A SUBSPACE METHOD BASED ON A DIFFERENTIAL EQUATION APPROACH TO SOLVE UNCONSTRAINED OPTIMIZATION PROBLEMS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF

MANAGEMENT SCIENCE AND ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ENGINEERING-ECONOMIC SYSTEMS AND OPERATIONS RESEARCH

Antonino Del Gatto

June 2000

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Walter Murray, Principal Adviser

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Richard W. Cottle

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

B. Curtis Eaves

Approved for the University Committee on Graduate Studies:

To Ernesto:
my son, my father and
my great-grandfather.

Abstract

This dissertation investigates the use of ordinary differential equations (ODEs) to determine a search path to find a solution of an unconstrained optimization problem. Two strategies, linesearch and trust-region, are the basis of current optimization algorithms. They are affected either by the definiteness of the Hessian matrix at any iteration or by the scaling of the optimization problem. A third strategy to solve unconstrained optimization problems is the method of gradients. It was originally proposed in 1941 by Courant as a variational method for numerically solving partial differential equations (PDEs) arising in problems of equilibrium and vibrations. While historically much of the motivation for the method of gradients and the mathematical analysis behind it came from the desire to solve PDEs, the function it seeks to minimize need not come from a PDE. Indeed, the method of gradients is a general-purpose method to solve unconstrained optimization problems. Its convergence theory requires knowledge of the objective function at a continuum of points. The method has not found general acceptance because of the difficulties inherent in solving a system of nonlinear ODEs. Recently, Behrman proposed a method of gradients based upon the solution of a system of n linear ODEs, where n denotes the number of independent variables of the objective function. Here we propose a method similar in spirit to the original method of gradients, but one that requires the solution of a system of only two linear ODEs. These equations are definable and solvable regardless of the nature of the Hessian matrix, and their solutions are unaffected by poor scaling. We show that the new algorithm converges to a point satisfying the first and second-order necessary conditions for local unconstrained optimality. This approach is also efficient for large-scale problems. Numerical testing was performed on a Silicon Graphics Origin 2000 running IRIX64 release 6.5.6m.

Acknowledgments

I would like to express my deep gratitude towards my principal adviser, Professor Walter Murray, for his consistent guidance, support, inspiration and enthusiasm since my first days at Stanford.

My gratitude also extends to Professors Richard W. Cottle and B. Curtis Eaves for their critical and insightful comments.

I would like to thank my colleagues Angel-Victor de Miguel Campos, Michael Paul Friedlander, Kien-Ming Ng and Heiko Pieper for the numerous and stimulating discussions that were not all about optimization.

Special acknowledgments go to "il gomitolo d'imbecilli": these people have made my stay at Stanford a pleasant and enjoyable one.

I am greatly indebted to Stefania for her steadfast and everlasting companionship and support throughout my years at Stanford. Without her encouragement and sacrifice I could not have completed this quest. I will always appreciate her understanding of times when research consumed my days and nights.

Finally and most importantly I would like to express my deepest gratitude and love to my parents who never had any doubts that I would succeed.

Ad maiora!

Contents

List of Figures

Chapter 1

INTRODUCTION

Usually the solution of a difficult problem in analysis proceeds according to a general scheme: the given problem $\mathcal{P}$ with the solution $\mathcal{S}$ is replaced by a related problem $\mathcal{P}_n$ so simple that its solution $\mathcal{S}_n$ can be found with comparative ease. Then by improving the approximation $\mathcal{P}_n$ to $\mathcal{P}$ we may expect, or we may assume, or we may prove, that $\mathcal{S}_n$ tends to the desired solution $\mathcal{S}$ of $\mathcal{P}$. The essential point in an individual case is to choose the sequence $\mathcal{P}_n$ in a suitable manner (Richard Courant [19]).

1.1 Unconstrained optimization

We consider the nonlinear unconstrained optimization problem

$$\underset{x \in D}{\text{minimize}} \quad F(x), \tag{1}$$

where $F(x)$ is a twice-continuously differentiable real-valued function defined on an open set $D \subseteq \mathbf{R}^n$. By Schwarz's Theorem, the Hessian matrix $\nabla^2 F(x)$ is symmetric for all $x \in D$. All algorithms for unconstrained minimization require the user to supply or to generate randomly a starting point, which is usually denoted by x_0. Beginning at $x_0 \in D$, the aim is to generate a sequence $(x_k)_{k=0}^{+\infty}$ of iterates in D that converges to a point $x^* \in D$ satisfying the first and second-order necessary conditions for a local multivariate unconstrained minimum, that is, $\nabla F(x^*) = 0$ and $\nabla^2 F(x^*)$ positive semidefinite.

Problem (1) has been extensively studied and various methods for its solution have been proposed, methods that use first and second partial derivatives of function F. Algorithms for the solution of (1) are usually *descent methods*. A descent method imposes the *descent condition* that $F(x_{k+1}) < F(x_k)$ for all $k \in \{0, 1, \ldots\}$. Therefore, descent methods decrease the value of the objective function F at every step. There exist nondescent methods that do not insist on a decrease in the value of F at every step, but even these methods require the value of F to be decreased after some prescribed number of iterations.

There are three fundamental strategies for moving from the current point x_k to a new iterate x_{k+1}: the *linesearch method*, the *trust-region method* and the *method of gradients*. Both the linesearch method and the trust-region method rely on the quadratic approximation

$$m_k(x_k + p) = F(x_k) + \nabla F(x_k)^T p + \frac{1}{2} p^T B_k p$$

of F at the current point x_k, where $(B_k)_{k=0}^{+\infty}$ is a sequence of appropriate symmetric matrices. Algorithms based on setting $B_k = \nabla^2 F(x_k)$ for all $k \in \{0, 1, \ldots\}$ are called *Newton methods*.

Although both the linesearch method and the trust-region method generate steps with the help of a quadratic model of the objective function, they use this model in different ways. Linesearch methods use it to generate, at each iterate x_k, a search direction p_k. Assuming that B_k is positive definite, we obtain the search direction p_k as the unique minimizer of the quadratic model function $m_k(x_k + p)$. The distance α_k to move along p_k can be found by approximately solving the following one-dimensional minimization problem:

$$\underset{\alpha \in \,]0, +\infty[}{\text{minimize}} \quad F(x_k + \alpha\, p_k). \tag{2}$$

By solving (2) exactly, we would derive the maximum benefit from the direction p_k, but an exact minimization is computationally expensive and unnecessary. Instead, a linesearch algorithm generates a limited number of trial step-lengths until it finds one that loosely approximates the solution of (2). Thus we set $x_{k+1} = x_k + \alpha_k p_k$ and the process is repeated.

Instead, trust-region methods start by fixing at each iterate x_k a maximum distance, the trust-region radius Δ_k, and then seek a direction p_k that is an approximate minimizer of the quadratic model function $m_k(x_k + p)$, subject to the distance constraint $\|p\|_2 \leq \Delta_k$. If this step proves not to improve satisfactorily the value of the objective function, we reduce the distance measure Δ_k and try again. When a suitable p_k is found, we set $x_{k+1} = x_k + p_k$.

In a sense, the linesearch and the trust-region approaches differ in the order in which they choose the *direction* and the *length* of the move to the next iterate, as it can be noted from Figure 1.1 on the following page. In a linesearch method the search direction is chosen but then left fixed while the step-length is computed. In a trust-region method the choice of the trust-region radius affects both the length and the direction of the step.

Newton methods are the basis of current unconstrained optimization algorithms. In a linesearch implementation, at each iterate x_k, Newton's method search direction is given by $p_k = -\nabla^2 F(x_k)^{-1} \nabla F(x_k)$. Close to a minimizer of F with a positive definite Hessian matrix, there is a neighborhood where Newton's method works very well, converging quadratically to a solution of (1). However, outside this neighborhood several important issues arise. Indeed, the Hessian matrix $\nabla^2 F(x_k)$ *need not be positive definite*. First, if $\nabla^2 F(x_k)$ is singular, the Newton step is not even defined. Second, even if $\nabla^2 F(x_k)$ is not singular but has a negative eigenvalue, the Newton step may not decrease the value of the objective function F, the quadratic model function is unbounded below and the choice of the step-length varies. In particular, when $\nabla^2 F(x_k)$ is not positive semidefinite but F is bounded below, a conflict exists between the quadratic model function $F(x_k) + \nabla F(x_k)^T p + \frac{1}{2} p^T \nabla^2 F(x_k) p$ and the original nonlinear function F, since the model function indicates that an infinite step should be taken from x_k. Thus, in contrast to the positive-definite case, there is no natural choice of the step-length in the nonpositive-definite case. Consequently, if it is required that we generate a sequence of points whose corresponding values of the objective function are strictly decreasing, some modifications of the algorithm are necessary.

Most methods that use second partial derivatives can be viewed as extensions of

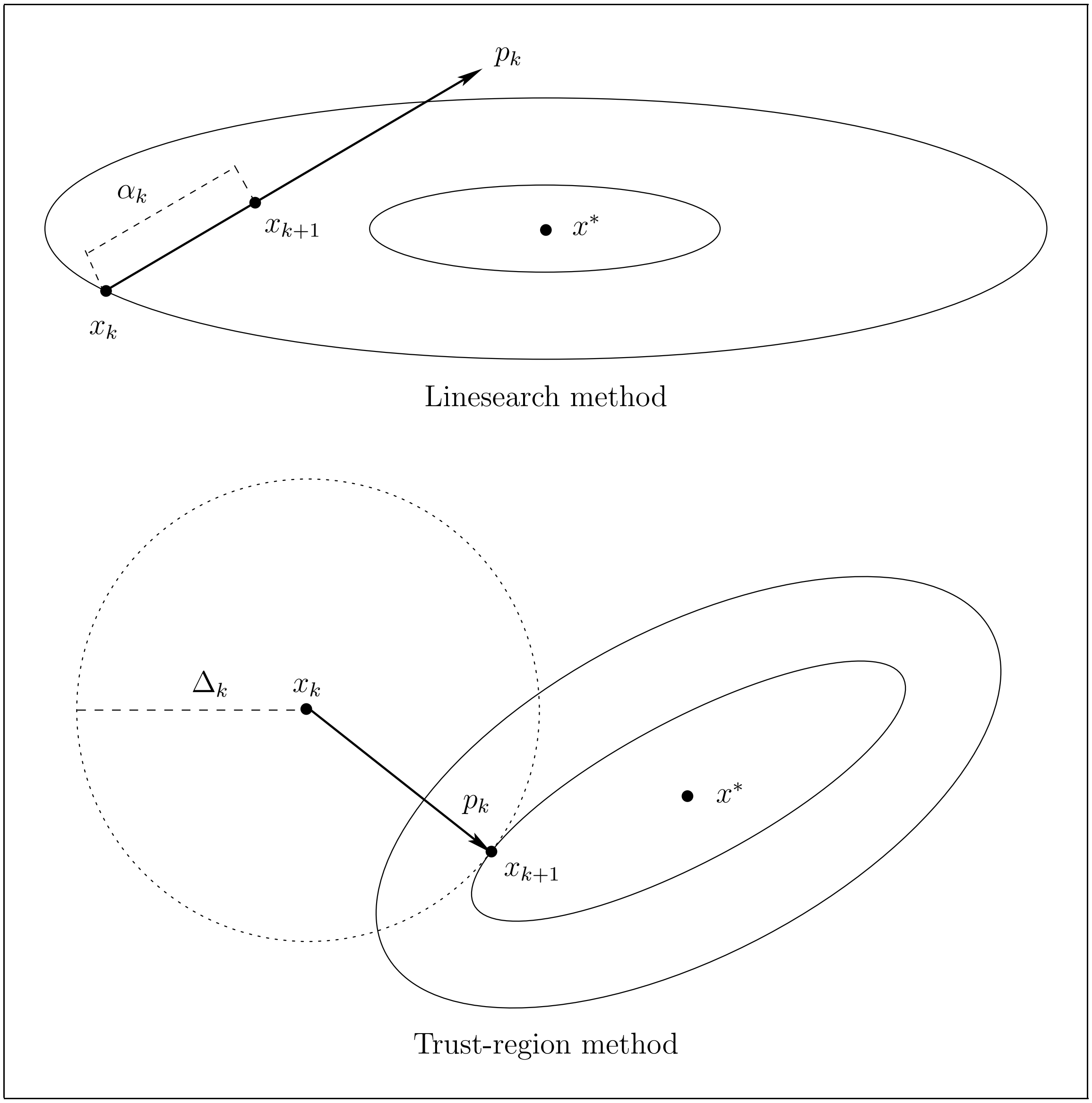

Figure 1.1: Direction and length in linesearch and trust-region methods.

Newton's method. One approach consists in modifying the Hessian matrix $\nabla^2 F(x_k)$ so that it becomes sufficiently positive definite. This is called the *modified Newton method*. The choice of Hessian modification is crucial to the effectiveness of the method. Different implementations have been proposed most notably by Fiacco and McCormick [23], Fletcher and Freeman [24], Forsgren, Gill and Murray [25], Gill and Murray [27], Kaniel and Dax [34] and by Mukai and Polak [43].

Unlike linesearch implementations of Newton's method, trust-region methods do not require the Hessian matrix to be positive definite. Because of the trust-region restriction $\|p\|_2 \leq \Delta_k$, there is no need to do anything special when $\nabla^2 F(x_k)$ is not positive definite, since the constrained minimization of the quadratic model function $m_k(x_k + p)$ within a sphere centered at the current point x_k is guaranteed to have a solution p_k, even when the quadratic model function is unbounded below. However, trust-region methods are highly sensitive to poor scaling, while linesearch implementations of Newton's method are unaffected by it.

The unconstrained optimization problem (1) is said to be *poorly scaled* if changes to the variable x in a certain direction produce much larger variations in the value of the objective function F than do changes to x in another direction. Topologically, a symptom of poor scaling is that the minimizer lies in a narrow valley, so that the level curves of the objective function F near the minimizer tend towards highly eccentric ellipsoids.

It is easy to see that a spherical trust region is not appropriate to the case of poorly scaled problems. In fact, we can trust our model m_k to be reasonably accurate only for short distances along the highly sensitive directions, while it is reliable for longer distances along the less sensitive directions. Since the shape of our trust region should be such that our confidence in the model is more or less the same at all points on the boundary of the region, we are led naturally to consider *ellipsoidal* trust regions in which the axes are short in the sensitive directions and longer in the less sensitive directions.

Figure 1.2 on the next page shows the good performance of a standard trust-region

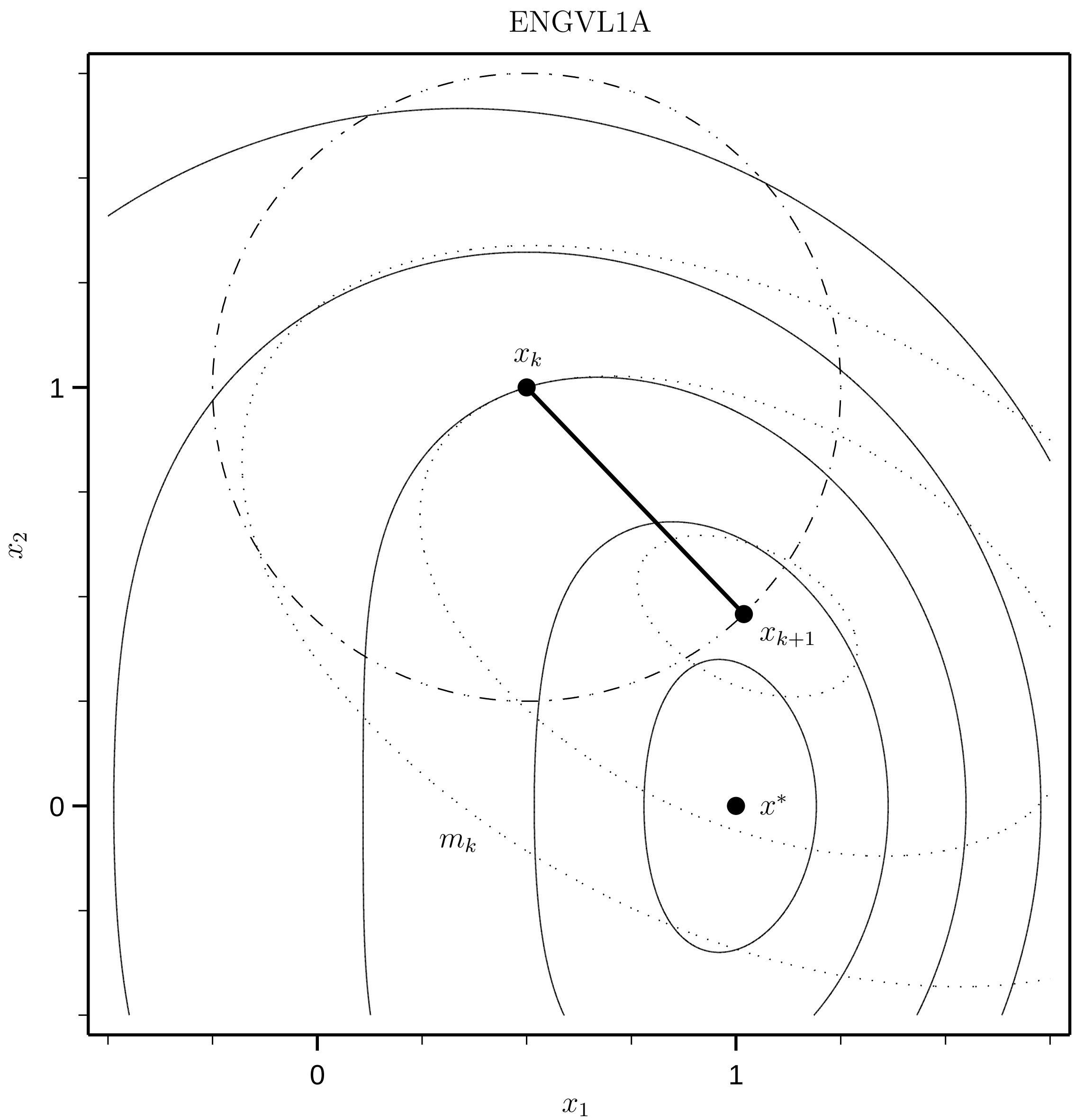

Figure 1.2: Performance of trust-region methods on a well-scaled problem.

