldsymbol{x}^{\{k-1\}}) = \nabla_A f(\boldsymbol{x}^{\{k\}}) + c^{\{k\}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})$, we have

$$E^{\{k\}} = (\nabla_A f(\boldsymbol{x}^{\{k\}}) + c^{\{k\}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}) - \nabla_A f(\boldsymbol{x}^{\{k-1\}}))^{\mathrm{T}} (\nabla_A f(\boldsymbol{x}^{\{k\}}) + c^{\{k\}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}) - \nabla_A f(\boldsymbol{x}^{\{k-1\}})). \quad (4.9)$$

Minimization of the least squares error $E^{\{k\}}$ w.r.t. $c^{\{k\}}$ then gives

$$\frac{dE^{\{k\}}}{dc^{\{k\}}} = (\nabla_A f(\boldsymbol{x}^{\{k\}}) + c^{\{k\}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}) - \nabla_A f(\boldsymbol{x}^{\{k-1\}}))^{\mathrm{T}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}) + (\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})^{\mathrm{T}}(\nabla_A f(\boldsymbol{x}^{\{k\}}) + c^{\{k\}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}}) - \nabla_A f(\boldsymbol{x}^{\{k-1\}})) = 0, \quad (4.10)$$

hence

$$c^{\{k\}} = \frac{(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})^{\mathrm{T}}(\nabla_A f(\boldsymbol{x}^{\{k-1\}}) - \nabla_A f(\boldsymbol{x}^{\{k\}}))}{(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})^{\mathrm{T}}(\boldsymbol{x}^{\{k-1\}} - \boldsymbol{x}^{\{k\}})}. \quad (4.11)$$

If the approximation is required to be strictly convex, we can enforce $c^{\{k\}} = \max(\beta, c^{\{k\}})$, with $\beta > 0$ small and prescribed.

Since the sequential *approximate* subproblems are smooth, they may be solved analytically; the minimizer (or gradient projection point) of subproblem k follows from setting (4.7) equal to $\mathbf{0}$ [19], to give

$$\boldsymbol{x}^{\{k*\}} = \boldsymbol{x}^{\{k\}} - \frac{\nabla_A f(\boldsymbol{x}^{\{k\}})}{c^{\{k\}}}. \quad (4.12)$$

4.4 Conservative approximations

Global convergence of sequential approximation methods may for example be affected through the notion of conservatism. Classical conservatism is based solely on function values, for which Svanberg [21] demonstrated that an approximation sequence $k = 1, 2, \cdots$ will terminate at the global minimizer $\boldsymbol{x}^* \leftrightarrow f^*$, if each k^{th} approximation $\tilde{f}(\boldsymbol{x}^{\{k*\}})$ is conservative, i.e. if

$$\tilde{f}(\boldsymbol{x}^{\{k*\}}) \geq f(\boldsymbol{x}^{\{k*\}}) \ \forall \ k. \tag{4.13}$$

A mechanism similar to conservatism may also be affected using only *associated gradient* information; we will simply refer to this as conservatism, albeit that the description is possibly not completely apt. At iterate $\boldsymbol{x}^{\{k*\}}$, the update is given by $\boldsymbol{x}^{\{k*\}} - \boldsymbol{x}^{\{k\}}$, and conservatism is affected if the projection of the *associated gradient* $\nabla_A f(\boldsymbol{x}^{\{k*\}})$ of the actual function $f(\boldsymbol{x})$ onto the update direction $\boldsymbol{x}^{\{k*\}} - \boldsymbol{x}^{\{k\}}$ is negative. For univariate functions, an update is conservative if it is an associated derivative descent update step (see Definition 3.8). For multivariate functions an update is conservative if it is a conservative associated derivative descent update step (see Definition 3.9), i.e. if

$$\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}^{\{k*\}})(\boldsymbol{x}^{\{k*\}} - \boldsymbol{x}^{\{k\}}) < 0. \tag{4.14}$$

Hence, enforcement of the conditions given by Definition 3.8 or 3.9 suffices to ensure a sequence of derivative descent sequences for which proofs of convergence are offered in Appendix A. To allow for update steps that are computable we employ a trust region strategy where we limit $\|\boldsymbol{x}^* - \boldsymbol{x}^{\{k\}}\| \leq \gamma$.

4.4.1 Algorithmic implementation

Given an initial point $\boldsymbol{x}^{\{0\}}$, a {gradient-only}/classical conservative algorithm based on convex separable spherical quadratic approximations (SSA) for unconstrained gradient-only optimization problems proceeds as follows:

1. **Initialization:** Select real constants $\epsilon > 0$, $\alpha > 1$ and initial curvature $c^{\{0\}} > 0$. Set $k := 0$, $l := 0$.
2. **Gradient evaluation:** Compute $\{\nabla_A f(\boldsymbol{x}^{\{k\}})\}/f(\boldsymbol{x}^{\{k\}})$ and $\nabla_A f(\boldsymbol{x}^{\{k\}})$.
3. **Approximate optimization:** Construct local approximate subproblem $\{(4.7)\}/(4.4)$ at $\boldsymbol{x}^{\{k\}}$, using $\{(4.11)\}/(4.6)$ unless inside an inner loop then use $\boldsymbol{c}^{\{k\}}$ as calculated in Step 7(b). Solve this subproblem analytically, to arrive at $\boldsymbol{x}^{\{k*\}}$.
4. **Trust region:** If the $\|\boldsymbol{x}^* - \boldsymbol{x}^{\{k\}}\| > \gamma$ then $\boldsymbol{x}^* = -\gamma \frac{\nabla_A f(\boldsymbol{x}^{\{k*\}})}{\|\nabla_A f(\boldsymbol{x}^{\{k*\}})\|}$ and $c^{\{k\}} = \frac{\|\nabla_A f(\boldsymbol{x}^{\{k*\}})\|}{\gamma}$.
5. **Evaluation:** Compute $\{\nabla_A f(\boldsymbol{x}^{\{k*\}})\}/f(\boldsymbol{x}^{\{k*\}})$.
6. **Test if $\boldsymbol{x}^{\{k*\}}$ is acceptable:** if $\{(4.14)\}/(4.13)$ is satisfied, goto Step 8.
7. **Initiate an inner loop to effect conservatism:**
   (a) Set $l := l + 1$.

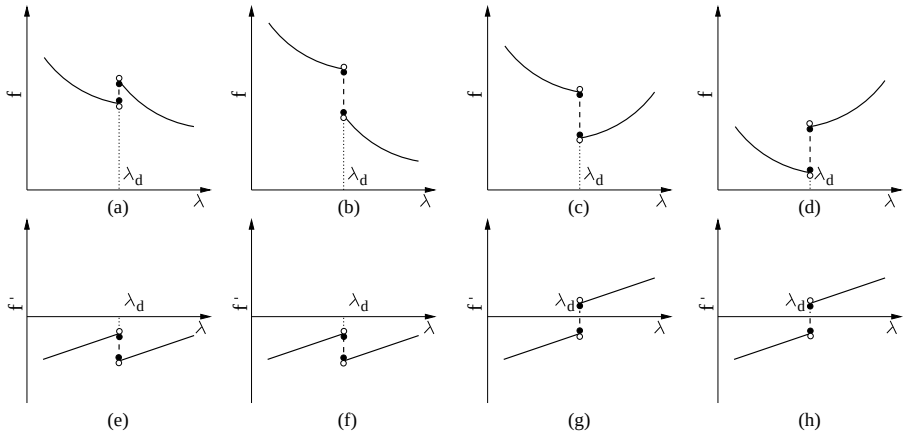


Fig. 5.1: Plots depicting (a)-(d) the function values, and (e)-(h) the corresponding derivatives of four instances of step-discontinuous univariate functions.