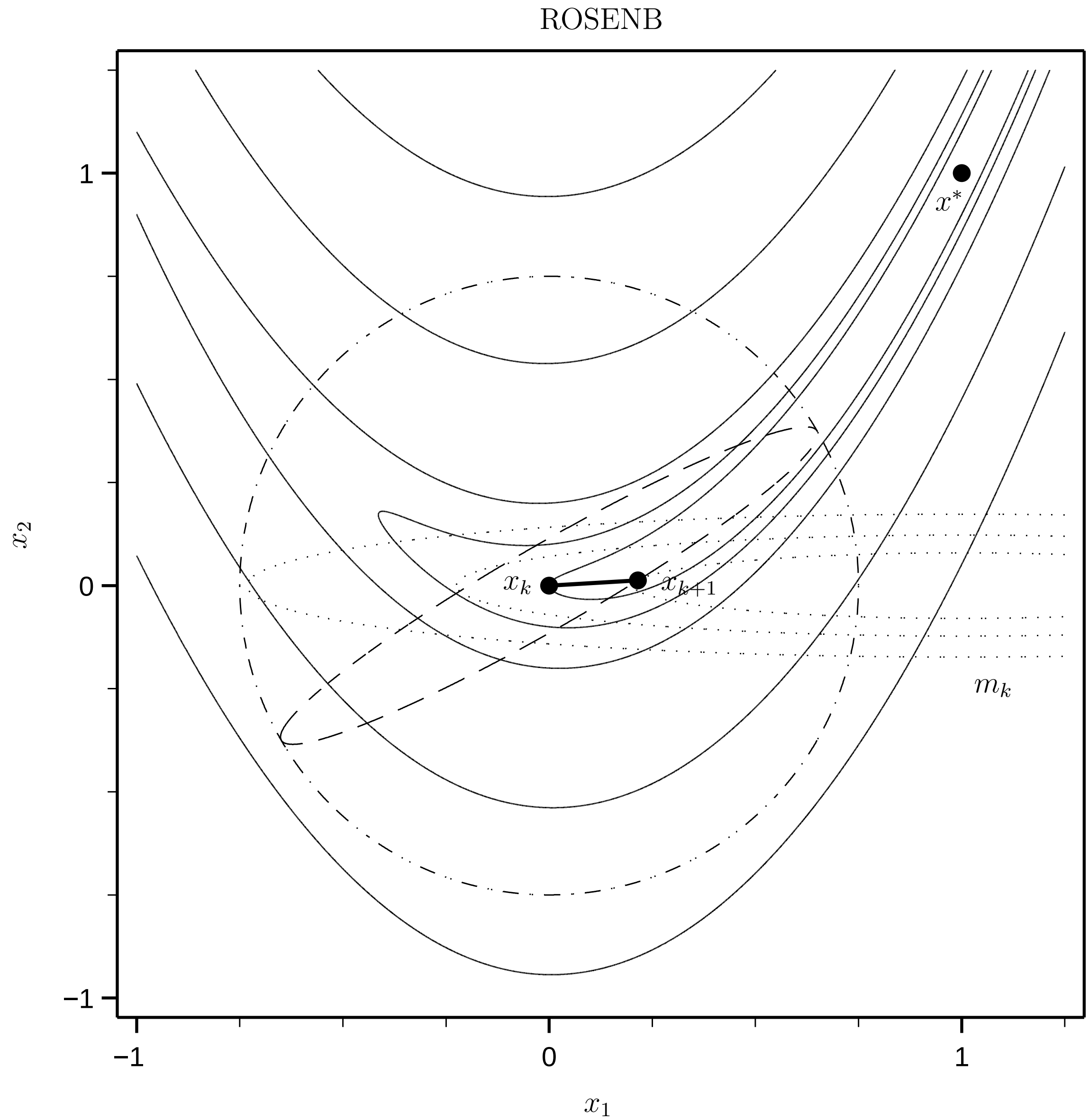

Figure 1.3: Performance of ellipsoidal trust-region methods on the Rosenbrock function.

method on the well scaled problem

$$\underset{x \in \mathbf{R}^2}{\text{minimize}} \quad \left(x_1^2 + x_2^2\right)^2 - 4x_1 + 3\,.$$

On the other hand, Figure 1.3 on the preceding page shows the poor performance of a spherical trust-region method on the two-dimensional Rosenbrock problem

$$\underset{x \in \mathbf{R}^2}{\text{minimize}} \quad 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2\,. \tag{3}$$

The objective function of (3) has a unique minimum at the point $(1, 1)$ which lies at the base of a banana-shaped valley where the level curves of F are highly eccentric quasi-ellipses. If at $x_k = (0, 0)$ we had used the spherical trust region that is depicted in the figure, then we would have had $x_k + p_k$ parallel to the x_1-axis and pointing towards the minimizer of the model function m_k. However, such a step would *not have decreased* the value of the objective function. Indeed, in order to get a decreasing step we should have halved the radius of the trust region twice and, consequently, solved two more trust-region subproblems. Instead, it is clear that it is more appropriate to use an elliptical trust region with the semimajor axis along the elongated curves of the banana-shaped valley.

Another drawback of trust-region methods is that they are not second order stationary point convergent when they are extended to constrained optimization problems, even though they incorporate second order information. However, different implementations of trust-region methods have been proposed over the years. Moré [40] summarizes developments in trust-region algorithms and software before 1982, paying particular attention to the importance of using a scaled trust-region norm. Moré and Sorenson [42] discuss methods that use nearly exact solutions to the constrained subproblem, while Shultz, Schnabel and Byrd [51] provide a general theory for inexact trust-region methods.

1.2 The method of gradients

On May 3rd, 1941 Richard Courant [19] delivered an address before the meeting of the American Mathematical Society in Washington, DC, by invitation of the Program Committee. He proposed three variational methods for numerical calculation of the solution of partial differential equations arising in problems of equilibrium and vibrations: the Rayleigh-Ritz method, the method of finite differences and the method of gradients. The first two methods went into widespread use; in particular, the Rayleigh-Ritz method has evolved into what today is known as the method of finite elements. On the contrary, the method of gradients was not pursued as much because difficulties in convergence theory were encountered and because the size of computer memory placed severe constraints on the size of programs for numerical algorithms.

The idea behind the method of gradients is very old. Courant himself states that it goes back to a paper published by Jacques Hadamard in 1908. Problems of equilibrium and vibrations lead to linear partial differential equations for an unknown function $u(x_1, x_2)$ in a bounded subset B of $\mathbf{R}^2$, or rather to equivalent variational problems for the kinetic and potential energies of the system. In order to proceed in our discussion, we now need to recall a few definitions and theorems of functional analysis. For an extensive presentation of this material we refer to Zeidler [58], [59], [60], [61] and [62].

Definition 1.1 Let $T : U \subseteq V \to Y$ be a transformation from a subset U of a linear vector space V to a normed linear vector space Y. If for fixed $u \in U$ such that $u + tv \in U$ for all $v \in V$ and all real numbers t in some neighborhood of zero, the limit

$$\lim_{t\to 0} \frac{T(u+tv) - T(u)}{t}$$

exists for all $v \in V$, then the transformation T is said to be *Gâteaux differentiable at* u. Such a limit is called the *Gâteaux differential of* T *at* u *with increment* v and is denoted by $\delta T(u; v)$. If T is Gâteaux differentiable at every point u in a subset X of U, then the transformation T is said to be *Gâteaux differentiable on* X.

The Gâteaux differential generalizes the concept of directional derivative that is familiar in finite-dimension analysis. However, the existence of the Gâteaux differential is a rather weak requirement, since its definition needs no norm on V. Therefore, properties of the Gâteaux differential are not easily related to continuity. When V is normed, a more satisfactory definition is given by the Fréchet differential.

Definition 1.2 Let $T : U \subseteq V \to Y$ be a transformation from an open subset U of a normed linear vector space V to a normed linear vector space Y. If for fixed $u \in U$ and for all $v \in V$ there exists a $dT(u; v) \in Y$, which is linear and continuous with respect to v, such that

$$\lim_{\|v\|_V \to 0} \frac{\|T(u + v) - T(u) - dT(u; v)\|_V}{\|v\|_V} = 0,$$

then the transformation T is said to be *Fréchet differentiable at* u, and $dT(u; v)$ is said to be the *Fréchet differential of* T *at* u *with increment* v. If T is Fréchet differentiable at u, the *Fréchet derivative of* T *at* u, which we denote either by $T'(u)$ or by $DT(u)$, is defined by $dT(u; v) = T'(u)v$. If T is Fréchet differentiable at every point u in a subset X of U, then the transformation T is said to be *Fréchet differentiable on* X. The transformation $DT : u \in X \subseteq V \to DT(u) \in B(X, Y)$, defined as u varies over X, where $B(X, Y)$ is the normed vector space of all bounded linear transformations from X into Y, is called the *Fréchet derivative of* T *on* X.

We now state a few basic and useful properties of a transformation that is Fréchet differentiable.

Proposition 1.3 *Let* $T : U \subseteq V \to Y$ *be a transformation from an open subset* U *of a normed linear vector space* V *to a normed linear vector space* Y*.*

i) *If the transformation* T *has a Fréchet derivative on a subset* X *of* U*, it is unique.*

ii) *If the transformation* T *is Fréchet differentiable at* $u \in U$*, then* T *is continuous at* u*.*

iii) *If the transformation T is Fréchet differentiable at $u \in U$, then the T is Gâteaux differentiable at u and, for all $v \in V$, the Gâteaux differential of T at u with increment v is equal to the Fréchet differential of T at u with increment v.*

By far the most frequent application of Definitions 1.1 and 1.2 is in the case where the normed linear vector space Y is the real field $\mathbf{R}$ and hence the transformation T reduces to a real-valued functional $E : U \subseteq V \to \mathbf{R}$. Moreover, if the linear vector space V is a normed space, then we can define the normed dual V^* of V and the duality pairing $\langle \cdot, \cdot \rangle : V \times V^* \to \mathbf{R}$. In these conditions, for each fixed $u \in U$ the Fréchet derivative of E at u is an element of V^* and we can write $dE(u; v) = \langle v, E'(u) \rangle$. Finally, assume the real-valued functional E is defined on an open subset U of an Hilbert space H with scalar product $(\cdot, \cdot) : U \times U \subseteq H \times H \to \mathbf{R}$ and induced norm $\|\cdot\|_H : U \subseteq H \to \mathbf{R}$. Then we can define a gradient vector field $\nabla E : U \subseteq H \to H$, as in the finite-dimensional case, by letting $\nabla E(u)$ at any $u \in U$ be the unique vector in H such that $(\nabla E(u), v) = \langle v, E'(u) \rangle$ for all $v \in H$, or equivalently characterized by $\|\nabla E(u)\|_H = \|DE(u)\|_H$. The vector $\nabla E(u)$ is naturally called the *gradient of E at u*.

A particular class of functional equations $F(u) = 0$, for u belonging to some Banach space V, is the class of Euler-Lagrange equations

$$DE(u) = 0$$

for a functional E on V which is Fréchet differentiable with derivative DE. Such equations are called of *variational form*. In particular, for each partial differential equation

$$\frac{\partial}{\partial x_1} L_{u_{x_1}}(P) + \frac{\partial}{\partial x_2} L_{u_{x_2}} - L_u(P) = 0, \qquad u = g \quad \text{on} \quad \partial B, \tag{4}$$

on a bounded subset B of $\mathbf{R}^2$, where $P = (x_1, x_2, u(x_1, x_2), u_{x_1}(x_1, x_2), u_{x_2}(x_1, x_2))$, $u \in C^2(\bar{B})$ and g is given, there exists a functional

$$E(u) = \int_B L(x_1, x_2, u, u_{x_1}, u_{x_2})\, dB \tag{5}$$

such that a sufficiently smooth minimizer u^* of (5), with $u^* = g$ on ∂B, satisfies partial differential equation (4). For example, the problem of determining the *equilibrium of a membrane with given boundary deflections* $g(s)$ can be formulated as either the partial differential equation

$$\Delta_2 u = 0, \qquad u = g \quad \text{on} \quad \partial B,$$

where Δ_2 denotes the Laplace operator $\sum_{i=1}^{2} \frac{\partial}{\partial x_i}$, or as the variational problem

$$\underset{u}{\text{minimize}} \quad E(u) = \int_B \left(u_{x_1}^2 + u_{x_2}^2\right) \, dB, \qquad u = g \quad \text{on} \quad \partial B.$$

The principle of the method of gradients may be understood from the elementary geometric concept of a gradient vector. Let $u = f(x_1, \ldots, x_n)$ be a nonnegative function of the n variables $x_1, \ldots, x_n$ or, as we might say, of the position vector $X = (x_1, \ldots, x_n)$, and let us seek to determine a vector X^* for which u is at lest stationary. We then proceed as follows: on the surface $u = f(x)$ we move a point $(x_1, \ldots, x_n, u)$ so that $x_1(t), ldots, x_n(t)$ and $u(t)$ become functions of a time-parameter t. Then the velocity of ascent or descent along the line $X = X(t), u = u(t)$ on the surface is

$$\begin{aligned} \frac{du}{dt} &= \dot{u} \sum_{i=1}^{n} \dot{x}_i f_{x_i} \\ &= \dot{X} \nabla f. \end{aligned}$$

We now choose the line along which the motion proceeds so that the descent is as steep as possible. This means to make $\dot{u}$ negative and as large as possible in absolute value, for example, by choosing

$$\dot{X} = -\nabla f \tag{6}$$

so that

$$\dot{u} = -\|\nabla f\|_2^2.$$

Hence the position vector X moves according to the system of ordinary differential equations (6) along the lines of steepest descent with respect to the function f. Under very general assumptions, it is clear that X, starting from an *arbitrary* initial point, will, as t goes to $+\infty$, approach a position for which $\nabla f = 0$, and therefore for which f is stationary and possibly a minimum.

This elementary idea can be generalized to variational problems. If we wish to determine a function u^* defined in $B \subseteq \mathbf{R}^2$ and having *prescribed boundary values* such that u^* solves the variational problem

$$\underset{u}{\text{minimize}} \quad E(u) = \int_B L(x_1, x_2, u, u_{x_1}, u_{x_2})\, dB,$$

then we interpret the desired function u^* as

$$\lim_{t \to +\infty} u(x_1, x_2, t)$$

where the values of $u(x_1, x_2, t)$ are chosen as follows:

1. for $t = 0$, the value of u is chosen arbitrarily;

2. for $t > 0$, the values of u are chosen in such a way that the expression $E(u)$, considered as a function $E(t)$ of t, *decreases as rapidly as possible toward its minimal value* $E(u^*)$.

Of course the boundary values of $u(x_1, x_2, t)$ are the same as those for $u^*(x_1, x_2)$, so that $\dfrac{\partial u}{\partial t}$ must vanish at the boundary. Moreover, we find

$$\frac{\partial}{\partial t} E(u(x_1, x_2, 0), t) = -\int_B u_t L(u)\, dB. \tag{7}$$

We now assure ourselves of a steady descent or decrease of $E(t)$ by choosing u_t in accordance with the differential equation

$$u_t = kL(u),$$

where k is an arbitrary positive function of x_1 and x_2. Equation (7) then becomes

$$\frac{\partial}{\partial t} E(u(x_1, x_2, 0), t) = -\int_B k \left[L(u)\right]^2\, dB,$$

and again we can infer that, as t goes to $+\infty$, $u(x_1, x_2, t)$ will tend to the solution $u^*(x_1, x_2)$ of the corresponding boundary value problem $L(u) = 0$.

However, even in a finite-dimensional setting with suitable assumptions about the topology of the level sets of E, the convergence of u as t goes to $+\infty$ in general cannot be proved unless a certain compactness property holds. We shall require the so-called *Palais-Smale condition* to be satisfied by E. See [55] for a treatise on this subject.

In the original work of Palais and Smale, see Palais [47], [48], Palais and Smale [49] and Smale [52], the compactness property is stated as follows:

Definition 1.4 Let $E : V \to \mathbf{R}$ be a real-valued functional defined and Gâteaux differentiable on a Banach space V. The functional E is said to satisfy *condition (C)* if given any subset S of V on which $|E|$ is bounded but on which $\|DE\|$ is not bounded away from zero there is a critical point of E in the closure of S.

We replace condition (C) by a slightly stronger condition which is easier to work with.

Definition 1.5 (Palais-Smale Condition) Let $E : V \to \mathbf{R}$ be a real-valued functional defined and Gâteaux differentiable on a Banach space V. A sequence $(u_n)_{n\in\mathbf{N}}$ in V is called a *Palais-Smale sequence for* E if $(E(u_n))_{n\in\mathbf{N}}$ is uniformly bounded, while $\lim\limits_{n\to+\infty} \|DE(u_n)\|_V = 0$. The functional E is said to satisfy the *Palais-Smale condition* if any Palais-Smale sequence for E has a (strongly) convergent subsequence.

Condition (C) implies the Palais-Smale condition. However, the converse is not true. In fact, the functional $E \equiv 0$ satisfies (C) but in general will not satisfy the Palais-Smale condition.

We can now state the main theorem in this section. Towards this end, for $\beta \in \mathbf{R}$, let

$$E_\beta = \{u \in V : \ E(u) < \beta\} ,$$
$$K_\beta = \{u \in V : \ E(u) = \beta, \ DE(u) = 0\} .$$

Theorem 1.6 (Deformation Lemma) *Suppose $E : V \to \mathbf{R}$ is a real-valued functional defined and continuously Fréchet differentiable on a Banach space V and satisfying the Palais-Smale condition. Let $\beta \in \mathbf{R}$, $\bar{\varepsilon} > 0$ be given and let N be any neighborhood of K_β. Then there exist a number $\varepsilon \in \,]0, \bar{\varepsilon}[$ and a continuous 1-parameter family of homeomorphisms $\Phi(\cdot, t)$ of V, $t \in [0, +\infty[$, with properties*

1. *$\Phi(u, t) = u$, if $t = 0$, or if $DE(u) = 0$, or if $|E(u) - \beta| \geq \bar{\varepsilon}$;*
2. *$E(\Phi(u, t))$ is nonincreasing in t for all $u \in V$;*
3. *$\Phi(E_{\beta+\varepsilon} - N, 1) \subseteq E_{\beta-\varepsilon}$ and $\Phi(E_{\beta+\varepsilon}, 1) \subseteq E_{\beta-\varepsilon} \cup N$.*

Moreover, $\Phi : V \times [0, +\infty[\to V$ has the semi-group property; that is, $\Phi(\cdot, t) \circ \Phi(\cdot, s) = \Phi(\cdot, s + t)$ for all s, $t \geq 0$.

Consider the special case when $E : H \to \mathbf{R}$ is a real-valued functional defined and *twice-continuously* Fréchet differentiable on a *real Hilbert space* H with scalar product $(\cdot, \cdot) : H \times H \to \mathbf{R}$ and induced norm $\|\cdot\|_H : H \to \mathbf{R}$. In this case, we have already seen that we can define a *gradient vector field* $\nabla E : H \to H$. Moreover, since

$$\|\nabla E(u) - \nabla E(v)\|_H = \|DE(u) - DE(v)\|_H,$$

if $E \in C^2$, then ∇E is of class C^1 and defines a *local gradient flow* Φ by letting

$$\begin{cases} \dfrac{\partial}{\partial t}\Phi(u, t) & = -\nabla E(\Phi(u, t)) \\ \Phi(u, 0) & = u\,. \end{cases} \tag{8}$$

If we identify E with its graph $\mathcal{G}(E) = \{(u, E(u)) \in H \times \mathbf{R}\}$, then the *flow-lines* of Φ become *paths of steepest descent*, and the *rest points* of Φ are precisely the *critical points* of E where $\mathcal{G}(E)$ has a horizontal tangent plane. If $K_\beta = \varnothing$ in the Deformation Lemma, we may choose $N = \varnothing$ and hence, from item *3*, obtain a uniform reduction of E near β.

A solution Φ to equation (8) is called an *integral curve of* $-\nabla F$ and is simply the curve that at each instant proceeds in the direction of the steepest descent of F. The integral curve of $-\nabla F$ from $x_0 = (0, 0)$ for the two-dimensional test problem

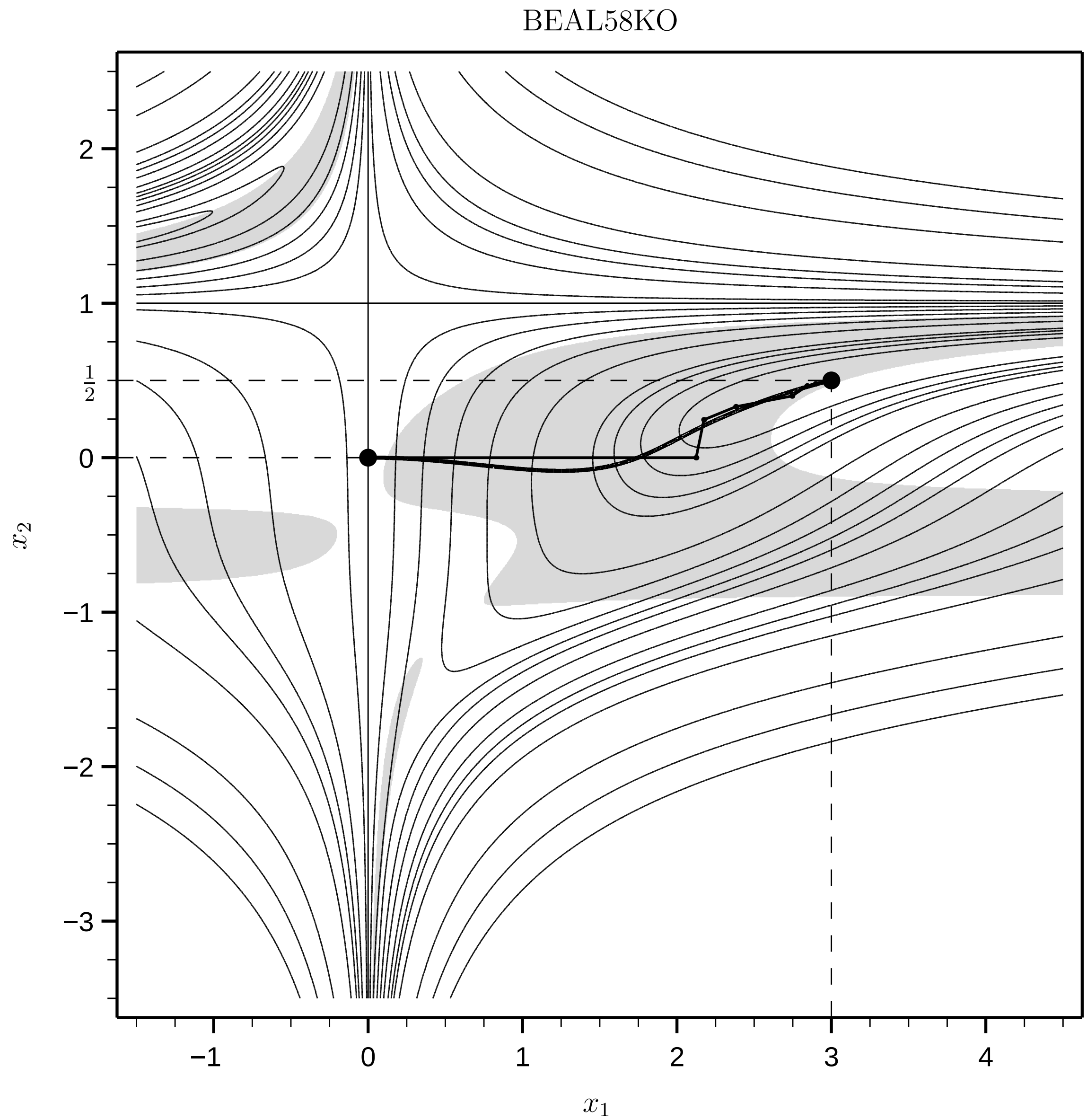

Figure 1.4: Level curves of $F(x) = \left[\frac{3}{2} - x_1(1-x_2)\right]^2 + \left[\frac{9}{4} - x_1(1-x_2^2)\right]^2 + \left[\frac{21}{8} - x_1(1-x_2^3)\right]^2$ and integral curve of $-\nabla F(x)$ from $x_0 = (0,0)$; the shaded zones are the set of points where the Hessian matrix is positive definite.

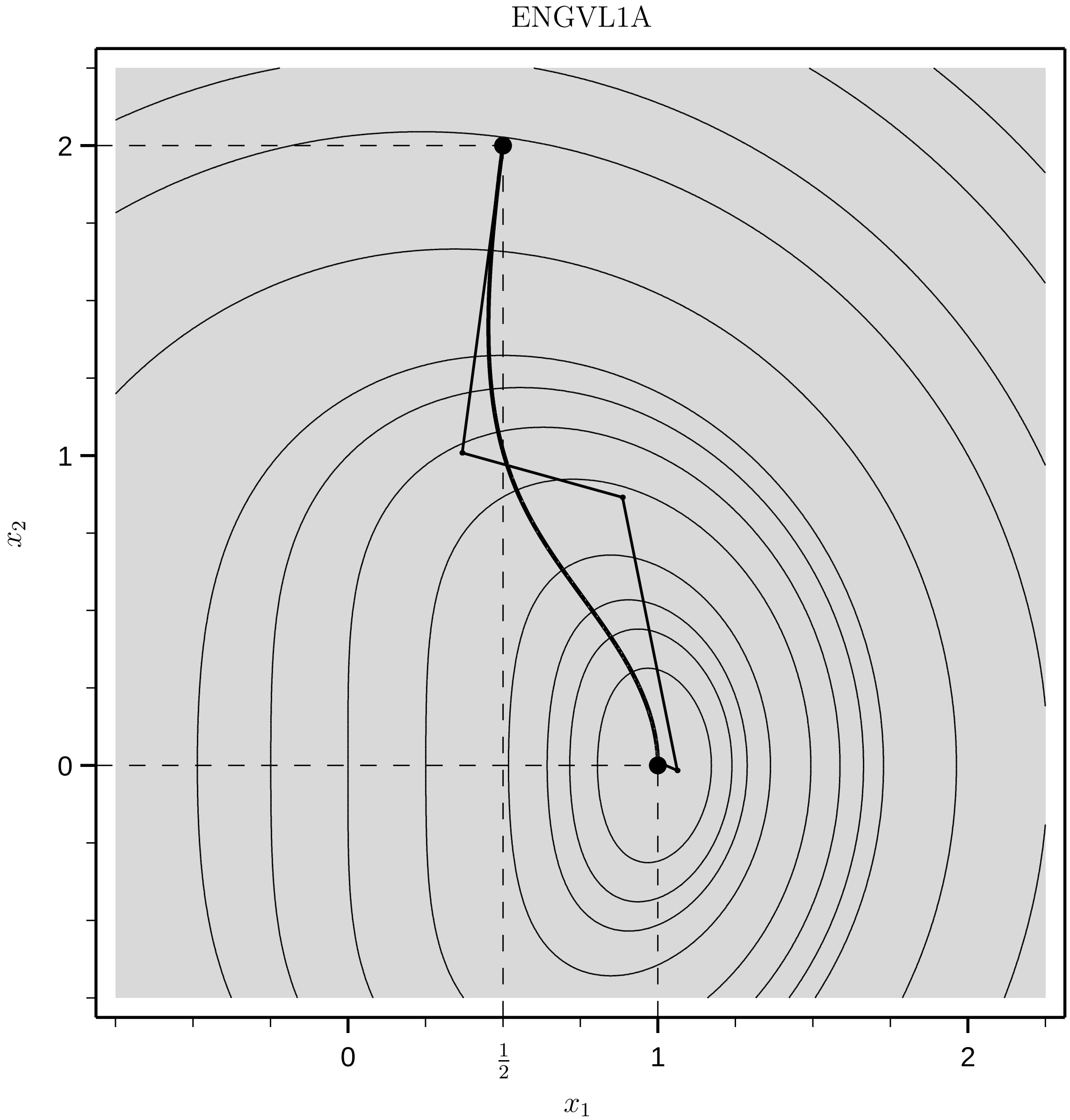

Figure 1.5: Level curves of $F(x) = (x_1^2 + x_2^2)^2 - 4x_1 + 3$ and integral curve of $-\nabla F(x)$ from $x_0 = \left(\frac{1}{2}, 2\right)$;the shaded zone is the set of points where the Hessian matrix is positive definite.

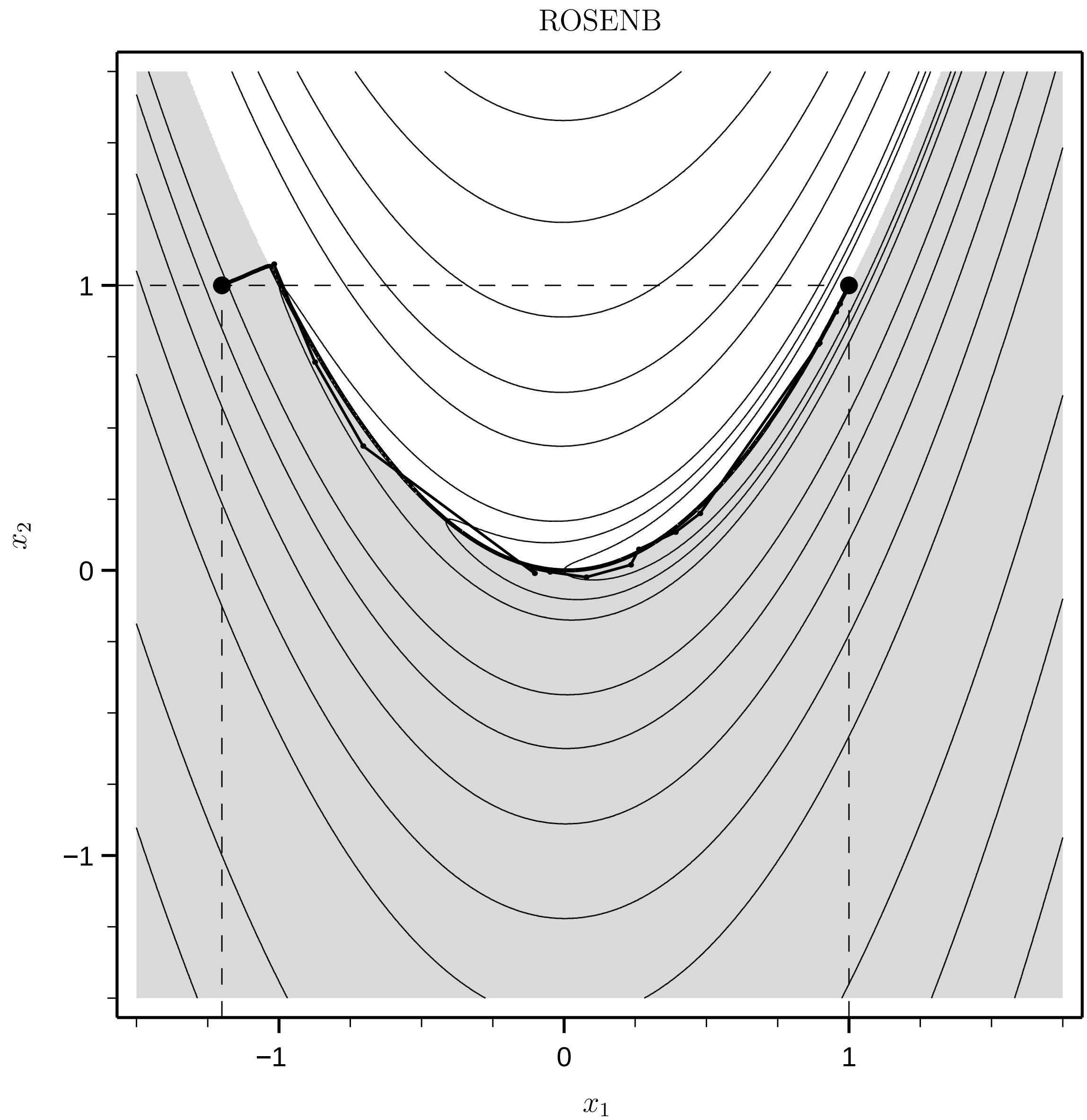

Figure 1.6: Level curves of $F(x) = 100\,(x_2 - x_1^2)^2 + (1 - x_1)^2$ and integral curve of $-\nabla F(x)$ from $x_0 = (-1.2, 1)$; the shaded zone is the set of points where the Hessian matrix is positive definite.

BEAL58KO is shown in Figure 1.4 on page 16. Analogously, Figure 1.5 on page 17 shows the integral curve of $-\nabla F$ from $x_0 = \left(\frac{1}{2}, 2\right)$ for the test problem ENGVL1A, and Figure 1.6 on the preceding page shows the integral curve of $-\nabla F$ from $x_0 = (-1.2, 1)$ for the test problem ROSENB. In these figures, the integral curves connect the initial point x_0 with a minimizer and are normal to all level curves. The shaded zones are the set of points where the Hessian matrix is positive definite. Moreover, the figures also illustrate the solution paths of a quasi-Newton algorithm that was implemented with the BFGS update to generate the sequence of matrices $(B_k)_{k=0}^{+\infty}$ in the definition of the quadratic model functions $m_k(x_k + p)$.

Since the early days of the method of gradients a number of significant changes have occurred. As we have seen, in the years between 1963 and 1966, the theory of nonlinear functional analysis was expanded and included a convergence theory for the method of gradients in the case of exact integral curves. Moreover, the size of computer memory has greatly increased and no longer places severe constraints on the size of programs for numerical algorithms.

While historically much of the motivation for the method of gradients and the mathematical analysis behind it came from the desire to solve partial differential equations, the function it seeks to minimize need not come from a partial differential equation. Indeed, the method of gradients is a general-purpose method for unconstrained optimization. However, its convergence theory requires knowledge of the objective function at a continuum of points, which is not practical for computer implementation.

The method of gradients has not found general acceptance because of the difficulties inherent in solving a system of *nonlinear* ordinary differential equations. Recently, William Behrman [6] proposed a gradient flow method based upon the solution of a system of n *linear* ordinary differential equations, where n denotes the number of independent variables of the objective function. He also reviewed older methods based on the idea of using ordinary differential equations for unconstrained optimization, namely, Aluffi-Pentini, Parisi and Zirilli [1], [2], Boggs [8], Botsaris [10], [11], [12], Botsaris and Jacobson [13], Brown and Bartholomew-Biggs [15], Vial and Zang [56] and Zang [57].