(b) Set $c^{\{k\}} := \alpha c^{\{k\}}$.
(c) Goto Step 3.

8. **Move to the new iterate:** Set $\boldsymbol{x}^{\{k+1\}} := \boldsymbol{x}^{\{k*\}}$.
9. **Convergence test:** if $\|\boldsymbol{x}^{\{k+1\}} - \boldsymbol{x}^{\{k\}}\| \leq \epsilon$, OR $k = k_{max}$, stop.
10. **Initiate an additional outer loop:** Set $k := k + 1$ and goto Step 2.

4.5 Termination criteria

Termination criteria also need some consideration: if the function values and *associated gradients* of an objective or cost function contain step discontinuities, these quantities may not provide robust termination information. Accordingly, we only advocate the robust termination criterion

$$\|\Delta\boldsymbol{x}^{\{k+1\}}\| = \|\boldsymbol{x}^{\{k+1\}} - \boldsymbol{x}^{\{k\}}\| < \epsilon, \tag{4.15}$$

with ϵ small, positive and prescribed. (A maximum number of iterations may of course also be prescribed, but this is not robust.)

5 Mathematical programming vs. gradient-only optimization

We now briefly reflect on some differences between gradient-only optimization and classical 'mathematical programming'. Consider the step discontinuities depicted in Figure 5.1.

In classical mathematical programming, the inconsistent step discontinuity depicted in Figure 5.1(a) result in a local minimum, whereas the function with the consistent step discontinuity depicted in Figure 5.1(b) is monotonically

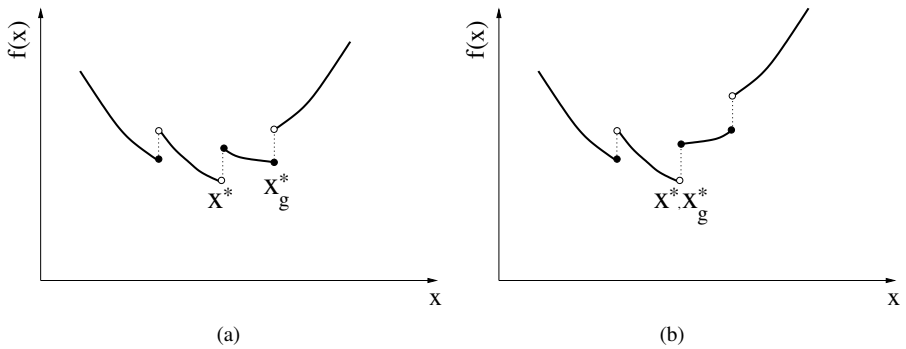


Fig. 5.2: Plots depicting a step discontinuous objective function with a (a) distinct minimizer x^* and gradient projection point (GPP) x_g^* and (b) coinciding minimizer x^* and GPP x_g^*.

decreasing. The step discontinuities depicted in Figures 5.1(c)-(d) again result in local minima.

In gradient-only optimization, the inconsistent step discontinuity in Figure 5.1(a) is derivative negative, as is the consistent step discontinuity depicted in Figure 5.1(b). The step discontinuities depicted in Figures 5.1(c)-(d) represent gradient projection points.

Consider the objective functions depicted in Figure 5.2 (b). Clearly, classical optimization approaches may get stuck in local minima caused by inconsistent step discontinuities, whereas gradient-only optimization approaches will not. Hence, gradient-only optimization allows for a robust strategy to avoid inconsistent step discontinuities when the minimizer x^* of an objective function coincides with a gradient projection point (GPP) x_g^* as shown in Figure 5.2 (b).

However, gradient-only approaches will ignore a global minimizer x_f^* of an objective function that occurs over an inconsistent step discontinuity as depicted in Figure 5.2 (a) and converge to a GPP x_g^*. It is important to note that gradient-only approaches used to minimize functions e.g. Shor's r-algorithm [16] will report the lowest function value evaluated along the search path but will converge to a positive projection point x_g^*. For example in Figure 5.2 (a) the reported optimum by such methods will depend on the initial starting point whereas the point of convergence will be consistently the gradient projection point x_g^* irrespective of the starting point. Whether, x^* or x_g^* is to be considered will depend on which point best describes or approximates the solution to the optimization problem.

Although no formal estimates of efficiency for the gradient-only optimization algorithms are attempted in this study. We note that the efficiency of gradient-only optimization algorithms will be similar to their classical gradient based counterparts provided that the gradient computation is efficient.

Table 6.1: Algorithmic settings used in the numerical experiment.

ϵ	c_1	c_2	α	γ	β	k_{max}	l_{max}
10^{-5}	10^{-4}	0.9	10^{-3}	2	2	3000	3000

Table 6.2: Tabulated results obtained for the unconstrained Michell-like structure.

Algorithm	$f(\boldsymbol{x}^{\{N_k\}})$	$\|\boldsymbol{\nabla} f(\boldsymbol{x}^{\{N_k\}})\|$	$\|\Delta\boldsymbol{x}^{\{N_k\}}\|$	N_{k}	N_{l}
BFGS(f)	5.263E-01	1.49E-03	2.404E-07	49	92
BFGS(g)	5.260E-01	1.66E-03	6.599E-06	61	69
R-ALG(g)†	5.211E-01	1.49E-03	8.83E-03	104	114
SSA(f)	5.805E-01	1.37E-2	6.692E-06	18	66
SSA(g)	5.269E-01	1.62E-03	8.233E-06	75	203

† The r-algorithm was unable to converge solely on the update termination criteria. In addition we allowed a convergence criteria based on the gradient norm being less than 10^{-6}.

6 Numerical study

We start our numerical study with a practical shape optimization problem using a remeshing strategy that results in a discontinuous objective function, additional problems are presented in [23]. We then proceed with a set of discontinuous test functions aimed to "mimic" non-physical discontinuities in functions. The advantage of introducing a set of test problems is that they are easily implemented which allows for focused research on algorithm development and testing, without requiring access to a variable discretization PDE solver. The disadvantage of test problems in turn is that only part of the complexity of PDE based objective functions is captured.

The algorithmic settings used in the numerical study are presented in Table 6.1 for the algorithms outlined in Sections 4.1.1. The settings for the strong Wolfe condition (c_1 and c_2) are recommended when using the BFGS algorithm [11].

6.1 Shape optimization

We now consider the isotropic shape optimization problem outlined in Section 1.2. The results for the BFGS(f), BFGS(g), SSA(f) and SSA(g) algorithms are summarized in Table 6.2 with the respective final designs depicted in Figures 6.1 (a)-(e). Recall that the (f) postfix indicates classical function-value based algorithms, whereas the (g) postfix indicates gradient-only optimization algorithms. Presented in Table 6.2 are the function value $f(\boldsymbol{x}^{\{N_k\}})$, gradient norm $\|\boldsymbol{\nabla} f(\boldsymbol{x}^{\{N_k\}})\|$, convergence tolerance $\|\Delta\boldsymbol{x}^{\{N_k\}}\|$, number of outer iterations N_k as well as the number of inner iterations N_l.

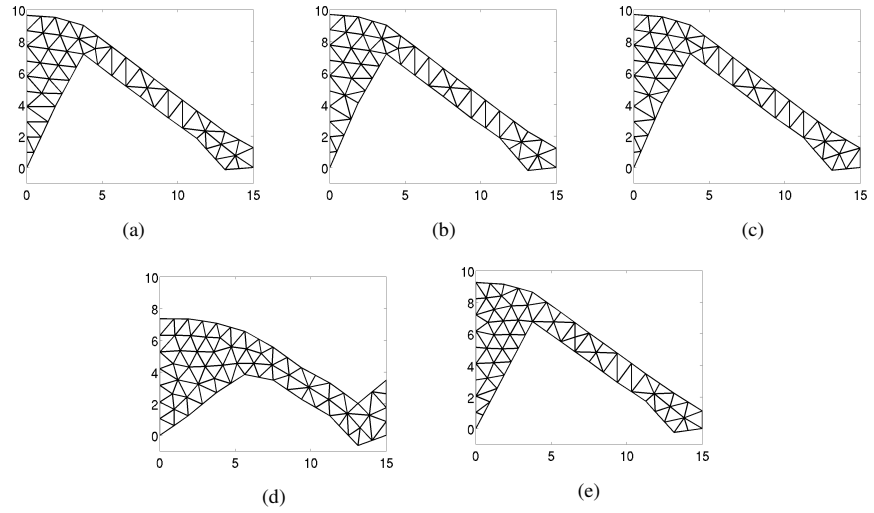


Fig. 6.1: Michell-like structure: converged designs obtained with (a) BFGS(f), (b) BFGS(g), (c) R-ALG(g), (d) SSA(f), and (e) SSA(g).

The BFGS(f), BFGS(g) and R-ALG(g) algorithms were able to converge to a solution which compares well with published literature. The BFGS(f) managed to do so in 49 outer iterations N_k whereas BFGS(g) and R-ALG(g) completed it in respectively 61 and 104 outer iterations. The BFGS(g) algorithm managed to converge in the smallest number of total iterations i.e. inner and outer iterations followed respectively by BFGS(f) and R-ALG(g). The final designs are represented in Figure 6.1 (a). Clearly, inexact line search methods significantly improve an algorithms ability to overcome step discontinuities, however, whether it is able to do so robustly remain a concern as illustrated on the test problems.

In turn, SSA(f) converged after 18 outer iterations N_k after getting trapped in a step discontinuous minimum. The behaviour of the cost function around the converged solution of SSA(f) is depicted in Figure 1.3 of Section 1.2. The premature converged design is evident from Figure 6.1 (d). Clearly, conservative approximation methods are able to overcome *some* step discontinuities and of course even more so when conservatism is relaxed.

Conversely, BFGS(f), BFGS(g), SSA(g) and R-ALG(g) were able to optimize the Michell structure without getting trapped in numerical induced step discontinuities. Consider the similar designs depicted in Figures 6.1 (a)-(c) and (e). It is clear that BFGS(f), BFGS(g), R-ALG(g) and SSA(g) improved notably on the designs obtained with SSA(f).

We further present for each algorithm their respective histories w.r.t. function value $f(\boldsymbol{x}^{\{k\}})$, gradient norm $\|\boldsymbol{\nabla} f(\boldsymbol{x}^{\{k\}})\|$ and convergence tolerance $\|\boldsymbol{\Delta x}^{\{k\}}\|$. The respective histories for the BFGS algorithms and R-ALG(g) are

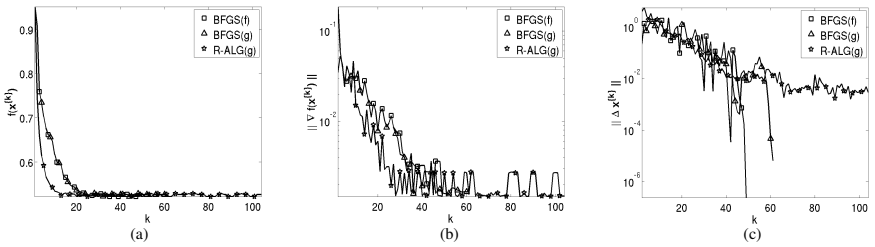


Fig. 6.2: Michell-like structure: BFGS(f), BFGS(g) and R-ALG(g) algorithms convergence history plot of the (a) function value $f(x^{\{k\}})$, (b) gradient norm $\|\nabla f(\boldsymbol{x}^{\{k\}})\|$, and (c) and convergence tolerance $\|\Delta \boldsymbol{x}^{\{k\}}\|$.

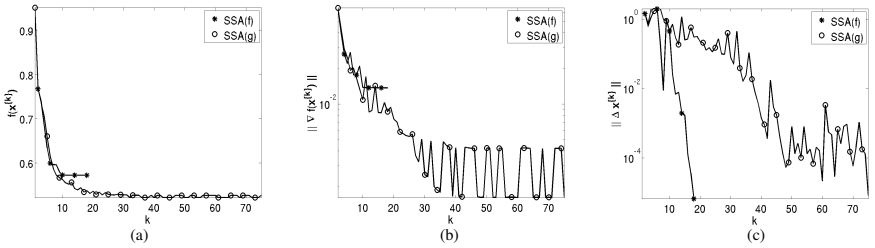


Fig. 6.3: Michell-like structure: SSA(f) and SSA(g) algorithms convergence history plot of the (a) function value $f(x^{\{k\}})$, (b) gradient norm $\|\nabla f(\boldsymbol{x}^{\{k\}})\|$, and (c) and convergence tolerance $\|\Delta \boldsymbol{x}^{\{k\}}\|$.

depicted in Figures 6.2 (a)-(c) and for the SSA algorithms in Figures 6.3 (a)-(c).

Monotonic function value decrease for SSA(f) is clearly depicted in respectively Figure 6.2(a) and Figure 6.3(a) with the respective associated gradient norms depicted in Figure 6.2(b) and Figure 6.3(b). The convergence histories are depicted in Figure 6.2(c) and Figure 6.3(c).

Conversely, non-monotonic function value decrease for BFGS(f), BFGS(g), R-ALG(g) and SSA(g) is evident in Figure 6.2(a) and Figure 6.3(a) with the respective associated gradient norms depicted in Figure 6.2(b) and Figure 6.3(b). The convergence histories are depicted in Figure 6.2(c) and Figure 6.3(c).

6.2 Analytical set of test problems

We now present a set of five analytical step discontinuous test problems in order to further illustrate the advantages of gradient-only optimization.