The method presented in this dissertation is based upon the solution of a system of *only two* ordinary differential equations. These equations are defined and solvable regardless of the nature of the Hessian matrix. Their solutions are curves that are unaffected by poor scaling and are guaranteed to have a point at which the objective function has a lower value than at the current point.

Chapter 2

THE SUBSPACE METHOD

This chapter presents the motivation and the mathematical analysis for a subspace method based on a differential equation approach to solve the following nonlinear unconstrained optimization problem:

$$\underset{x \in D}{\text{minimize}} \quad F(x), \tag{1}$$

where $F(x)$ is a twice-continuously differentiable real-valued function defined on an open set $D \subseteq \mathbf{R}^n$. In the following, assume x_0 is an element of the set D.

2.1 Motivation

In Chapter 1 we have seen that, starting at x_0, we can determine a stationary point x^* of F by following the integral curve of $-\nabla F$ from x_0. Such a curve is the unique solution of the following system of n nonlinear ordinary differential equations:

$$\begin{cases} \Phi'(x_0, t) & = -\nabla F(\Phi(x_0, t)) \\ \Phi(x_0, 0) & = x_0. \end{cases} \tag{2}$$

The first step towards practical implementation of a minimization algorithm is to *discretize* problem (2). Let $(x_k)_{k=0}^{+\infty}$ be a sequence of iterates in D such that

$$\lim_{k \to +\infty} x_k = x^*.$$

At each iterate x_k, we may consider the integral curve Φ_k of $-\nabla F$ from x_k, that is, the solution of

$$\begin{cases} \Phi_k'(x_k, t) & = -\nabla F(\Phi_k(x_k, t)) \\ \Phi_k(x_k, 0) & = x_k. \end{cases}$$

We then approximate Φ_k with a continuous curve $x_k + q_k(t)$ by approximating the gradient vector field $-\nabla F$ in a neighborhood of x_k with a vector field g_k and letting $q_k(t)$ be the solution of

$$\begin{cases} q_k'(t) & = -g_k(x_k + q_k(t)) \\ q_k(0) & = 0. \end{cases} \tag{3}$$

The second step is to *linearize* system (3). In particular, since $F \in C^2(D)$, we have $\nabla F \in C^1(D)$ and the linear vector field

$$g_k(x_k + q_k(t)) = \nabla F(x_k) + \nabla^2 F(x_k) q_k(t)$$

is the *Taylor polynomial of order one of function* ∇F *at* x_k. Hence we consider the curve $q_k(t)$, $t \geq 0$, which is the solution of the following system of n *linear* ordinary differential equations:

$$\begin{cases} q_k'(t) & = -\nabla F(x_k) - \nabla^2 F(x_k) q_k(t) \\ q_k(0) & = 0. \end{cases} \tag{4}$$

While linesearch and trust-region methods rely at each iterate x_k on a *quadratic* approximation of the *objective function* F in a neighborhood of x_k, the method of gradients that results from the solution of (4) relies on a *linear* approximation, in a neighborhood of x_k, of the vector field ∇F, which is the *gradient* of the objective function.

Figure 2.1 on the following page illustrates the spirit of the method proposed by Behrman [6] and based, at each iteration, upon the solution of problem (4). Starting at at point $x_0 \in D$, the algorithm computes $q_0(t)$ and finds a point x_1 along the curve $x_0 + q_0(t)$. The search is constrained from going too far into a region in which the approximation used to define the search curve is no longer valid. The search is also constrained from being too close to the initial point, in order to avoid taking

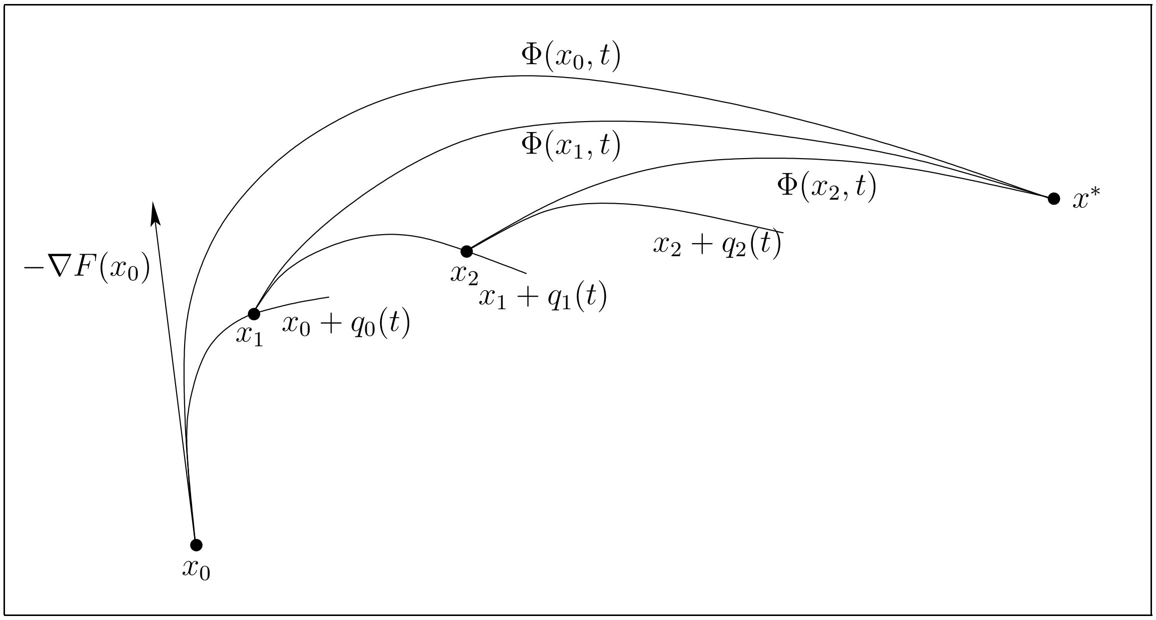

Figure 2.1: Behrman's method.

too small steps, which would prevent the iterates from converging to a critical point. Since the integral curve $\Phi(x_0, t)$ of the negative gradient field from x_0 is not available to the algorithm, it measures the validity of the search curve at a point by comparing the tangent to the search curve and the negative gradient field at that point. At x_1 the algorithms computes $q_1(t)$ and then finds a point x_2 along the curve $x_1 + q_1(t)$ by comparing the tangent to it with the tangent to the integral curve $\Phi(x_1, t)$ of $-\nabla F$ from x_1, and so on.

We now analyze the properties of the curve $x_k + q_k(t)$ that results from the solution of problem (4). Since $\nabla^2 F(x_k)$ is symmetric, all its *eigenvalues are real* and it possesses a *complete set of real orthonormal eigenvectors*: see Theorem 8.1.1 in Golub and Van Loan [30], or Strang [54]. Let $\lambda_1(x_k) \geq \cdots \geq \lambda_n(x_k)$ denote the eigenvalues of $\nabla^2 F(x_k)$, and let $v_1(x_k), \ldots, v_n(x_k)$ denote a set of orthonormal eigenvectors of $\nabla^2 F(x_k)$ corresponding to $\lambda_1(x_k), \ldots, \lambda_n(x_k)$, respectively. The solution of (4) is

$$q_k(t) = \sum_{i=1}^{n} \left(\frac{e^{-\lambda_i(x_k)t} - 1}{\lambda_i(x_k)} \, I_{\{\lambda_i(x_k) \neq 0\}} - t \, I_{\{\lambda_i(x_k) = 0\}} \right) v_i(x_k) v_i(x_k)^T \nabla F(x_k), \qquad (5)$$

$t \in [0, +\infty[$, where $I_{\{\lambda_i(x_k)\neq 0\}}$ and $I_{\{\lambda_i(x_k)=0\}}$ are the following indicator functions:

$$I_{\{\lambda_i(x_k)\neq 0\}} = \begin{cases} 1 & \text{if } \lambda_i(x_k) \neq 0 \\ 0 & \text{if } \lambda_i(x_k) = 0 \end{cases}, \qquad I_{\{\lambda_i(x_k)=0\}} = \begin{cases} 1 & \text{if } \lambda_i(x_k) = 0 \\ 0 & \text{if } \lambda_i(x_k) \neq 0 \end{cases},$$

with $i \in \{1, \ldots, n\}$ and $k \in \{0, 1, \ldots\}$.

Clearly, $q_k(0) = 0$, which means that x_k is the point on the search curve corresponding to $t = 0$. Furthermore,

$$q_k'(t) = -\sum_{i=1}^{n} e^{-\lambda_i(x_k)t} v_i(x_k) v_i(x_k)^T \nabla F(x_k),$$

and

$$\begin{aligned} q_k'(0) &= -\sum_{i=1}^{n} v_i(x_k) v_i(x_k)^T \nabla F(x_k) \\ &= -\nabla F(x_k). \end{aligned}$$

Therefore, at each iterate x_k the *tangent to the curve* $x_k + q_k(t)$ is equal to $-\nabla F(x_k)$, i.e., it is the *steepest descent direction* at x_k.

Consider now the limit values of $x_k + q_k(t)$, namely, the following limit:

$$\lim_{t\to+\infty} q_k(t).$$

Towards this end we now have to make a case distinction. If $\nabla^2 F(x_k)$ is *positive definite*, then $\lambda_i(x_k) > 0 \;\; \forall i \in \{1, \ldots, n\}$ and

$$\begin{aligned} \lim_{t\to+\infty} q_k(t) &= \lim_{t\to+\infty} \sum_{i=1}^{n} \frac{e^{-\lambda_i(x_k)t} - 1}{\lambda_i(x_k)} v_i(x_k) v_i(x_k)^T \nabla F(x_k) \\ &= -\sum_{i=1}^{n} \frac{v_i(x_k) v_i(x_k)^T}{\lambda_i(x_k)} \nabla F(x_k) \\ &= -\nabla^2 F(x_k)^{-1} \nabla F(x_k), \end{aligned}$$

which is the *Newton step* at x_k. Therefore, if $\nabla^2 F(x_k)$ is positive definite, the curve $x_k + q_k(t)$ is *bounded*, as shown in Figure 2.2 on the next page.

If $\nabla^2 F(x_k)$ is *not positive definite*, then we first make the following assumptions in order to simplify the discussion.

Assumption 2.1 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$. If at point $x_k \in D$ the Hessian matrix $\nabla^2 F(x_k)$ is not positive definite, then the gradient $\nabla F(x_k)$ has a component in the eigenspace of nonpositive eigenvalues of $\nabla^2 F(x_k)$, that is,*

$$\exists\, i \in \{1, \dots, n\} : \lambda_i(x_k) \leq 0 \Rightarrow v_i(x_k)^T \nabla F(x_k) \neq 0,$$

where $\lambda_1(x_k) \geq \cdots \geq \lambda_n(x_k)$ and $v_1(x_k), \dots, v_n(x_k)$ denote the eigenvalues of $\nabla^2 F(x_k)$ and a set of corresponding orthonormal eigenvectors, respectively.

Assumption 2.2 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$. If at point $x_k \in D$ the Hessian matrix $\nabla^2 F(x_k)$ is not positive definite, then the smallest eigenvalue of $\nabla^2 F(x_k)$, say $\lambda_{\sigma_k}(x_k)$, corresponding to an eigenvector with a component in the direction of the gradient $\nabla F(x_k)$, has algebraic multiplicity equal to one.*

Under Assumption 2.1 and Assumption 2.2, we know that the minimum of the set

$$\mathcal{E}_k = \left\{ \lambda_i(x_k) : \lambda_i(x_k) \leq 0,\, v_i(x_k)^T \nabla F(x_k) \neq 0 \right\} \tag{6}$$

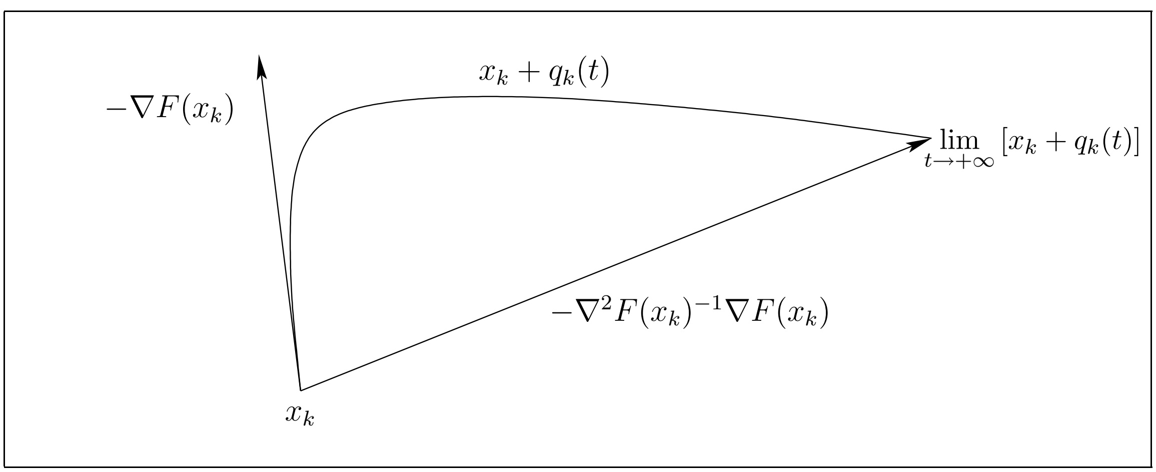

Figure 2.2: Search curve when $\nabla^2 F(x_k)$ is positive definite.

exists and is *unique*. Therefore, we can define the index

$$\sigma_k = \operatorname*{argmin}_{i \in \{1,\ldots,n\}} \left\{ \lambda_i(x_k) : \lambda_i(x_k) \leq 0,\, v_i(x_k)^T \nabla F(x_k) \neq 0 \right\}. \tag{7}$$

As t goes to $+\infty$, the absolute value of the coefficient

$$\left(\frac{e^{-\lambda_{\sigma_k}(x_k)t} - 1}{\lambda_{\sigma_k}(x_k)} \, I_{\{\lambda_{\sigma_k}(x_k) \neq 0\}} - t\, I_{\{\lambda_{\sigma_k}(x_k)=0\}} \right) v_{\sigma_k}(x_k)^T \nabla F(x_k)$$

of $v_{\sigma_k}(x_k)$ in (5) tends to $+\infty$, dominating the absolute values of the analogous coefficients of all other $v_i(x_k)$, $i \in (\{1,\ldots,n\} - \{\sigma_k\})$. This implies

$$\lim_{t\to+\infty} q_k(t) = -\text{sign}\left(\nabla F(x_k)^T v_{\sigma_k}(x_k)\right) v_{\sigma_k}(x_k) \lim_{t\to+\infty} \psi_k(t), \tag{8}$$

where $\psi_k(t)$ is a continuous real-valued function such that

$$\lim_{t\to+\infty} \psi_k(t) = +\infty.$$

Therefore, under Assumption 2.1 and Assumption 2.2, if $\nabla^2 F(x_k)$ is not positive definite, then as t goes to $+\infty$, the curve $x_k + q_k(t)$ tends to $+\infty$ along the direction of an eigenvector of $\nabla^2 F(x_k)$ corresponding to the smallest eigenvalue that has eigenvectors with a component in the direction of the gradient $\nabla F(x_k)$.