Rosenbrock step discontinuous function f_1 is piecewise defined as follows:

$$f_1(\boldsymbol{x}) = \begin{cases} \sum_{i=1}^{\frac{n}{2}} \frac{1}{1.2}\left(100\left(x(2i) - x^2(2i-1)\right)^2 + (1-x(2i-1))^2\right), \\ \qquad \text{if } 0 \le \sin(2\|\boldsymbol{x}\|) < \frac{2}{3}. \\ \sum_{i=1}^{\frac{n}{2}} 1.2\left(100\left(x(2i) - x^2(2i-1)\right)^2 + (1-x(2i-1))^2\right), \\ \qquad \text{if } -\frac{2}{3} \le \sin(2\|\boldsymbol{x}\|) < 0. \\ \sum_{i=1}^{\frac{n}{2}} \left(100\left(x(2i) - x^2(2i-1)\right)^2 + (1-x(2i-1))^2\right), \\ \qquad \text{if } -\frac{2}{3} > \sin(2\|\boldsymbol{x}\|) \ge \frac{2}{3}. \end{cases} \tag{6.1}$$

Quadric step discontinuous function f_2 is piecewise defined as follows:

$$f_2(\boldsymbol{x}) = \begin{cases} \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x(j)\right)^2, & \text{if } \sin(8\|\boldsymbol{x}\|) > 0.5. \\ \sum_{i=1}^{n} 1.2\left(\sum_{j=1}^{i} x(j)\right)^2, & \text{if } \sin(8\|\boldsymbol{x}\|) < -0.5. \\ \sum_{i=1}^{n} \frac{1}{1.2}\left(\sum_{j=1}^{i} x(j)\right)^2, & \text{if } -0.5 \le \sin(8\|\boldsymbol{x}\|) \le 0.5. \end{cases} \tag{6.2}$$

Sum squares step discontinuous function f_3 is piecewise defined as follows:

$$f_3(\boldsymbol{x}) = \begin{cases} \sum_{i=1}^{n} \frac{1}{1.5} i x^2(i), & \text{if } \sin\left(2\sum_{j=1}^{n} x(j)\right) > 0.5. \\ \sum_{i=1}^{n} 1.5 i x^2(i), & \text{if } \sin\left(2\sum_{j=1}^{n} x(j)\right) < -0.5. \\ \sum_{i=1}^{n} i x^2(i) + \frac{1}{n}, & \text{if } -0.5 \le \sin\left(2\sum_{j=1}^{n} x(j)\right) \le 0.5. \end{cases} \tag{6.3}$$

Zakharov step discontinuous function f_4 is piecewise defined as follows:

$$f_4(\boldsymbol{x}) = \begin{cases} \frac{1}{1.5}\sum_{i=1}^{n} x^2(i) + \left(\sum_{i=1}^{n} \frac{i x^2(i)}{2}\right)^2 + \left(\sum_{i=1}^{n} \frac{i x^2(i)}{2}\right)^4, \\ \qquad \text{if } \sin(\|\boldsymbol{x}\|) > 0.5. \\ 1.5\sum_{i=1}^{n} x^2(i) + \left(\sum_{i=1}^{n} \frac{i x^2(i)}{2}\right)^2 + \left(\sum_{i=1}^{n} \frac{i x^2(i)}{2}\right)^4 + 0.5, \\ \qquad \text{if } \sin(\|\boldsymbol{x}\|) < -0.5. \\ \sum_{i=1}^{n} x^2(i) + \left(\sum_{i=1}^{n} \frac{i x^2(i)}{2}\right)^2 + \left(\sum_{i=1}^{n} \frac{i x^2(i)}{2}\right)^4 + 1, \\ \qquad \text{if } -0.5 \le \sin(\|\boldsymbol{x}\|) \le 0.5. \end{cases} \tag{6.4}$$

Hyper ellipsoid step discontinuous function f_5 is piecewise defined as follows:

$$f_5(\boldsymbol{x}) = \begin{cases} \displaystyle\sum_{i=1}^{n} \frac{1}{1.1} 2^{i-1} x^2(i) + \frac{1}{n}, & \text{if } \sin\left(2\displaystyle\sum_{j=1}^{n} x(j)\right) > 0.5. \\ \displaystyle\sum_{i=1}^{n} 1.1 \times 2^{i-1} x^2(i) + \frac{1}{n}, & \text{if } \sin\left(2\displaystyle\sum_{j=1}^{n} x(j)\right) < 0, \\ \displaystyle\sum_{i=1}^{n} 2^{i-1} x^2(i), & \text{if } 0 \leq \sin\left(2\displaystyle\sum_{j=1}^{n} x(j)\right) \leq 0.5. \end{cases} \tag{6.5}$$

This set of step discontinuous test problems "mimics" functions that contain non-physical discontinuities. Our aim is to overcome the discontinuities to obtain $\boldsymbol{x}_g^*$ as outlined in Definition 3.7. The region around the solution of f_1, f_2 and f_4 is continuous as opposed to the region around the solution of f_3 and f_5 which are discontinuous. The solution of f_1 is given by $x^*(i) = 1,\ i = 1, 2, \ldots, n$ with $f_1^* = 0$ whereas the solution of f_2 and f_4 is given by $x_g^*(i) = 0,\ i = 1, 2, \ldots, n$ with $f_2^* = 0$ and $f_4^* = 1$ respectively. For f_3 and f_5 the derivative critical set S is defined by $\boldsymbol{x}_g^*$ where $x_g^*(i) = 0,\ i = 1, 2, \ldots, n$ with $f_3^* = \{0, 1\}$ and $f_5^* = \{0, 1\}$ respectively. The gradient field for each test function is given by the analytical gradient of each test function whereas the gradient at a discontinuous point is defined by the analytical gradient of the active equation of a test function at that point.

Results are presented for dimension $n = 10$ of the test problem set given in Section 6.2. The starting of each algorithm for each problem is $x(i)^{\{0\}} = 4,\ i = 1, 2, \ldots n$.

Numerical results are presented in Table 6.3. N_k and N_l respectively represent the number of function or gradient evaluations in the outer and inner loops. We have not limited the step size of the approximation algorithms; this is normally not done in algorithms based on conservatism (although it may sometimes be beneficial).

The results presented in Table 6.3 show that gradient-only optimization algorithms are able to robustly optimize step discontinuous objective functions. In contrast, the classical function-value based optimization algorithms converged to local minima on some of the problems. It is clear from Table 6.3 that the function-value based approximation algorithms BFGS(f) and SSA(f) are able to overcome many of the non-physical local minima. However, both BFGS(f) and SSA(f) do not represent a robust strategy as it still converged to local minima on some of the test problems.

The required number of inner and outer loops of the gradient-only implementations is also fewer when compared to their conventional function value based counterparts.

Table 6.3: Results for the step discontinuous test problem set.