If we remove Assumption 2.2, then the set $\mathcal{E}_k$ defined by (6) may not have a unique minimum any more. However, we can analyze the limit values of $x_k + q_k(t)$ simply in a few more steps. First, define the following index sets:

$$\mathcal{I}_k = \operatorname*{argmin}_{i \in \{1,\ldots,n\}} \left\{ \lambda_i(x_k) : \lambda_i(x_k) \leq 0,\, v_i(x_k)^T \nabla F(x_k) \neq 0 \right\}, \tag{9}$$

$$\mathcal{K}_k = \left\{ i : i \in \{1,\ldots,n\},\, \lambda_i(x_k) \leq 0,\, v_i(x_k)^T \nabla F(x_k) = 0 \right\}, \tag{10}$$

$$\mathcal{J}_k = \{1,\ldots,n\} - (\mathcal{I}_k \cup \mathcal{K}_k). \tag{11}$$

Clearly, $\{\mathcal{I}_k, \mathcal{J}_k, \mathcal{K}_k\}$ forms a partition of $\{1,\ldots,n\}$. We now define

$$\sigma_k = \max\, \mathcal{I}_k. \tag{12}$$

We then extend the real affine space $\mathbf{R}^n$ into the corresponding real *projective* space $\mathbf{R}_\pi^{n+1}$ and introduce the *homogeneous* Cartesian coordinates. Hence we transform x_k and $q_k(t)$ into

$$\hat{x}_k = \begin{pmatrix} (\hat{x}_k)_{1:n} \\ (\hat{x}_k)_{n+1} \end{pmatrix} = \begin{pmatrix} \dfrac{x_k}{t} \\ \dfrac{1}{t} \end{pmatrix} \text{ for any } t > 0, \quad \hat{q}_k(0) = \begin{pmatrix} (\hat{q}_k(0))_{1:n} \\ (\hat{q}_k(0))_{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \text{and}$$

$$\hat{q}_k(t) = \begin{pmatrix} \eta_k\, q_k(t) \left[\dfrac{\lambda_{\sigma_k}(x_k)}{e^{-\lambda_{\sigma_k}(x_k)t} - 1}\, I_{\{\lambda_{\sigma_k}(x_k) \neq 0\}} - \dfrac{1}{t}\, I_{\{\lambda_{\sigma_k}(x_k) = 0\}} \right] \\ \eta_k \left[\dfrac{\lambda_{\sigma_k}(x_k)}{e^{-\lambda_{\sigma_k}(x_k)t} - 1}\, I_{\{\lambda_{\sigma_k}(x_k) \neq 0\}} - \dfrac{1}{t}\, I_{\{\lambda_{\sigma_k}(x_k) = 0\}} \right] \end{pmatrix} \text{ for } t \in \,]0, +\infty[\,, \tag{13}$$

where

$$\eta_k = -\frac{1}{\left\| \sum_{i \in \mathcal{I}_k} v_i(x_k) v_i(x_k)^T \nabla F(x_k) \right\|_2} < 0. \tag{14}$$

It is easy to see that $\hat{x}_k$ and $\hat{q}_k(t)$ in (13) and (14) are well defined homogeneous Cartesian coordinates of *proper points* because $(\hat{x}_k)_{n+1} \neq 0$, $(\hat{q}_k(t))_{n+1} \neq 0$ for all $t \geq 0$,

$$x_k = \frac{(\hat{x}_k)_{1:n}}{(\hat{x}_k)_{n+1}} \quad \text{and} \quad q_k(t) = \frac{(\hat{q}_k(t))_{1:n}}{(\hat{q}_k(t))_{n+1}} \quad \forall t \in [0, +\infty[\,.$$

When $t > 0$, the first n components $(\hat{q}_k(t))_{1:n}$ of $\hat{q}_k(t)$ are

$$\begin{aligned}
&\eta_k\, q_k(t) \left[\frac{\lambda_{\sigma_k}(x_k)}{e^{-\lambda_{\sigma_k}(x_k)t} - 1}\, I_{\{\lambda_{\sigma_k}(x_k) \neq 0\}} - \frac{1}{t}\, I_{\{\lambda_{\sigma_k}(x_k) = 0\}} \right] \\
&= \left[\sum_{i=1}^{n} \left(\frac{e^{-\lambda_i(x_k)t} - 1}{\lambda_i(x_k)}\, I_{\{\lambda_i(x_k) \neq 0\}} - t\, I_{\{\lambda_i(x_k) = 0\}} \right) v_i(x_k) v_i(x_k)^T \nabla F(x_k) \right] \\
&\quad \left[\frac{\lambda_{\sigma_k}(x_k)}{e^{-\lambda_{\sigma_k}(x_k)t} - 1}\, I_{\{\lambda_{\sigma_k}(x_k) \neq 0\}} - \frac{1}{t}\, I_{\{\lambda_{\sigma_k}(x_k) = 0\}} \right] \eta_k \\
&= \sum_{i=1}^{n} \left[\frac{\lambda_{\sigma_k}(x_k) \left(e^{-\lambda_i(x_k)t} - 1 \right)}{\lambda_i(x_k) \left(e^{-\lambda_{\sigma_k}(x_k)t} - 1 \right)}\, I_{\{\lambda_i(x_k) \neq 0,\, \lambda_{\sigma_k}(x_k) \neq 0\}} - \frac{\lambda_{\sigma_k}(x_k)t}{e^{-\lambda_{\sigma_k}(x_k)t} - 1}\, I_{\{\lambda_i(x_k) = 0,\, \lambda_{\sigma_k}(x_k) \neq 0\}} \right.
\end{aligned}$$

$$- \left. \frac{e^{-\lambda_i(x_k)t}-1}{\lambda_i(x_k)t} I_{\{\lambda_i(x_k)\neq 0,\, \lambda_{\sigma_k}(x_k)=0\}} + I_{\{\lambda_i(x_k)=0,\, \lambda_{\sigma_k}(x_k)=0\}} \right] v_i(x_k)v_i(x_k)^T \nabla F(x_k)\eta_k.$$

Since all the factors $v_i(x_k)v_i(x_k)^T\nabla F(x_k)$ indexed by $\mathcal{K}_k$ are, by definition of $\mathcal{K}_k$, equal to zero, and since $I_{\{\lambda_{\sigma_k}(x_k)\neq 0\}} + I_{\{\lambda_{\sigma_k}(x_k)=0\}} = 1$, we have:

$$\begin{aligned}
(\hat{q}_k(t))_{1:n} = &\left[\sum_{i\in\mathcal{I}_k} v_i(x_k)v_i(x_k)^T\nabla F(x_k)\eta_k\right] \\
&+ \sum_{i\in\mathcal{J}_k}\left[\frac{\lambda_{\sigma_k}(x_k)\left(e^{-\lambda_i(x_k)t}-1\right)}{\lambda_i(x_k)\left(e^{-\lambda_{\sigma_k}(x_k)t}-1\right)} I_{\{\lambda_i(x_k)\neq 0,\, \lambda_{\sigma_k}(x_k)\neq 0\}}\right. \\
&\qquad - \frac{\lambda_{\sigma_k}(x_k)t}{e^{-\lambda_{\sigma_k}(x_k)t}-1} I_{\{\lambda_i(x_k)=0,\, \lambda_{\sigma_k}(x_k)\neq 0\}} \\
&\qquad \left. - \frac{e^{-\lambda_i(x_k)t}-1}{\lambda_i(x_k)t} I_{\{\lambda_i(x_k)\neq 0,\, \lambda_{\sigma_k}(x_k)=0\}}\right] v_i(x_k)v_i(x_k)^T\nabla F(x_k)\eta_k.
\end{aligned} \tag{15}$$

As t goes to $+\infty$, the second summation in (15) tends to zero, because $\lambda_i(x_k) > \lambda_{\sigma_k}(x_k)$ for all $i \in \mathcal{J}_k$. Moreover, from (13) and (14) we see that the last component $(\hat{q}_k(t))_{n+1}$ of $\hat{q}_k(t)$ also tends to zero as t goes to $+\infty$. Thus

$$\begin{aligned}
\lim_{t\to+\infty}\left[\hat{x}_k + \hat{q}_k(t)\right] &= \lim_{t\to+\infty}\begin{pmatrix} \dfrac{x_k}{t} + (\hat{q}_k(t))_{1:n} \\ \dfrac{1}{t} + (\hat{q}_k(t))_{n+1}\end{pmatrix} \\
&= \begin{pmatrix} \sum_{i\in\mathcal{I}_k} v_i(x_k)v_i(x_k)^T\nabla F(x_k)\eta_k \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} -\dfrac{\sum_{i\in\mathcal{I}_k} v_i(x_k)v_i(x_k)^T\nabla F(x_k)}{\left\|\sum_{i\in\mathcal{I}_k} v_i(x_k)v_i(x_k)^T\nabla F(x_k)\right\|_2} \\ 0 \end{pmatrix} = q_\infty(x_k).
\end{aligned} \tag{16}$$

Since the first n components of $q_\infty(x_k)$ are not all equal to zero, while the last one is zero, equality (16) defines an *improper point* of the real projective space $\mathbf{R}_\pi^{n+1}$.

Such an improper point is the direction of the directed line of $\mathbf{R}^n$ represented by the following unit vector:

$$v(x_k) = -\frac{\sum_{i \in \mathcal{I}_k} v_i(x_k) v_i(x_k)^T \nabla F(x_k)}{\sqrt{\sum_{i \in \mathcal{I}_k} \left[v_i(x_k)^T \nabla F(x_k)\right]^2}} \tag{17}$$

When the set $\mathcal{I}_k$ is a singleton, $v(x_k)$ coincides with the direction given by (8), that is, the direction of an eigenvector of $\nabla^2 F(x_k)$ corresponding to the smallest eigenvalue that has eigenvectors with a component in the direction of the gradient $\nabla F(x_k)$.

Therefore, under Assumption 2.1 only, if the Hessian matrix $\nabla^2 F(x_k)$ is not positive definite, then the curve $x_k + q_k(t)$ is *unbounded* and, as t goes to $+\infty$, it tends to $+\infty$ along a direction that results from a linear combination of orthonormal eigenvectors of $\nabla^2 F(x_k)$ corresponding to the smallest eigenvalue that has an eigenvector with a component in the direction of the gradient $\nabla F(x_k)$. Figure 2.3 illustrates this case.

Finally, we also remove Assumption 2.1. If at any iterate x_k where the Hessian matrix $\nabla^2 F(x_k)$ is not positive definite the gradient $\nabla F(x_k)$ *has no component* in the eigenspace of nonpositive eigenvalues of $\nabla^2 F(x_k)$, then there is an index $n_k \in$

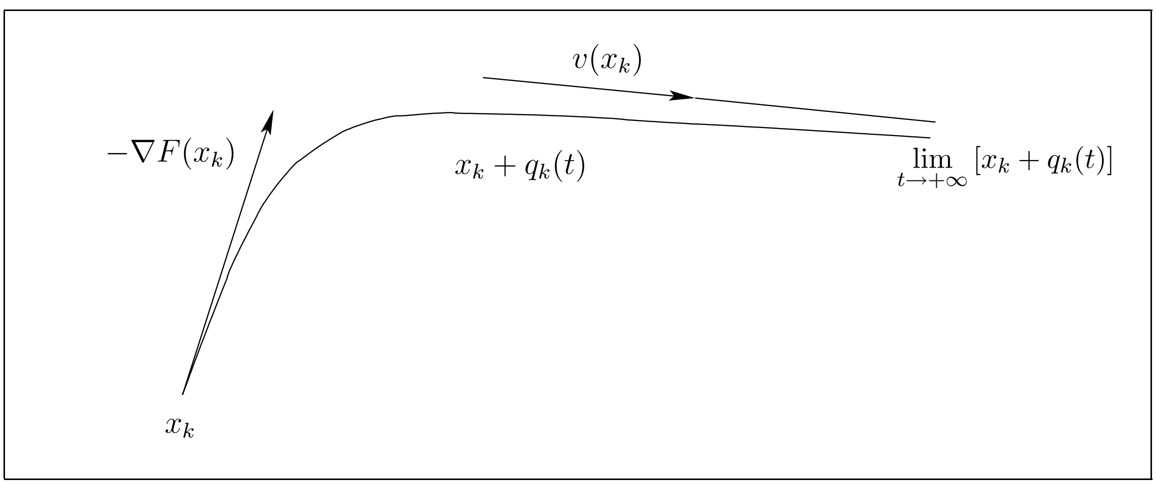

Figure 2.3: Search curve when $\nabla^2 F(x_k)$ is *not* positive definite.

$\{1, \ldots, n-1\}$ such that

$$\lambda_i(x_k) > 0 \quad \forall i \in \{1, \ldots, n_k\} \quad \text{and} \quad v_i(x_k)^T \nabla F(x_k) = 0 \quad \forall i \in \{n_k+1, \ldots, n\}. \tag{18}$$

Therefore, from (5) and (18) we get

$$q_k(t) = \sum_{i=1}^{n_k} \frac{e^{-\lambda_i(x_k)t} - 1}{\lambda_i(x_k)} v_i(x_k) v_i(x_k)^T \nabla F(x_k)$$

and define direction $d(x_k)$ as

$$d(x_k) = \lim_{t \to +\infty} q_k(t) = -\sum_{i=1}^{n_k} \frac{v_i(x_k) v_i(x_k)^T}{\lambda_i(x_k)} \nabla F(x_k). \tag{19}$$

Equation (19) shows that the curve $x_k + q_k(t)$ is *bounded* as in the case in which $\nabla^2 F(x_k)$ is positive definite, even though its limit value is $-\nabla^2 F(x_k)^{-1} \nabla F(x_k)$ only when $\nabla^2 F(x_k)$ is not singular. In such a case, vector $d(x_k)$ given by (19) is indeed the unique solution of the following system of linear equations:

$$\nabla^2 F(x_k)\, z = -\nabla F(x_k). \tag{20}$$

On the other hand, if the Hessian matrix $\nabla^2 F(x_k)$ is singular, it is easy to see that the same vector $d(x_k)$ is the *minimum norm* solution of (20), a system of linear equations that is still *compatible* but now *undetermined*. As shown, for instance, in Strang [54], the minimum norm solution of (20) is

$$z_k^+ = -\nabla^2 F(x_k)^+ \, \nabla F(x_k), \tag{21}$$

where $\nabla^2 F(x_k)^+$ denotes the *pseudoinverse* of $\nabla^2 F(x_k)$. Let

$$\mathcal{V}_k = \{i : i \in \{1, \ldots, n\},\, \lambda_i(x_k) < 0\}$$

be the possibly empty set of the indexes of the negative eigenvalues. Moreover, let

$$\nabla^2 F(x_k) = V(x_k) \Sigma_k V_2^T(x_k) = \sum_{i=1}^{n} \lambda_i(x_k) v_i(x_k) v_i(x_k)^T \tag{22}$$

be the singular value decomposition of $\nabla^2 F(x_k)$; then its pseudoinverse is

$$\nabla^2 F(x_k)^+ = V_2(x_k)\Sigma_k^+ V^T(x_k) = \sum_{i=1}^{n_k} \frac{v_i(x_k)v_i(x_k)^T}{\lambda_i(x_k)} + \sum_{i\in\mathcal{V}_k} \frac{v_i(x_k)v_i(x_k)^T}{\lambda_i(x_k)}. \tag{23}$$

From (21) and (23) we have

$$\begin{aligned} z_k^+ &= -\left[\sum_{i=1}^{n_k} \frac{v_i(x_k)v_i(x_k)^T}{\lambda_i(x_k)} + \sum_{i\in\mathcal{V}_k} \frac{v_i(x_k)v_i(x_k)^T}{\lambda_i(x_k)}\right] \nabla F(x_k) \\ &= -\sum_{i=1}^{n_k} \frac{v_i(x_k)v_i(x_k)^T}{\lambda_i(x_k)} \nabla F(x_k) \\ &= d(x_k), \end{aligned}$$

where the second equality above is justified by the fact that the gradient $\nabla F(x_k)$ has no component in the eigenspace of nonpositive eigenvalues of $\nabla^2 F(x_k)$.

The preceding discussion has proved the following theorem.

Theorem 2.3 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$. For any point $x_k \in D$, the initial value problem (4) has solution $q_k(t)$ given by (5) for all $t \in [0, +\infty[$. At $t = 0$, the curve $x_k + q_k$ is tangent to the gradient vector $\nabla F(x_k)$ of function F at x_k. If the Hessian matrix $\nabla^2 F(x_k)$ is positive definite or if the gradient $\nabla F(x_k)$ has no component in the eigenspace of nonpositive eigenvalues of $\nabla^2 F(x_k)$, then the curve $x_k + q_k(t)$ is bounded and, as t goes to $+\infty$, its limit point is equal to x_k plus the minimum norm solution of the system of linear equations (20). Otherwise, the curve $x_k + q_k(t)$ is unbounded and, as t goes to $+\infty$, it tends to $+\infty$ along the direction $v(x_k)$ defined by (9) and (17), which is a linear combination of orthonormal eigenvectors of $\nabla^2 F(x_k)$ corresponding to the smallest eigenvalue that has an eigenvector with a component in the direction of $\nabla F(x_k)$.*

The properties stated in Theorem 2.3 are also shared by the solution of the trust-region subproblem as a function of the radius. As an example, Figure 2.4 on the next page illustrates the curve $p_k^*(\Delta_k)$, which the exact solution of the trust-region subproblem as a function of the trust-region radius Δ_k.

The method of gradients discussed in this section and the calculation of exact solutions of the trust-region subproblem can be practically employed to determine search directions only when the optimization problems are relatively small. In fact, the solution (5) of problem (4) requires at each iterate x_k the computation of the complete set of eigenvectors and eigenvalues of $\nabla^2 F(x_k)$, which in the worst case take $O(n^3)$ time, where n denotes the number of independent variables of the objective function.

However, both methods are convergent and share the same property of identifying at each iterate x_k a curve that has the steepest descent direction as a tangent at x_k,

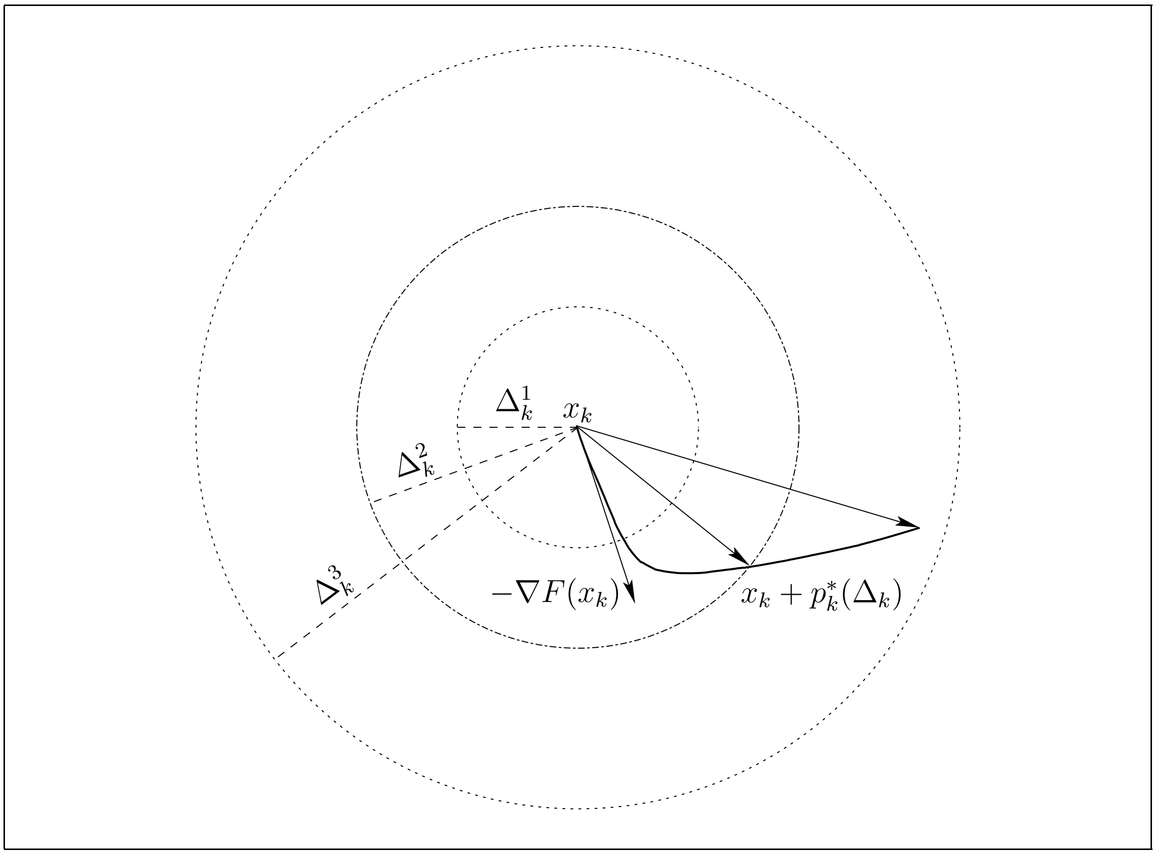

Figure 2.4: Solution $p_k^*(\Delta_k)$ of the trust-region subproblem as a function of the trust-region radius Δ_k.

and whose other extreme behaves as stated in Theorem 2.3. This suggests that what is important for convergence may be indeed these said properties.

2.2 The search curves

We now describe how to determine search curves in a descent method for problem (1) through the solution of a system of *only two* linear ordinary differential equations, so that the computation of the eigenvalues and eigenvectors required to solve it can be done analytically.

Modifying a terminology introduced by Moré and Sorenson [41], we give the following definition.

Definition 2.4 Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$.

i) A point $x \in D$ is called a *positive semidefinite point of F* if all eigenvalues of $\nabla^2 F(x)$ are nonnegative.

ii) A vector $d \in \mathbf{R}^n$ is called a *direction of negative curvature of F at the point $x \in D$* if x is not a positive semidefinite point of F and $d^T \nabla^2 F(x) d < 0$.

iii) A pair (s, d) of vectors in $\mathbf{R}^n \times \mathbf{R}^n$ is called a *nonascent pair of F at the point $x \in D$* when

$$\left.\begin{array}{l} \nabla F(x)^T s \leq 0 \\ \nabla F(x)^T d \leq 0 \\ d^T \nabla^2 F(x) d < 0 \end{array}\right\} \quad \text{if } x \text{ is } \textit{not} \text{ a positive semidefinite point of } F, \tag{24}$$

$$\left.\begin{array}{l} \nabla F(x)^T s < 0 \\ \nabla F(x)^T d \leq 0 \end{array}\right\} \quad \text{if } x \text{ is a positive semidefinite point of } F. \tag{25}$$

Note that a nonascent pair of F does not exist at a point $x \in D$ if and only if $\nabla F(x) = 0$ and $\nabla^2 F(x)$ is positive semidefinite. Of course, this is just the kind of point that we would like our minimization algorithm to find.

We shall consider a descent method similar in spirit to the original method of gradients and to Behrman's method. We shall generate a sequence $(x_k)_{k=0}^{+\infty}$ of iterates in $D \subseteq \mathbf{R}^n$ and prove that it converges to a point $x^* \in D$ that satisfies the first and second order necessary conditions for local unconstrained optimality.

Starting from $x_0 \in D$, at each iterate x_k where the gradient vector $\nabla F(x_k) \neq 0$ and the Hessian matrix $\nabla^2 F(x_k)$ is not positive semidefinite, we define a nonascent pair (s_k, d_k) of F by

$$s_k = -\nabla F(x_k),$$

$$d_k = \begin{cases} -\nabla^2 F(x_k)^+ \, \nabla F(x_k) & \text{if } \nabla^2 F(x_k) \text{ is positive definite or} \\ & v_i(x_k)^T \nabla F(x_k) = 0 \ \forall i \in \{n_k+1, \ldots, n\} \\ -\dfrac{\sum\limits_{i \in \mathcal{I}_k} v_i(x_k) v_i(x_k)^T \nabla F(x_k)}{\sqrt{\sum\limits_{i \in \mathcal{I}_k} [v_i(x_k)^T \nabla F(x_k)]^2}} & \text{otherwise,} \end{cases} \tag{26}$$

in which n_k and $\mathcal{I}_k$ are defined by (9) and (18), respectively. Instead, at any x_k at which $\nabla F(x_k) = 0$ but $\nabla^2 F(x_k)$ is not positive semidefinite we define a nonascent pair (s_k, d_k) of F by

$$\begin{aligned} s_k &= 0, \\ d_k &= \frac{\sum\limits_{i \in \mathcal{W}_k} v_i(x_k)}{\sqrt{\operatorname{card}(\mathcal{W}_k)}}, \end{aligned} \tag{27}$$

where

$$\mathcal{W}_k = \underset{i \in \{1, \ldots, n\}}{\operatorname{argmin}} \{\lambda_i(x_k)\}.$$

In fact, according to the Deformation Lemma, Theorem 1.6, the integral curve of $-\nabla F$ from a stationary point point $x_k \in D$ is $\Phi(x_k, t) = x_k$ for all $t \geq 0$. Therefore, $\sum\limits_{i \in \mathcal{W}_k} v_i(x_k)$ is the best direction to move away from the stationary x_k when the Hessian matrix $\nabla^2 F(x_k)$ is not positive semidefinite.

We then determine a search direction p_k as a linear combination of s_k and d_k, that is,

$$p_k = \bar{\vartheta}_1(x_k) s_k + \bar{\vartheta}_2(x_k) d_k \qquad \forall k \in \{0, 1, \ldots\},$$

with $\bar{\vartheta}_1(x_k)$, $\bar{\vartheta}_2(x_k) \in \mathbf{R}$. Finally, we use the relationship

$$x_{k+1} = \begin{cases} x_k + p_k & \text{if } x_k \text{ is not stationary or is not positive semidefinite} \\ x_k & \text{otherwise} \end{cases}$$

and proceed until certain termination criteria are satisfied.

In order to avoid dealing with the trivial case in which s_k and d_k are linearly dependent and the search curve reduces to a straight line, we make the following assumption.

Assumption 2.5 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$. The nonascent pair (s_k, d_k) of F defined at a point $x_k \in D$ by (26) is such that s_k and d_k are linearly independent.*

Under Assumption 2.5 we can always define two orthonormal vectors that lie in the subspace spanned by s_k and d_k. Let

$$\begin{aligned} w_1(x_k) &= \frac{1}{\|d_k\|_2 \sqrt{\|s_k\|_2^2 \, \|d_k\|_2^2 - (s_k^T d_k)^2}} \left[\|d_k\|_2^2 \, s_k - (s_k^T d_k) d_k \right], \\ w_2(x_k) &= \frac{d_k}{\|d_k\|_2}. \end{aligned} \tag{28}$$

In order to simplify the notation, we let

$$r_k = \sqrt{\|s_k\|_2^2 \, \|d_k\|_2^2 - (s_k^T d_k)^2}, \tag{29}$$

so we can write $w_1(x_k) = \frac{1}{\|d_k\|_2 \, r_k} \left[\|d_k\|_2^2 \, s_k - (s_k^T d_k) d_k \right]$. It is clear that

$$\|w_1(x_k)\|_2 = \|w_2(x_k)\|_2 = 1 \quad \text{and} \quad w_1(x_k)^T w_2(x_k) = 0.$$

Hence, the matrix $W_k = (w_1(x_k), \, w_2(x_k))$ is such that

$$W_k^T W_k = I_2. \tag{30}$$

Therefore, $w_1(x_k)$ and $w_2(x_k)$ form an *orthonormal basis* for the subspace spanned by (s_k, d_k). Moreover, for any vector $h \in \mathbf{R}^n$, $W_k^T h$ is the *two-dimensional projection* of h onto this subspace. In particular, for all $t \in [0, +\infty[\,$, $W_k^T\,[x_k + q_k(t)]$ is the two-dimensional projection of the search curve $x_k + q_k(t)$ onto the subspace spanned by the nonascent pair (s_k, d_k), as shown in Figure 2.5 for the case $\nabla^2 F(x_k)$ positive definite.

Finally, we let

$$G_k = W_k^T \nabla^2 F(x_k) W_k = \begin{pmatrix} w_1(x_k)^T \nabla^2 F(x_k) w_1(x_k) & w_1(x_k)^T \nabla^2 F(x_k) w_2(x_k) \\ w_1(x_k)^T \nabla^2 F(x_k) w_2(x_k) & w_2(x_k)^T \nabla^2 F(x_k) w_2(x_k) \end{pmatrix}. \tag{31}$$

Theorem 2.6 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$, and (s_k, d_k) the nonascent pair of F defined at a point $x_k \in D$ by (26) under Assumption 2.5. If the Hessian matrix $\nabla^2 F(x_k)$ is positive definite or the gradient $\nabla F(x_k)$ has no component in the eigenspace of nonpositive eigenvalues of $\nabla^2 F(x_k)$, then the matrix G_k defined by (31) is positive definite. Otherwise, G_k is not positive definite, one of its orthonormal eigenvector*

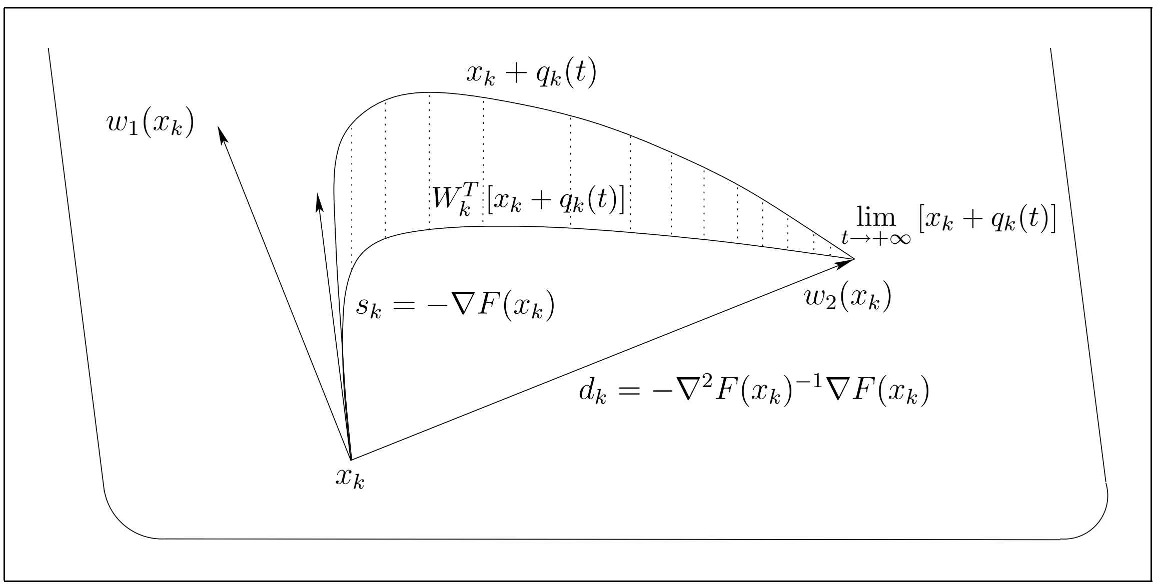

Figure 2.5: Projection of the search curve $x_k + q_k(t)$ onto the subspace spanned by the nonascent pair (s_k, d_k) when $\nabla^2 F(x_k)$ is positive definite.

matrix is the two-dimensional identity matrix and its smaller eigenvalue is equal to $\lambda_{\sigma_k}(x_k)$, which is the smallest eigenvalue of $\nabla^2 F(x_k)$ that has an eigenvector with a component in the direction of the gradient $\nabla F(x_k)$.

Proof. If $\nabla^2 F(x_k)$ is positive definite or the gradient $\nabla F(x_k)$ has no component in the eigenspace of nonpositive eigenvalues of $\nabla^2 F(x_k)$, then it follows from (26) that

$$\nabla^2 F(x_k) d_k = -\nabla F(x_k) = s_k.$$

Therefore, from (28) we have

$$\begin{aligned}
w_1(x_k)^T \nabla^2 F(x_k) w_1(x_k) &= \frac{1}{\|d_k\|_2\, r_k} w_1(x_k)^T \nabla^2 F(x_k) \left[\|d_k\|_2^2\, s_k - (s_k^T d_k) d_k\right] \\
&= \frac{1}{\|d_k\|_2\, r_k} w_1(x_k)^T \left[\|d_k\|_2^2\, \nabla^2 F(x_k) s_k - (s_k^T d_k) \nabla^2 F(x_k) d_k\right] \\
&= \frac{1}{\|d_k\|_2\, r_k} w_1(x_k)^T \left[\|d_k\|_2^2\, \nabla^2 F(x_k) s_k - (s_k^T d_k) s_k\right] \\
&= \frac{1}{\|d_k\|_2^2\, r_k^2} \left[\|d_k\|_2^2\, s_k - (s_k^T d_k) d_k\right] \left[\|d_k\|_2^2\, \nabla^2 F(x_k) s_k - (s_k^T d_k) s_k\right] \\
&= \frac{1}{\|d_k\|_2^2\, r_k^2} \left[\|d_k\|_2^4\, s_k^T \nabla^2 F(x_k) s_k - 2\|s_k\|_2^2\, \|d_k\|_2^2\, (s_k^T d_k) + (s_k^T d_k)^3\right],
\end{aligned}$$

where r_k is defined by (29); moreover,

$$\begin{aligned}
w_1(x_k)^T \nabla^2 F(x_k) w_2(x_k) &= w_1(x_k)^T \nabla^2 F(x_k) \frac{d_k}{\|d_k\|_2} \\
&= w_1(x_k)^T \frac{s_k}{\|d_k\|_2} \\
&= \frac{1}{\|d_k\|_2\, r_k} \left[\|d_k\|_2^2\, s_k - (s_k^T d_k) d_k\right]^T \frac{s_k}{\|d_k\|_2} \\
&= \frac{r_k}{\|d_k\|_2^2},
\end{aligned}$$

and

$$\begin{aligned} w_2(x_k)^T \nabla^2 F(x_k) w_2(x_k) &= w_2(x_k)^T \frac{d_k}{\|d_k\|_2} \\ &= \frac{d_k^T}{\|d_k\|_2} \frac{s_k}{\|d_k\|_2} \\ &= \frac{s_k^T d_k}{\|d_k\|_2^2}. \end{aligned}$$

Hence we can write

$$G_k = \begin{pmatrix} w_1(x_k)^T \nabla^2 F(x_k) w_1(x_k) & \frac{r_k}{\|d_k\|_2^2} \\ \frac{r_k}{\|d_k\|_2^2} & \frac{s_k^T d_k}{\|d_k\|_2^2} \end{pmatrix}. \tag{32}$$

Since the (real) symmetric matrix G_k is positive definite if and only if all its lower right submatrices have positive determinants, we now analyze the signs of $\frac{s_k^T d_k}{\|d_k\|_2^2}$ and $\det(G_k)$:

$$\frac{s_k^T d_k}{\|d_k\|_2^2} = -\frac{s_k^T}{\|d_k\|_2^2} \sum_{i=1}^{n_k} \frac{v_i(x_k) v_i(x_k)^T}{\lambda_i(x_k)} \nabla F(x_k) = \frac{1}{\|d_k\|_2^2} \sum_{i=1}^{n_k} \frac{\left[s_k^T v_i(x_k)\right]^2}{\lambda_i(x_k)} > 0,$$

because, by definition of n_k, $\lambda_i(x_k) > 0$ for all $i \in \{1, \ldots, n_k\}$ and there exists an $i \in \{1, \ldots, n_k\}$ such that $s_k^T v_i(x_k) \neq 0$; moreover,

$$\begin{aligned} \det(G_k) =& \frac{1}{\|d_k\|_2^2 \, r_k^2} \left[\|d_k\|_2^4 \, s_k^T \nabla^2 F(x_k) s_k - 2\|s_k\|_2^2 \, \|d_k\|_2^2 \, (s_k^T d_k) + (s_k^T d_k)^3 \right] \frac{s_k^T d_k}{\|d_k\|_2^2} \\ & - \frac{r_k^2}{\|d_k\|_2^4} \\ =& \frac{1}{\|d_k\|_2^4 \, r_k^2} \left[\|d_k\|_2^4 \, (s_k^T d_k) s_k^T \nabla^2 F(x_k) s_k - 2\|s_k\|_2^2 \, \|d_k\|_2^2 \, (s_k^T d_k)^2 + (s_k^T d_k)^4 \right] \\ & + \frac{1}{\|d_k\|_2^4 \, r_k^2} \left[-\|s_k\|_2^4 \, \|d_k\|_2^4 + 2\|s_k\|_2^2 \, \|d_k\|_2^2 \, (s_k^T d_k)^2 - (s_k^T d_k)^4 \right] \\ =& \frac{1}{r_k^2} \left[(s_k^T d_k) s_k^T \nabla^2 F(x_k) s_k - \|s_k\|_2^4 \right] \\ =& \frac{1}{r_k^2} \left\{ \sum_{i=1}^{n_k} \frac{\left[s_k^T v_i(x_k)\right]^2}{\lambda_i(x_k)} \sum_{i=1}^{n} \lambda_i(x_k) \left[s_k^T v_i(x_k)\right]^2 - \left[\sum_{i=1}^{n} (s_k^T v_i(x_k))^2 \right]^2 \right\} \end{aligned}$$

$$
\begin{aligned}
=&\frac{1}{r_k^2}\left\{\sum_{i=1}^{n_k-1}\sum_{j=i+1}^{n_k}\left[\frac{\lambda_i(x_k)}{\lambda_j(x_k)}+\frac{\lambda_j(x_k)}{\lambda_i(x_k)}\right]\left[s_k^T v_i(x_k)\right]^2\left[s_k^T v_j(x_k)\right]^2\right.\\
&\left.\quad+\sum_{i=1}^{n_k}\left[s_k^T v_i(x_k)\right]^4\right\}\\
&+\frac{1}{r_k^2}\left\{-\sum_{i=1}^{n_k}\left[s_k^T v_i(x_k)\right]^4-2\sum_{i=1}^{n_k-1}\sum_{j=i+1}^{n_k}\left[s_k^T v_i(x_k)\right]^2\left[s_k^T v_j(x_k)\right]^2\right\}\\
=&\frac{1}{r_k^2}\sum_{i=1}^{n_k-1}\sum_{j=i+1}^{n_k}\left[\frac{\lambda_i(x_k)}{\lambda_j(x_k)}+\frac{\lambda_j(x_k)}{\lambda_i(x_k)}-2\right]\left[s_k^T v_i(x_k)\right]^2\left[s_k^T v_j(x_k)\right]^2\\
=&\frac{1}{r_k^2}\sum_{i=1}^{n_k-1}\sum_{j=i+1}^{n_k}\frac{\left[\lambda_i(x_k)-\lambda_j(x_k)\right]^2}{\lambda_i(x_k)\lambda_j(x_k)}\left[s_k^T v_i(x_k)\right]^2\left[s_k^T v_j(x_k)\right]^2>0\,,
\end{aligned}
$$

since $\lambda_i(x_k) > 0 \ \forall i \in \{1, \ldots, n_k\}$, $s_k^T v_i(x_k) = 0 \ \forall \in \{n_k + 1, \ldots, n\}$ and, under Assumption 2.5, there exist at least two distinct indexes i and j in $\{1, \ldots, n_k\}$ such that both $s_k^T v_i(x_k) \neq 0$ and $s_k^T v_j(x_k) \neq 0$.