Function	Solution	BFGS(f)	BFGS(g)	R-ALG	SSA(f)	SSA(g)
f_1	f^*	7.098E-09	2.40E-12	1.59E-10	5.09E-07	5.34E-08
	$\|\nabla f^*\|$	6.91E-05	3.40E-05	5.87E-04	5.83E-04	1.89E-04
	$\|\boldsymbol{x}^{\{N_k*\}} - \boldsymbol{x}_g^*\|$	5.34E-07	1.05E-07	1.48E-04	8.72E-07	2.26E-07
	N_k	35	30	98	235	106
	N_l	81	36	84	488	176
f_2	f^*	2.004E-15	2.00E-15	2.28E-10	2.26E+01	5.47E-10
	$\|\nabla f^*\|$	6.13E-08	6.13E-08	5.33E-05	2.99E+01	2.20E-05
	$\|\boldsymbol{x}^{\{N_k*\}} - \boldsymbol{x}_g^*\|$	5.22E-07	5.22E-07	7.93E-05	2.48E-07	3.02E-07
	N_k	48	48	74	20	132
	N_l	14	11	33	133	190
f_3	f^*	1.000E+00	1.05E-12	5.07E-10	4.43E-02	1.73E-11
	$\|\nabla f^*\|$	7.05E-03	3.61E-06	7.29E-05	7.02E-01	8.43E-06
	$\|\boldsymbol{x}^{\{N_k*\}} - \boldsymbol{x}_g^*\|$	1.06E-09	6.21E-07	4.87E-05	5.06E-07	8.13E-07
	N_k	20	20	80	25	33
	N_l	111	12	34	130	23
f_4	f^*	3.438E+01	1.00E+00	1.00E+00	8.97E+05	1.00E+00
	$\|\nabla f^*\|$	1.61E+01	2.08E-03	2.32E-03	1.41E+06	2.57E-03
	$\|\boldsymbol{x}^{\{N_k*\}} - \boldsymbol{x}_g^*\|$	1.51E-08	2.65E-05	9.25E-04	9.54E-07	1.68E-05
	N_k	19	26	60	18	82
	N_l	152	112	42	135	100
f_5	f^*	2.196E+00	4.86E-10	1.98E-10	1.94E+02	1.01E-08
	$\|\nabla f^*\|$	5.57E+00	7.95E-05	4.02E-04	9.48E+01	8.41E-04
	$\|\boldsymbol{x}^{\{N_k*\}} - \boldsymbol{x}_g^*\|$	1.85E-07	6.54E-07	6.91E-06	7.12E-07	9.24E-07
	N_k	51	95	89	25	231
	N_l	76	25	35	145	367

7 Conclusions

We have studied the unconstrained optimization of functions containing step or jump discontinuities. Although these functions may become discontinuous and non-differentiable we can compute exact gradient information where the function is differentiable and define approximate gradient information where it is non-differentiable. At a non-differential point a partial derivative of the gradient vector can be approximated by a one-sided directional derivative or by the partial derivative itself when the function is respectively non-differentiable or differentiable along the partial derivative direction. Step or jump discontinuities arises during the solution of systems of (partial) differential equations, when variable spatial and temporal discretization techniques produce discontinuities that are artifacts of the approximate numerical strategies used. While discontinuous, we demonstrate that these problems may effectively be optimized if only gradient information is used. Various algorithmic options were discussed and numerical results presented for a practical shape optimization problem as well as a set of analytical test functions.

The implications of our approach are that variable discretization strategies, which are so important in numerical discretization methods, may be used in

combination with efficient local optimization algorithms, notwithstanding the fact that these strategies themselves introduce step discontinuities.

Among others, future endeavors should in our opinion concentrate on the inclusion of constraints, and reduction of the required computational effort.

8 Acknowledgment

The first author gratefully acknowledges financial assistance from the National Research Foundation (NRF) of South Africa.

References

1. G. Allaire, F. Jouve, A.-M. Toader, Structural optimization using sensitivity analysis and a level-set method, J. Comput. Phys. 194 (1) (2004) 363–393.
2. M. S. Bazaraa, H. D. Sherali, C. M. Shetty, Nonlinear Programming - Theory and Algorithms, 2nd Edition, John Wiley & Sons, Inc., New York, NY, USA, 1993.
3. S. K. Berberian, A First Course in Real Analysis, Springer-Verlag, New York, NY, USA, 1994.
4. B. R. Brandstatter, W. Ring, C. Magele, K. R. Richter, Shape design with great geometrical deformations using continuously moving finite element nodes, IEEE Trans. Magn. 34 (5) (1998) 2877–2880.
5. A. R. Conn, M. Mongeau, Discontinuous piecewise linear optimization, Math. Program. 80 (1998) 315–380.
6. R. D. Cook, D. S. Malkus, M. E. Plesha, R. J. Witt, Concepts and Applications of Finite Element Analysis, 4th Edition, John Wiley & Sons, Inc., New York, NY, USA, 2002.
7. M. J. Garcia, C. A. Gonzalez, Shape optimisation of continuum structures via evolution strategies and fixed grid finite element analysis, Struct. Multidiscip. Optim. V26 (1) (2004) 92–98.
8. A. A. Groenwold, L. F. P. Etman, J. A. Snyman, J. E. Rooda, Incomplete series expansion for function approximation, Struct. Multidiscip. Optim. 34 (2007) 21–40.
9. S. Kodiyalam, P. B. Thanedar, Some practical aspects of shape optimization and its influence on intermediate mesh refinement, Finite Elem. Anal. Des. 15 (2) (1993) 125–133.
10. D. Kroshko, OpenOpt Free GNU GPL2 MATLAB/Octave optimization toolbox, version 0.36, http://openopt.org (2006).
11. J. Nocedal and S. J. Wright, Numerical Optimization, 2nd Edition, Operation Research and Financial Engineering, Springer, 2006.
12. N. Olhoff, J. Rasmussen, E. Lund, A method of exact numerical differentiation for error elimination in finite element based semi-analytical shape sensitivity analysis, Mech. Struct. Mech. 21 (1993) 1–66.
13. A. Peressini, F. Sullivan, J. Uhl, The Mathematics of Nonlinear Programming, Springer-Verlag, New York, NY, USA, 1988.
14. R. L. Rardin, Optimization in Operations Research, Prentice Hall Inc., Upper Saddle River, NJ, USA, 1998.
15. A. Schleupen, K. Maute and E. Ramm, Adaptive FE-procedures in shape optimization, Struct. Multidiscip. Optim. 4 (2000) 282–302.
16. N. Z. Shor, K. C. Kiwiel, A. Ruszcayński, Minimization methods for non-differentiable functions, Springer-Verlag New York, Inc., New York, NY, USA, 1985.
17. J. A. Snyman, A gradient-only line search method for the conjugate gradient method applied to constrained optimization problems with severe noise in the objective function, Internat. J. Numer. Methods Engrg. 62 (1) (2005) 72–82.