If the Hessian matrix $\nabla^2 F(x_k)$ is not positive definite and the gradient $\nabla F(x_k)$ has a component in the eigenspace of nonpositive eigenvalues of $\nabla^2 F(x_k)$, then it follows from (26) that

$$
d_k = -\frac{\sum\limits_{i\in\mathcal{I}_k} v_i(x_k)v_i(x_k)^T\nabla F(x_k)}{\sqrt{\sum\limits_{i\in\mathcal{I}_k}\left[v_i(x_k)^T\nabla F(x_k)\right]^2}}\ \text{, with } \|d_k\|_2 = 1.
$$

Let σ_k be the index defined by (9) and (12). Hence $\lambda_{\sigma_k}(x_k)$ denotes the smallest, possibly multiple, eigenvalue of $\nabla^2 F(x_k)$ that has an eigenvector with a component in the direction of the gradient $\nabla F(x_k)$. From (28) we have

$$
\begin{aligned}
w_1(x_k)^T\nabla^2 F(x_k)w_2(x_k) &= w_1(x_k)^T\nabla^2 F(x_k)d_k\\
&= \lambda_{\sigma_k}(x_k)\, w_1(x_k)^T d_k\\
&= 0,
\end{aligned}
$$

since $w_1(x_k)$ and d_k are orthogonal as it follows from (28) and (30). Analogously,

$$\begin{aligned} w_2(x_k)^T \nabla^2 F(x_k) w_2(x_k) &= w_2(x_k)^T \nabla^2 F(x_k) d_k \\ &= \lambda_{\sigma_k}(x_k)\, w_2(x_k)^T d_k \\ &= \lambda_{\sigma_k}(x_k)\, d_k^T d_k \\ &= \lambda_{\sigma_k}(x_k). \end{aligned}$$

Finally, if $\mathcal{J}_k$ is the index set defined by (10) and (11), then

$$\begin{aligned} w_1(x_k)^T \nabla^2 F(x_k) w_1(x_k) &= \frac{1}{r_k} w_1(x_k)^T \nabla^2 F(x_k) \left[s_k - (s_k^T d_k) d_k \right] \\ &= \frac{1}{r_k} w_1(x_k)^T \left[\nabla^2 F(x_k) s_k - (s_k^T d_k) \nabla^2 F(x_k) d_k \right] \\ &= \frac{1}{r_k} w_1(x_k)^T \left[\nabla^2 F(x_k) s_k - \lambda_{\sigma_k}(x_k)(s_k^T d_k) d_k \right] \\ &= \frac{1}{r_k} \left[w_1(x_k)^T \nabla^2 F(x_k) s_k - \lambda_{\sigma_k}(x_k)(s_k^T d_k) w_1(x_k)^T d_k \right] \\ &= \frac{1}{r_k} w_1(x_k)^T \nabla^2 F(x_k) s_k \\ &= \frac{1}{r_k^2} \left[s_k - (s_k^T d_k) d_k \right]^T \nabla^2 F(x_k) s_k \\ &= \frac{1}{r_k^2} \left[s_k^T \nabla^2 F(x_k) s_k - \lambda_{\sigma_k}(x_k) \left(s_k^T d_k \right)^2 \right] \\ &= \frac{1}{r_k^2} \left\{ \left[\sum_{i=1}^{n} \lambda_i(x_k) s_k^T v_i(x_k) v_i(x_k)^T s_k \right] - \lambda_{\sigma_k}(x_k) \sum_{i \in \mathcal{I}_k} \left[s_k^T v_i(x_k) \right]^2 \right\} \\ &= \frac{1}{r_k^2} \sum_{i \in \mathcal{J}_k} \lambda_i(x_k) \left[s_k^T v_i(x_k) \right]^2 . \end{aligned}$$

Hence, if $\nabla^2 F(x_k)$ is not positive definite and the gradient $\nabla F(x_k)$ has a component in the eigenspace of nonpositive eigenvalues of $\nabla^2 F(x_k)$, then

$$G_k = \begin{pmatrix} \dfrac{1}{r_k^2} \displaystyle\sum_{i \in \mathcal{J}_k} \lambda_i(x_k) \left[s_k^T v_i(x_k) \right]^2 & 0 \\ 0 & \lambda_{\sigma_k}(x_k) \end{pmatrix}, \tag{33}$$

from which we can trivially read off the two eigenvalues of G_k and the corresponding unit eigenvectors. We now have to show that $\lambda_{\sigma_k}(x_k)$ is the smaller eigenvalue of G_k. Indeed, since $\lambda_i > \lambda_{\sigma_k}$ $\forall i \in \mathcal{J}_k$, we have

$$\begin{aligned}\frac{1}{r_k^2}\sum_{i\in\mathcal{J}_k}\lambda_i(x_k)\left[s_k^T v_i(x_k)\right]^2 &> \frac{\lambda_{\sigma_k}(x_k)}{r_k^2}\sum_{i\in\mathcal{J}_k}\left[s_k^T v_i(x_k)\right]^2\\ &= \frac{\lambda_{\sigma_k}(x_k)}{r_k^2}\left\{\|s_k\|_2^2 - \sum_{i\in\mathcal{I}_k}\left[s_k^T v_i(x_k)\right]^2\right\}\\ &= \frac{\lambda_{\sigma_k}(x_k)}{\|s_k\|_2^2-(s_k^T d_k)^2}\left[\|s_k\|_2^2-(s_k^T d_k)^2\right]\\ &= \lambda_{\sigma_k}(x_k).\end{aligned}$$

■

At each iterate x_k we now consider the following system of two linear ordinary differential equation:

$$\begin{cases}\varphi_k'(t) &= -W_k^T\nabla F(x_k) - G_k\varphi_k(t)\\ \varphi_k(0) &= 0.\end{cases} \tag{34}$$

The solution of (34) is

$$\varphi_k(t) = \sum_{i=1}^{2}\left(\frac{e^{-\mu_i(x_k)t}-1}{\mu_i(x_k)}\,I_{\{\mu_i(x_k)\neq 0\}} - t\,I_{\{\mu_i(x_k)=0\}}\right)u_i(x_k)u_i(x_k)^T W_k^T\nabla F(x_k), \tag{35}$$

where $\mu_1(x_k) \geq \mu_2(x_k)$ are the (real) eigenvalues of the symmetric matrix G_k, and $u_1(x_k)$, $u_2(x_k)$ are the corresponding unit eigenvectors forming an orthonormal basis for the two-dimensional vector space.

Theorem 2.7 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$, and (s_k, d_k) the nonascent pair of F defined at a point $x_k \in D$ by (26) under Assumption 2.5. If s_k and d_k lie in the plane spanned by two orthogonal eigenvectors of $\nabla^2 F(x_k)$, then the solution $\varphi_k(t)$ in (35) of system (34) is the two-dimensional projection onto that plane of the solution $q_k(t)$ in (5) of system (4).*

Proof. Given the nonascent pair (s_k, d_k) at x_k, the two-dimensional projection of $q_k(t)$ onto the subspace spanned by (s_k, d_k) is $\zeta_k(t) = W_k^T q_k(t)$. Premultiplying (4) by W_k^T yields

$$\begin{cases} W_k^T q_k'(t) & = -W_k^T \nabla F(x_k) - W_k^T \nabla^2 F(x_k) q_k(t) \\ W_k^T q_k(0) & = 0. \end{cases}$$

Since $\zeta_k'(t) = W_k^T q_k'(t)$, the above equalities can be rewritten as

$$\begin{cases} \zeta_k'(t) & = -W_k^T \nabla F(x_k) - W_k^T \nabla^2 F(x_k) q_k(t) \\ \zeta_k(0) & = 0. \end{cases} \tag{36}$$

If vectors $w_1(x_k)$ and $w_2(x_k)$, and therefore (s_k, d_k), lie in the subspace spanned by two orthogonal eigenvectors of $\nabla^2 F(x_k)$, then the three matrices

$$W_k, \ \begin{pmatrix} W_k & \nabla^2 F(x_k) w_1(x_k) \end{pmatrix} \text{ and } \begin{pmatrix} W_k & \nabla^2 F(x_k) w_2(x_k) \end{pmatrix}$$

have the same rank, namely, two. Therefore, there exists a matrix X_k such that

$$\begin{aligned} & W_k X_k^T = -\nabla^2 F(x_k) W_k \\ \Leftrightarrow \quad & X_k W_k^T = -W_k^T \nabla^2 F(x_k) && (37) \\ \Rightarrow \quad & X_k W_k^T q_k(t) = -W_k^T \nabla^2 F(x_k) q_k(t) \qquad \forall t \in [0, +\infty[\\ \Leftrightarrow \quad & X_k \zeta_k(t) = -W_k^T \nabla^2 F(x_k) q_k(t) \qquad \forall t \in [0, +\infty[\,. && (38) \end{aligned}$$

Postmultiplying (37) by W_k yields

$$X_k W_k^T W_k = -W_k^T \nabla^2 F(x_k) W_k$$

and, since $W_k^T W_k = I_2$,

$$X_k = -W_k^T \nabla^2 F(x_k) W_k\,.$$

Substituting in (38) for X_k from the above expression lets us rewrite (36) as

$$\begin{cases} \zeta_k'(t) & = -W_k^T \nabla F(x_k) - W_k^T \nabla^2 F(x_k) W_k \zeta_k(t) \\ \zeta_k(0) & = 0. \end{cases} \tag{39}$$

Comparing (39) with (34), where G_k is defined by (31), shows that $\varphi_k(t) = \zeta_k(t)$.

■

In general, even though $\varphi_k(t)$ is not the two-dimensional projection of the solution $q_k(t)$ in (5) of the system of n linear ordinary differential equation defined in (4), we still have

$$\varphi_k'(t) = -\sum_{i=1}^{2} e^{-\mu_i(x_k)t} u_i(x_k) u_i(x_k)^T W_k^T \nabla F(x_k), \tag{40}$$

and

$$\varphi_k'(0) = -\sum_{i=1}^{2} u_i(x_k) u_i(x_k)^T W_k^T \nabla F(x_k) = -W_k^T \nabla F(x_k), \tag{41}$$

which is the projection of the steepest descent direction $-\nabla F(x_k)$ onto the plane of (s_k, d_k). Moreover, if $\nabla^2 F(x_k)$ is positive definite or the gradient $\nabla F(x_k)$ has no component in the eigenspace of non positive eigenvalues of $\nabla^2 F(x_k)$, then, by Theorem 2.6, G_k is positive definite, and thus

$$\lim_{t \to +\infty} \varphi_k(t) = -\sum_{i=1}^{2} \frac{u_i(x_k) u_i(x_k)^T}{\mu_i(x_k)} W_k^T \nabla F(x_k) = -G_k^{-1} W_k^T \nabla F(x_k). \tag{42}$$

We are now able to define a search curve from x_k having the same properties of $x_k + q_k(t)$ stated in Theorem 2.3, but requiring the solution of a system of only two linear ordinary differential equations. At each iterate x_k, after computing $\varphi_k(t)$ according to (35), we let

$$\vartheta_k(t) = N_k \varphi_k(t), \tag{43}$$

where

$$N_k = \begin{pmatrix} \frac{\|d_k\|_2}{r_k} & 0 \\ -\frac{s_k^T d_k}{\|d_k\|_2 \, r_k} & \frac{1}{\|d_k\|_2} \end{pmatrix}. \tag{44}$$

The desired search curve from a point $x_k \in D$ is $x_k + p_k(t)$, where

$$\boxed{p_k(t) = \begin{cases} (s_k, d_k)\ \vartheta_k(t) & \text{if } x_k \text{ is not a stationary point of } F, \\ d_k\, t & \text{if } x_k \text{ is stationary and not positive semidefinite.} \end{cases}} \tag{45}$$

Theorem 2.8 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$, and (s_k, d_k) the nonascent pair of F defined at a point $x_k \in D$ by (26) under Assumption 2.5. The curve $x_k + p_k(t)$ defined by (35), (43), (44) and (45). At $t = 0$, the curve $x_k + p_k$ is tangent to the steepest descent direction at x_k. If the Hessian matrix $\nabla^2 F(x_k)$ is positive definite or the gradient $\nabla F(x_k)$ has no component in the eigenspace of non positive eigenvalues of $\nabla^2 F(x_k)$, then the curve $x_k + p_k(t)$ is bounded and, as t goes to $+\infty$, its limit point is equal to x_k plus the minimum norm solution of the system of linear equations (20). Otherwise, the curve $x_k + p_k(t)$ is unbounded and, as t goes to $+\infty$, it tends to $+\infty$ along the direction $v(x_k)$ defined by (9) and (17), which is a linear combination of orthonormal eigenvectors of $\nabla^2 F(x_k)$ corresponding to the smallest eigenvalue that has an eigenvector with a component in the direction of $\nabla F(x_k)$ the direction of the eigenvector corresponding to the smallest eigenvalue of $\nabla^2 F(x_k)$.*

Proof. The components of $-W_k^T \nabla F(x_k)$ are

$$-w_1^T(x_k)\nabla F(x_k) = w_1^T(x_k) s_k = \frac{r_k}{\|d_k\|_2}, \tag{46}$$

where r_k is defined by (29), and

$$-w_2^T(x_k)\nabla F(x_k) = w_2^T(x_k) s_k = \frac{s_k^T d_k}{\|d_k\|_2}. \tag{47}$$

Now, from (41) we get

$$\begin{aligned} p_k'(0) &= (s_k, d_k)\vartheta_k'(0) = (s_k, d_k) N_k \varphi_k'(0) = -(s_k, d_k) N_k W_k^T \nabla F(x_k) \\ &= (s_k, d_k) \begin{pmatrix} \frac{\|d_k\|_2}{r_k} \frac{r_k}{\|d_k\|_2} \\ -\frac{s_k^T d_k}{\|d_k\|_2 \, r_k} \frac{r_k}{\|d_k\|_2} + \frac{1}{\|d_k\|_2} \frac{s_k^T d_k}{\|d_k\|_2} \end{pmatrix} = (s_k, d_k) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = s_k \\ &\Leftrightarrow \quad p_k'(0) = -\nabla F(x_k)\,. \end{aligned} \tag{48}$$

If $\nabla^2 F(x_k)$ is positive definite or the gradient $\nabla F(x_k)$ has no component in the eigenspace of non positive eigenvalues of $\nabla^2 F(x_k)$, then from (42) we have

$$\lim_{t\to+\infty} p_k(t) = (s_k, d_k) N_k \lim_{t\to+\infty} \varphi_k(t) = -(s_k, d_k) N_k G_k^{-1} W_k^T \nabla F(x_k). \tag{49}$$

From (32) we can calculate

$$G_k^{-1} = \det(G_k)^{-1} \begin{pmatrix} \frac{s_k^T d_k}{\|d_k\|_2^2} & -\frac{r_k}{\|d_k\|_2^2} \\ -\frac{r_k}{\|d_k\|_2^2} & w_1(x_k)^T \nabla^2 F(x_k) w_1(x_k) \end{pmatrix},$$

where

$$\det(G_k) = w_1(x_k)^T \nabla^2 F(x_k) w_1(x_k) \frac{s_k^T d_k}{\|d_k\|_2^2} - \frac{r_k^2}{\|d_k\|_2^4}.$$

Now (46) and (47) may be used to obtain

$$\begin{aligned} -G_k^{-1} W_k^T \nabla F(x_k) &= \det(G_k)^{-1} \begin{pmatrix} \frac{s_k^T d_k}{\|d_k\|_2^2} \frac{r_k}{\|d_k\|_2} - \frac{r_k}{\|d_k\|_2^2} \frac{s_k^T d_k}{\|d_k\|_2} \\ -\frac{r_k}{\|d_k\|_2^2} \frac{r_k}{\|d_k\|_2} + w_1(x_k)^T \nabla^2 F(x_k) w_1(x_k) \frac{s_k^T d_k}{\|d_k\|_2} \end{pmatrix} \\ &= \det(G_k)^{-1} \begin{pmatrix} 0 \\ \det(G_k)\ \|d_k\|_2 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ \|d_k\|_2 \end{pmatrix}. \end{aligned}$$