18. J. A. Snyman, Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms, 2nd Edition, Applied Optimization, Vol. 97, Springer-Verlag New York, Inc., 2005.
19. J. A. Snyman, A. M. Hay, The spherical quadratic steepest descent (sqsd) method for unconstrained minimization with no explicit line searches, Comput. Math. Appl. 42 (2001) 169–178.
20. J. A. Snyman, A. M. Hay, The Dynamic-Q optimization method: an alternative to SQP?, Comput. Math. Appl. 44 (2002) 1589–1598.
21. K. Svanberg, A class of globally convergent optimization methods based on conservative convex separable approximations, SIAM J. Optim. 12 (2002) 555–573.
22. L. Van Miegroet, N. Moës, C. Fleury, P. Duysinx, Generalized shape optimization based on the level set method, in: 6th Congr. Struct. Multidiscip. Optim., 2005, pp. 1–10, paper no. 711.
23. D. N. Wilke, S. Kok, A. A. Groenwold, The application of gradient-only optimization methods for problems discretized using non-constant methods, Struct. Multidiscip. Optim. 40 (2010) 433–451.
24. D. N. Wilke, S. Kok, A. A. Groenwold, A quadratically convergent unstructured remeshing strategy for shape optimization, Internat. J. Numer. Methods Engrg. 65 (1) (2006) 1–17.
25. I. Zang, Discontinuous optimization by smoothing, Math. Oper. Res. 6 (1):140-152, 1981.

A Proofs of convergence for derivative descent sequences

Before we present proofs of convergence of (conservative) associated derivative descent sequences we include two gradient-only definitions of the well-known concepts in classical mathematical programming to simplify our proofs of convergence. First, we present a definition of coercive functions based solely on the *associated gradient* of a function [13]. Although this definition does not bear a strict analogy with the conventional coercive definition it suffices for our purposes.

Definition A.1 Let $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathbb{R}^n$. Then a real valued function $f : X \subset \mathbb{R}^n \to \mathbb{R}$ with *associated gradient* field $\nabla_A f(\boldsymbol{x})$ that is uniquely defined for every $\boldsymbol{x} \in X$, is associated derivative coercive if there exist a positive number R_M such that ${\nabla_A}^{\mathrm{T}} f(\boldsymbol{x}^2)(\boldsymbol{x}^2 - \boldsymbol{x}^1) > \epsilon$ with $\epsilon > 0 \in \mathbb{R}$ for non perpendicular $\nabla_A f(\boldsymbol{x}^2)$ and $(\boldsymbol{x}^2 - \boldsymbol{x}^1)$, whenever $\|\boldsymbol{x}^2\| \geq R_M$ and $\|\boldsymbol{x}^1\| < R_M$.

Secondly, we present definitions for univariate and multivariate associated gradient unimodality based solely on the *associated gradient* field of a real valued function [2].

Definition A.2 A univariate function $f : X \subset \mathbb{R} \to \mathbb{R}$ with associated derivative ${f'}^A(\lambda)$ uniquely defined for every $\lambda \in X$, is (resp., strictly) associated derivative unimodal over X if there exists a $x_g^* \in X$ such that

$$ {f'}^A(x_g^* + \lambda u)u \geq (\text{resp., } >)\ 0,\ \forall\ \lambda \in \{\beta : \beta > 0 \text{ and } \beta \subset \mathbb{R}\} \\ \text{and}\ \ \forall\ u \in \{-1, 1\} \text{ such that } [x_g^* + \lambda u] \in X. \quad \text{(A.1)} $$

We now consider (resp., strictly) associated derivative unimodality for multivariate functions [14].

Definition A.3 A multivariate function $f : X \subset \mathbb{R}^n \to \mathbb{R}$ is (resp., strictly) associated derivative unimodal over X if for all $\boldsymbol{x}^1$ and $\boldsymbol{x}^2 \in X$ and $\boldsymbol{x}^1 \neq \boldsymbol{x}^2$, every corresponding univariate function

$$ F(\lambda) = f(\boldsymbol{x}^1 + \lambda(\boldsymbol{x}^2 - \boldsymbol{x}^1)),\ \ \lambda \in [0, 1] \subset \mathbb{R} $$

is (resp., strictly) associated derivative unimodal according to Definition A.2.

A.1 Univariate functions

Now that we have an *associated derivative* based definition of unimodality for univariate functions we present a proof of convergence for strict univariate associated derivative unimodal functions when associated derivative descent sequences are considered.

Theorem A.4 *Let $f : \Lambda \subseteq \mathbb{R} \rightarrow]-\infty, \infty]$ be a univariate function that is strictly associated derivative unimodal as defined in Definition A.2, with first* associated derivative $f'^{A} : \Lambda \rightarrow]-\infty, \infty[$ *uniquely defined everywhere on Λ. If $\lambda^{\{0\}} \in \Lambda$ and $\{\lambda^{\{k\}}\}$ is an associated derivative descent sequence, as defined in Definition 3.8, for f with initial point $\lambda^{\{0\}}$, then every subsequence of $\{\lambda^{\{k\}}\}$ converges. The limit of any convergent subsequence of $\{\lambda^{\{k\}}\}$ is a strict non-negative associated gradient projection point (S-NN-GPP), as defined in Definition 3.4, of f.*

Proof Our assertion that f is strict associated derivative unimodal as defined in Definition A.2 implies that f has only one S-NN-GPS $S_{S-NN} \subset \Lambda$ as defined in Definition 3.7 at $\lambda^* \in \Lambda$. Let $\lambda^r \in S_{S-NN}$ such that $|\lambda^{\{k\}} - \lambda^r|$ is a maximum. Consider a sequence of 1-balls $\{B(b_k, \epsilon_k)\}$ defined around $b_k = \frac{1}{2}(\lambda^{\{k\}} + \lambda^r)$ with radius of $\frac{1}{2}|\lambda^{\{k\}} - \lambda^r|$. Then every $\lambda^{\{k+1\}} \in B(b_k, \epsilon_k)$, since $\{\lambda^{\{k\}}\}$ is an associated derivative descent sequence as defined in Definition 3.8 and f is strict associated derivative unimodal as defined in Definition A.2. Therefore, $k \rightarrow \infty$ implies $|\lambda^{\{k\}} - \lambda^r| \rightarrow 0$. It follows from the Cauchy criterion for sequences that $\{\lambda^{\{k\}}\}$ is convergent, which completes the proof of our first assertion.

Now let $\{\lambda^{\{k\}_m}\}$ be a convergent subsequence of $\{\lambda^{\{k\}}\}$ and let λ^{m*} be its limit. Suppose, contrary to the second assertion of the theorem, that λ^{m*} is not a S-NN-GPP as defined in Definition 3.4 of f. Since we assume that λ^{m*} is not a S-NN-GPP, and by Definition 3.8, there exist a $\lambda^{m*} + \delta$ for $\delta \neq 0 \in \mathbb{R}$ such that $f'^{A}(\lambda^{m*} + \delta) < 0$, which contradicts our assumption that λ^{m*} is the limit of the subsequence $\{\lambda^{\{k\}_m}\}$. Therefore, for λ^{m*} to be the limit of an associated derivative descent subsequence $\{\lambda^{\{k\}_m}\}$, $\lambda^{m*} \in S_{S-NN}$, which completes the proof.

We now proceed with a proof of convergence for generalized univariate associated derivative unimodal functions when associated derivative descent sequences are considered.

Theorem A.5 *Let $f : \Lambda \subseteq \mathbb{R} \rightarrow]-\infty, \infty]$ be a univariate function that is associated derivative unimodal, as defined in Definition A.2, with first* associated derivative $f'^{A} : \Lambda \rightarrow]-\infty, \infty[$ *uniquely defined everywhere on Λ. If $\lambda^{\{0\}} \in \Lambda$ and $\{\lambda^{\{k\}}\}$ is an associated derivative descent sequence, as defined in Definition 3.8, for f with initial point $\lambda^{\{0\}}$, then every subsequence of $\{\lambda^{\{k\}}\}$ converges. The limit of any convergent subsequence of $\{\lambda^{\{k\}}\}$ is a generalized G-NN-GPP, as defined in Definition 3.2, of f.*

Proof Our assertion that f is associated derivative unimodal as defined in Definition A.2 implies that f has at least one G-NN-GPS $S_{G-NN} \in \Lambda$ as defined in Definition 3.7. Let $S \subset \Lambda$ be the union of G-NN-GPSs S_{G-NN}. Consider the j^{th} sequence of 1-balls $\{B(b_k, \epsilon_k)\}_j$ defined around $b_k = \frac{1}{2}(\lambda^{\{k\}} + (\lambda_j^* \in S))$ and with radius $\epsilon_k = \frac{1}{2}|\lambda^{\{k\}} - (\lambda_j^* \in S)|$. Then $\lambda^{\{k+1\}} \in B(b_k, \epsilon_k)_j$ for every sequence j since $\{\lambda^{\{k\}}\}$ is a associated derivative descent sequence as defined in Definition 3.8 and f is associated derivative unimodal as defined in Definition A.2. Therefore $k \rightarrow \infty$ implies $|\lambda^{\{k\}} - (\lambda_j^* \in S)| \rightarrow a_j$ with a_j a constant. Since $|\lambda^{\{k\}} - (\lambda_j^* \in S)| - a_j \rightarrow 0$ for every j it follows from the Cauchy criterion for sequences that $\{\lambda^{\{k\}}\}$ is convergent, which completes the proof of our first assertion.

Now let $\{\lambda^{\{k\}_m}\}$ be a convergent subsequence of $\{\lambda^{\{k\}}\}$ and let λ^{m*} be its limit. Suppose, contrary to the second assertion of the theorem, that λ^{m*} is not a G-NN-GPP as defined in Definition 3.2 of f. Since we assume that λ^{m*} is not a G-NN-GPP, and by Definition 3.8, there exist a $\lambda^{m*} + \delta$ for $\delta \neq 0 \in \mathbb{R}$ such that $f'^{A}(\lambda^{m*} + \delta) < 0$ which contradicts our assumption that λ^{m*} is the limit of the subsequence $\{\lambda^{\{k\}_m}\}$. Therefore, for λ^{m*} to be the limit of an associated derivative descent subsequence (see Definition 3.8) $\{\lambda^{\{k\}_m}\}$, $\lambda^{m*} \in S$, which completes the proof.

Now that we have concluded our proofs of (strictly) associated derivative unimodal univariate functions, we present a proof of convergence for univariate associated derivative coercive functions that have at least one S-NN-GPS.

Theorem A.6 *Let $f : \Lambda \subseteq \mathbb{R} \to]-\infty, \infty]$ be a univariate associated derivative coercive function, as defined in Definition A.1, with first* associated derivative *$f'^{A} : \Lambda \to]-\infty, \infty[$ uniquely defined everywhere on Λ. If $\lambda^{\{0\}} \in \Lambda$ and $\{\lambda^{\{k\}}\}$ is an associated derivative descent sequence, as defined in Definition 3.8, for f with initial point $\lambda^{\{0\}}$, then there exists at least one convergent subsequence of $\{\lambda^{\{k\}}\}$. The limit of any convergent subsequence of $\{\lambda^{\{k\}}\}$ is a S-NN-GPP of f.*

Proof Since we only consider associated derivative descent sequences $\{\lambda^{\{k\}}\}$ our assertion that f is associated derivative coercive implies the closed interval $[a, b] \subset \Lambda$. The sequence $\{\lambda^{\{k\}}\}$ is bounded which follows from our premise of f. It follows from the Weierstrass-Bolzano theorem that in a closed interval $[a, b]$, every sequence has a subsequence that converges to a point in the interval [3].

Now let $\{\lambda^{\{k\}_m}\}$ be a convergent subsequence of $\{\lambda^{\{k\}}\}$ and let $\lambda^{m*} \in \Lambda$ be its limit. Suppose, contrary to the second assertion of the theorem, that λ^{m*} is not a S-NN-GPP of f. Since we assume that λ^{m*} is not a S-NN-GPP, and by Definition 3.8, there exist a $\lambda^{m*} + \delta$ for $\delta \neq 0 \in \mathbb{R}$ such that $f'^{A}(\lambda^{m*} + \delta) < 0$, which contradicts our assumption that λ^{m*} is the limit of the subsequence $\{\lambda^{\{k\}_m}\}$. Therefore, for λ^{m*} to be the limit of an associated derivative descent sequence (see Definition 3.8) $\{\lambda^{\{k\}_m}\}$, $\lambda^{m*} \in S_{S-NN}$ with $S_{S-NN} \subset \Lambda$ which completes the proof.

A.2 Multivariate functions

We begin our proof of convergence of associated derivative descent sequences for multivariate functions with C^1 continuous convex functions [13], whereupon we present proofs of convergence for broader classes of functions.

Theorem A.7 *Suppose $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is a C^1 continuous convex function with $\boldsymbol{x} \in X$. If $\boldsymbol{x}^{\{0\}} \in X$ and $\{\boldsymbol{x}^{\{k\}}\}$ is an associated derivative descent sequence, as defined in Definition 3.8, for f with initial point $\boldsymbol{x}^{\{0\}}$, then every subsequence of $\{\boldsymbol{x}^{\{k\}}\}$ converges. The limit of any convergent sequence of $\{\boldsymbol{x}^{\{k\}}\}$ is a S-NN-GPP as defined in Definition 3.4 of f.*

Proof Our assertion that f is convex and C^1 continuous ensures that f has a single global gradient projection point $\boldsymbol{x}_g^* \in X$. Also, by Definition 3.8 and the continuity of the first partial derivatives, we see that $\{f(\boldsymbol{x}^{\{k\}})\}$ is a decreasing sequence that is bounded below by $f(\boldsymbol{x}_g^*)$. It follows that $\{\boldsymbol{x}^{\{k\}}\}$ is a bounded sequence since f is convex. The Bolzano-Weierstrass theorem implies that $\{\boldsymbol{x}^{\{k\}}\}$ has at least one convergent subsequence, which completes the proof of our first assertion [13].

Now let $\{\boldsymbol{x}^{\{k\}_m}\}$ be a convergent subsequence of $\{\boldsymbol{x}^{\{k\}}\}$ and let $\boldsymbol{x}^{m*} \in X$ be its limit. Suppose, contrary to the second assertion of the theorem, that $\boldsymbol{x}^{m*}$ is not a S-NN-GPP as defined in Definition 3.4 of f which from our continuity assumption implies $\nabla_A f(\boldsymbol{x}^{m*}) \neq \mathbf{0}$, which in turn implies that there exists a descent direction $\boldsymbol{u}^{m*}$ at $\boldsymbol{x}^{m*}$, such that $\boldsymbol{u}^{m*} \neq \mathbf{0}$.

Since $\{\boldsymbol{x}^{\{k\}_m}\}$ is an associated derivative descent sequence as defined in Definition 3.8 of which the limit $\boldsymbol{x}^{m*}$ is by assumption not a S-NN-GPP i.e.

$$-\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}^{m*}) \nabla_A f(\boldsymbol{x}^{m*}) < 0.$$

It follows from the continuity assumptions that there exists a small $\lambda > 0 \in R$ such that $-\nabla_A{}^{\mathrm{T}} f(\boldsymbol{x}^{m*} + \lambda \boldsymbol{u}^{m*}) \nabla_A f(\boldsymbol{x}^{m*}) < 0$ which contradicts our assumption that $\boldsymbol{x}^{m*}$ is the limit of the sequence $\{\boldsymbol{x}^{\{km\}}\}$. Therefore, for $\boldsymbol{x}^{*m}$ to be the limit of an associated derivative descent sequence $\{\boldsymbol{x}^{\{km\}}\}$, $\nabla_A f(\boldsymbol{x}^{m*}) = \mathbf{0}$, which in turn implies $\boldsymbol{u}^{m*} = \mathbf{0}$. The limit $\boldsymbol{x}^{*m}$ of an associated derivative descent sequence as defined in Definition 3.8, is therefore a S-NN-GPP as defined in Definition 3.4, which completes the proof.

Before we proceed to present a proof of convergence for C^1 continuous associated derivative coercive functions, we show that if a function is associated derivative coercive and C^1 continuous it has at least one global gradient projection point.

Proposition A.8 *Suppose $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is a C^1 continuous associated derivative coercive function as defined in Definition A.1 with $\boldsymbol{x} \in X$, then f has at least one S-NN-GPP as defined in Definition 3.4.*

Proof Let $\boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3 \in \mathbb{R}^n$. Since f is associated derivative coercive as defined in Definition A.1, there exists by definition a number R_M such that for every $\{\boldsymbol{x}^2 : \|\boldsymbol{x}^2\| > R_M\}$, and every $\{\boldsymbol{x}^1 : \|\boldsymbol{x}^1\| < R_M\}$, the following holds: ${\nabla_A}^{\mathrm{T}} f(\boldsymbol{x}^2)(\boldsymbol{x}^2 - \boldsymbol{x}^1) > 0$, for non perpendicular $\nabla_A f(\boldsymbol{x}^2)$ and $(\boldsymbol{x}^2 - \boldsymbol{x}^1)$. In addition, there exists $\{\boldsymbol{x}^3 : \|\boldsymbol{x}^3\| < R_M\}$, such that ${\nabla_A}^{\mathrm{T}} f(\boldsymbol{x}^3)(\boldsymbol{x}^3 - \boldsymbol{x}^1) > 0$. Therefore, the set $\{\boldsymbol{x} : \|\boldsymbol{x}\| < R_M\}$ is closed and bounded, which by the continuity assumption implies that $f(\boldsymbol{x})$ assumes a minimum value on $\{\boldsymbol{x} : \|\boldsymbol{x}\| < R_M\}$ at a point $\boldsymbol{x}_g^* \in X$. From the continuity assumption of the first *partial associated derivatives*, it follows that $\nabla_A f(\boldsymbol{x}_g^*) = \boldsymbol{0}$ [13]. It therefore follows from the continuity assumptions that Definition 3.4 holds at $\boldsymbol{x}_g^*$.

Theorem A.9 *Suppose $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$ is a C^1 continuous associated derivative coercive function, as defined in Definition A.1, with $\boldsymbol{x} \in X$. If $\boldsymbol{x}^{\{0\}} \in X$, and $\{\boldsymbol{x}^{\{k\}}\}$ is a conservative associated derivative descent sequence, as defined in Definition 3.9, for f with initial point $\boldsymbol{x}^{\{0\}}$, then some subsequence of $\{\boldsymbol{x}^{\{k\}}\}$ converges. The limit of any convergent sequence of $\{\boldsymbol{x}^{\{k\}}\}$ is a G-NN-GPP, as defined in Definition 3.2, of f.*

Proof Our assertion that f is continuous and associated derivative coercive ensures that f has a global minimizer $\boldsymbol{x}_g^* \in X$. Also, by the definition of a conservative associated derivative descent sequence and the continuity of the first *partial associated derivatives*, we see that $\{f(\boldsymbol{x}^{\{k\}})\}$ is a decreasing sequence that is bounded below by $f(\boldsymbol{x}_g^*)$. Note that we require *conservative associated derivative* descent sequences, since *derivative* descent sequence is not sufficient to guarantee convergence as it may result in oscillatory behavior for $n > 1$. The remainder of the proof is similar to the proof of Theorem A.7.

We now proceed to functions that are either C^0 continuous or discontinuous, but for which the function values and *associated gradient* field are uniquely defined everywhere. We present classes of C^0 continuous or discontinuous functions for which convergence is guaranteed, since associated derivative descent sequences may not converge to NN-GPP when all C^0 continuous or discontinuous functions are considered, as is evident from the following example.

Consider the linear programming problem of finding the intersection between two intersecting planes. Since the *associated gradient* on each plane is constant, a steepest descent sequence that terminates at the intersection of the two planes is an example of a sequence that converges to some point that is not a NN-GPP.

Hence, we now present classes of well-posed discontinuous functions for which convergence is guaranteed.

Definition A.10 We consider the (resp. generalized / strict) gradient-only optimization problem to be well-posed (resp. convex / unimodal) associated derivative when

1. the *associated gradient* field is everywhere uniquely defined,
2. the problem is associated derivative coercive as defined in Definition A.1,
3. there exits one and only one (resp. G/S)-NN-GPS (resp. S_{G-NN} / S_{S-NN}) as defined in Definition 3.7, and
4. when every associated derivative descent sequence as defined in Definition 3.8 has at least one converging subsequence to a point in (resp. S_{G-NN} / S_{S-NN}).

We now present a class of well-posed associated derivative coercive functions; this includes multimodal functions.

Definition A.11 We consider the gradient-only optimization problem to be (resp. proper / generalized) well-posed associated derivative coercive when

1. the *associated gradient* field is everywhere uniquely defined,
2. the problem is associated derivative coercive as defined in Definition A.1,
3. there exits at least one (resp. G/S)-NN-GPS (resp. S_{G-NN} / S_{S-NN}) as defined in Definition 3.7, and
4. when every conservative associated derivative descent sequence as defined in Definition 3.9 has at least one converging subsequence to a point in (resp. S_{G-NN} / S_{S-NN}).

We note that the classes of functions defined in Definitions A.10 - A.11 still exclude many problems of practical significance e.g. linear programming problems. Many of these practically significant problems may be accommodated by altering Definitions A.10 - A.11 to hold only for specific associated derivative descent sequences.