Finally, it follows from (44) that

$$-N_k G_k^{-1} W_k^T \nabla F(x_k) = \begin{pmatrix} 0 \\ \frac{1}{\|d_k\|_2}\ \|d_k\|_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

so that from (49) we get

$$\begin{aligned} & \lim_{t \to +\infty} p_k(t) = (s_k, d_k) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = d_k \\ \Leftrightarrow \quad & \lim_{t \to +\infty} p_k(t) = -\nabla^2 F(x_k)^+ \nabla F(x_k). \end{aligned}$$

If $\nabla^2 F(x_k)$ is not positive definite and the gradient $\nabla F(x_k)$ has a component in the eigenspace of non positive eigenvalues of $\nabla^2 F(x_k)$, then by Theorem 2.6 we may write

$$G_k = \begin{pmatrix} \mu_1(x_k) & 0 \\ 0 & \mu_2(x_k) \end{pmatrix},$$

with $\mu_2(x_k) = \lambda_{\sigma_k}(x_k) \leq 0$. Clearly, the two orthonormal eigenvectors of G_k are

$$u_1(x_k) = e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } u_2(x_k) = e_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

so that from (35) and (43) we have

$$\begin{aligned} \vartheta_k(t) &= N_k \varphi_k(t) \\ &= N_k \sum_{i=1}^{2} \left(\frac{e^{-\mu_i(x_k)t} - 1}{\mu_i(x_k)} I_{\{\mu_i(x_k) \neq 0\}} - t\, I_{\{\mu_i(x_k)=0\}} \right) e_i e_i^T W_k^T \nabla F(x_k). \end{aligned} \tag{50}$$

In order to simplify the notation, we let

$$\gamma_{k,1}(t) = -\frac{e^{-\mu_1(x_k)t} - 1}{\mu_1(x_k)} I_{\{\mu_1(x_k) \neq 0\}} + t\, I_{\{\mu_1(x_k)=0\}}$$

and

$$\gamma_{k,2}(t) = -\frac{e^{-\lambda_{\sigma_k}(x_k)t} - 1}{\lambda_{\sigma_k}(x_k)} I_{\{\lambda_{\sigma_k}(x_k) < 0\}} + t\, I_{\{\lambda_{\sigma_k}(x_k)=0\}},$$

where

$$\lim_{t \to +\infty} \gamma_{k,1}(t) = \begin{cases} \frac{1}{\mu_1(x_k)} & \text{if } \mu_1(x_k) > 0 \\ +\infty & \text{otherwise} \end{cases},$$

$$\lim_{t \to +\infty} \gamma_{k,2}(t) = +\infty.$$

Recalling that if $\nabla^2 F(x_k)$ is not positive definite and the gradient $\nabla F(x_k)$ has a component in the eigenspace of non positive eigenvalues of $\nabla^2 F(x_k)$, then $\|d_k\|_2 = 1$, (46) and (47) may be used to obtain

$$e_1 e_1^T W_k^T \nabla F(x_k) = -\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} r_k \\ s_k^T d_k \end{pmatrix} = -\begin{pmatrix} r_k \\ 0 \end{pmatrix},$$

$$e_2 e_2^T W_k^T \nabla F(x_k) = -\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} r_k \\ s_k^T d_k \end{pmatrix} = -\begin{pmatrix} 0 \\ s_k^T d_k \end{pmatrix}.$$

Finally, it follows from (44) that

$$N_k e_1 e_1^T W_k^T \nabla F(x_k) = -\begin{pmatrix} 1 \\ -s_k^T d_k \end{pmatrix},$$
$$N_k e_2 e_2^T W_k^T \nabla F(x_k) = -\begin{pmatrix} 0 \\ s_k^T d_k \end{pmatrix}.$$

Substituting the above expressions in (50) yields

$$\vartheta_k(t) = \gamma_{k,1}(t) \begin{pmatrix} 1 \\ -s_k^T d_k \end{pmatrix} + \gamma_{k,2}(t) \begin{pmatrix} 0 \\ s_k^T d_k \end{pmatrix}.$$

Hence

$$\begin{aligned} \lim_{t\to+\infty} p_k(t) &= \lim_{t\to+\infty} (s_k, d_k)\vartheta_k(t) \\ &= \lim_{t\to+\infty} \gamma_{k,1}(t) s_k + \lim_{t\to+\infty} [\gamma_{k,2}(t) - \gamma_{k,1}(t)](s_k^T d_k) d_k\,. \end{aligned}$$

By Theorem 2.6 we have that $\mu_1(x_k) > \lambda_{\sigma_k}(x_k)$, which implies

$$\lim_{t\to+\infty} \frac{\gamma_{k,2}(t)}{\gamma_{k,1}(t)} = +\infty.$$

Therefore, since $(s_k^T d_k) > 0$, we may write

$$\lim_{t\to+\infty} p_k(t) = v(x_k) \lim_{t\to+\infty} \psi_k(t),$$

where $\psi_k(t)$ is a continuous real-valued function such that $\lim_{t\to+\infty} \psi_k(t) = +\infty$.

■

2.3 Properties of the search curves

At a point $x_k \in D$, let Γ_k denote the curve in $\mathbf{R}^n$ represented by the vector equation

$$y_k(t) = x_k + p_k(t), \tag{51}$$

where $p_k(t)$ is defined as in the preceding section. To simplify the notation, we let

$$\begin{aligned} \alpha_1(x_k) =& u_1(x_k)^T W_k^T \nabla F(x_k), \\ \alpha_2(x_k) =& u_2(x_k)^T W_k^T \nabla F(x_k). \end{aligned} \tag{52}$$

Note that, when s_k and d_k are linearly independent, the scalar products $\alpha_1(x_k)$ and $\alpha_2(x_k)$ are *never both equal to zero.* In fact, since $\|u_1(x_k)\|_2 = \|u_2(x_k)\|_2 = 1$ and $u_1(x_k)^T u_2(x_k) = 0$, we have that $\alpha_1(x_k)$ and $\alpha_2(x_k)$ are both equal to zero if and only if $W_k^T \nabla F(x_k) = 0$, because the vectors $u_1(x_k)$, $u_2(x_k)$ and $W_k^T \nabla F(x_k)$ cannot be pairwise orthogonal in $\mathbf{R}^2$. Moreover, from (46) and (47) it is clear that if s_k and d_k are linearly independent, then $W_k^T \nabla F(x_k) \neq 0$. We now show some properties of the curves Γ_k.

Theorem 2.9 *If Γ_k is a curve in $\mathbf{R}^n$ represented by vector equation (51), then Γ_k is an open simple curve:*

$$y_k(t') \neq y_k(t'') \qquad \forall\, t',\, t'' \geq 0,\, t' \neq t''.$$

Therefore, vector equation (51) gives a *parametric representation* of curve Γ_k.

Proof. Without loss of generality, we assume that Γ_k is not a straight line. Let t' and t'' two nonnegative real numbers such that $t' \neq t''$. Assume that $y_k(t') = y_k(t'')$. This implies

$$\begin{array}{lll} & p_k(t') = p_k(t'') & \\ \Rightarrow & \vartheta_k(t') = \vartheta_k(t'') & \text{from (45) and since } s_k,\, d_k \text{ are linearly independent} \\ \Rightarrow & N_k \varphi_k(t') = N_k \varphi_k(t'') & \text{from (43)} \\ \Rightarrow & \varphi_k(t') = \varphi_k(t'') & \text{because } N_k \text{ is nonsingular.} \end{array} \tag{53}$$

Equation (35), which defines $\varphi_k(t)$, shows that equality (53) above, together with the fact that $u_1(x_k)$ and $u_2(x_k)$ are orthogonal, implies

$$\begin{aligned} e^{-\mu_i(x_k)t'} = e^{-\mu_i(x_k)t''} & \quad \text{if } \mu_i(x_k) \neq 0, \\ t' = t'' & \quad \text{if } \mu_i(x_k) = 0, \end{aligned}$$

which is clearly a contradiction. ■

Theorem 2.10 *If Γ_k is a curve in $\mathbf{R}^n$ and equation (51) is a parametric representation of Γ_k, with base interval $[0, +\infty[$, then Γ_k is a regular (smooth) curve.*

Therefore, we say that vector equation (51) is a *regular* parametric representation of curve Γ_k.

Proof. If x_k is a stationary point, then from (45) and (51) we obtain

$$\|y_k'(t)\|_2^2 = \|p_k'(t)\|_2^2 = \frac{\left[\sum_{i\in\mathcal{W}_k} v_i(x_k)^T\right]\left[\sum_{i\in\mathcal{W}_k} v_i(x_k)\right]}{\text{card}\,(\mathcal{W}_k)} = 1 > 0. \tag{54}$$

If x_k is not a stationary point, then observe that from (40) we get

$$\begin{aligned}\|\varphi_k'(t)\|_2^2 &= \left\|-\sum_{i=1}^{2} e^{-\mu_i(x_k)t} u_i(x_k) u_i(x_k)^T W_k^T \nabla F(x_k)\right\|_2^2 \\ &= \left[e^{-\mu_1(x_k)t} u_1(x_k)^T W_k^T \nabla F(x_k)\right]^2 + \left[e^{-\mu_2(x_k)t} u_2(x_k)^T W_k^T \nabla F(x_k)\right]^2 \\ &= [\alpha_1(x_k)]^2 e^{-2\mu_1(x_k)t} + [\alpha_2(x_k)]^2 e^{-2\mu_2(x_k)t},\end{aligned} \tag{55}$$

where $\alpha_1(x_k)$ and $\alpha_2(x_k)$ are defined by (52). Moreover, (29) and (44) may be used to obtain

$$\begin{aligned}N_k^T (s_k, d_k)^T (s_k, d_k)\, N_k &= N_k^T \begin{pmatrix} \|s_k\|_2^2 & s_k^T d_k \\ s_k^T d_k & \|d_k\|_2^2 \end{pmatrix} N_k \\ &= N_k^T \begin{pmatrix} \|s_k\|_2^2 & s_k^T d_k \\ s_k^T d_k & \|d_k\|_2^2 \end{pmatrix} \begin{pmatrix} \frac{\|d_k\|_2}{r_k} & 0 \\ -\frac{s_k^T d_k}{\|d_k\|_2\, r_k} & \frac{1}{\|d_k\|_2} \end{pmatrix} \\ &= N_k^T \begin{pmatrix} \frac{r_k}{\|d_k\|_2} & \frac{s_k^T d_k}{\|d_k\|_2} \\ 0 & \|d_k\|_2 \end{pmatrix} \\ &= \begin{pmatrix} \frac{\|d_k\|_2}{r_k} & -\frac{s_k^T d_k}{\|d_k\|_2\, r_k} \\ 0 & \frac{1}{\|d_k\|_2} \end{pmatrix} \begin{pmatrix} \frac{r_k}{\|d_k\|_2} & \frac{s_k^T d_k}{\|d_k\|_2} \\ 0 & \|d_k\|_2 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2.\end{aligned} \tag{56}$$

Finally, (43), (45), (55) and (56) yield

$$\begin{aligned}\|y_k'(t)\|_2^2 =& \|p_k'(t)\|_2^2 \\ =& p_k'(t)^T p_k'(t) \\ =& \vartheta_k'(t)^T (s_k, d_k)^T (s_k, d_k)\, \vartheta_k'(t) \\ =& \varphi_k'(t)^T N_k^T (s_k, d_k)^T (s_k, d_k)\, N_k \varphi_k'(t) \\ =& \varphi_k'(t)^T I_2\, \varphi_k'(t) \\ =& \varphi_k'(t)^T \varphi_k'(t) \\ =& \|\varphi_k'(t)\|_2^2 \\ =& [\alpha_1(x_k)]^2 e^{-2\mu_1(x_k)t} + [\alpha_2(x_k)]^2 e^{-2\mu_2(x_k)t}. \end{aligned} \tag{57}$$

Since $\alpha_1(x_k)$ and $\alpha_2(x_k)$ are never both equal to zero, from (57) we may conclude

$$\|y_k'(t)\|_2 > 0 \qquad \forall t \in [0, +\infty[\,. \tag{58}$$

From (54), (58) and the observation that in the parametric equation (51) of Γ_k the function $p_k(t)$ is at least continuously differentiable, it follows that Γ_k is a regular curve. ■

Assume that the positive orientation fixed on Γ_k coincides with the positive orientation induced on Γ_k by the parametric representation (51), that is, with the orientation of increasing t. The *arc length parameter* of Γ_k, with base point $y_k(0) = x_k$, is the real-valued function $\varsigma_k : [0, +\infty[\, \subseteq \mathbf{R} \to \mathbf{R}$ defined in an interval of $\mathbf{R}$ by

$$\varsigma_k(t) = \int_0^t \|y_k'(\tau)\|_2 \, d\tau. \tag{59}$$

If x_k is not a stationary point, then it follows from (57) and (59) that

$$\varsigma_k(t) = \int_0^t \sqrt{[\alpha_1(x_k)]^2 e^{-2\mu_1(x_k)\tau} + [\alpha_2(x_k)]^2 e^{-2\mu_2(x_k)\tau}}\, d\tau. \tag{60}$$

Unfortunately, for general values of $\alpha_i(x_k)$ and $\mu_i(x_k)$ $(i = 1, 2)$, the integrand on the right-hand side of equation (60) does not have a primitive that can be expressed in elementary form. However, we can prove the following theorem.

Theorem 2.11 *Let Γ_k denote a curve in $\mathbf{R}^n$ and equation (51) be a regular parametric representation of Γ_k, with base interval $[0,+\infty[$, and x_k a nonstationary point. If the positive orientation fixed on Γ_k coincides with the positive orientation induced on Γ_k by the parametric representation (51), then the arc length parameter of Γ_k, with base point $y_k(0) = x_k$, is the continuous real-valued function $\varsigma_k : [0,+\infty[\subseteq \mathbf{R} \to \mathbf{R}$ defined as follows:*

$$
\begin{array}{ll}
|\alpha_i(x_k)|\, t & \text{if } \alpha_j(x_k) = \mu_i(x_k) = 0 \\
\dfrac{|\alpha_i(x_k)|}{\mu_i(x_k)} \left[1 - e^{-\mu_i(x_k)t}\right] & \text{if } \alpha_j(x_k) = 0,\ \mu_i(x_k) \neq 0 \\
\dfrac{1}{\mu_j(x_k)} \left[\varrho_k - \|p_k'(t)\|_2 + \dfrac{\alpha_i(x_k)}{2} \log\left|\dfrac{\alpha_i(x_k) + \|p_k'(t)\|_2}{\alpha_i(x_k) - \|p_k'(t)\|_2}\right|\right] & \text{if } \mu_i(x_k) = 0,\ \mu_j(x_k) \neq 0 \\
\dfrac{|\alpha_2(x_k)|}{\mu_2(x_k)} \left[H_k(1) - e^{-\mu_2(x_k)t}\, H_k\left(e^{-2[\mu_1(x_k)-\mu_2(x_k)]t}\right)\right] & \text{if } \mu_1(x_k) > \mu_2(x_k) > 0 \\
\dfrac{|\alpha_1(x_k)|}{\mu_1(x_k)} \left[K_k(1) - e^{-\mu_1(x_k)t}\, K_k\left(e^{2[\mu_1(x_k)-\mu_2(x_k)]t}\right)\right] & \text{if } \mu_2(x_k) < \mu_1(x_k) < 0 \\
\dfrac{1}{\mu_2(x_k)} \left[\Upsilon_k - \|p_k'(t)\|_2 + \dfrac{[\alpha_1(x_k)]^2}{2|\alpha_2(x_k)|\Lambda_k} e^{-[2\mu_1(x_k)-\mu_2(x_k)]t}\, M_k\left(e^{-2[\mu_1(x_k)-\mu_2(x_k)]t}\right)\right] & \\
 & \text{if } \mu_1(x_k) > 0,\ \mu_2(x_k) < 0,
\end{array}
$$

where $\|p_k'(t)\|_2$ is defined by (57), ϱ_k, H_k, K_k, Λ_k, M_k and Υ_k are defined by

$$\varrho_k = \|p_k'(0)\|_2 - \frac{\alpha_i(x_k)}{2} \log\left|\frac{\alpha_i(x_k) + \|p_k'(0)\|_2}{\alpha_i(x_k) - \|p_k'(0)\|_2}\right|, \tag{61}$$

$$H_k(u) = {}_2F_1\left(-\frac{1}{2}, \frac{\mu_2(x_k)}{2[\mu_1(x_k)-\mu_2(x_k)]}, \frac{\mu_2(x_k)}{2[\mu_1(x_k)-\mu_2(x_k)]} + 1, -\frac{[\alpha_1(x_k)]^2}{[\alpha_2(x_k)]^2}u\right), \tag{62}$$

$$K_k(u) = {}_2F_1\left(-\frac{1}{2}, \frac{-\mu_1(x_k)}{2[\mu_1(x_k)-\mu_2(x_k)]}, \frac{-\mu_1(x_k)}{2[\mu_1(x_k)-\mu_2(x_k)]} + 1, -\frac{[\alpha_2(x_k)]^2}{[\alpha_1(x_k)]^2}u\right), \tag{63}$$

$$\Lambda_k = \frac{2\mu_1(x_k) - \mu_2(x_k)}{2[\mu_1(x_k)-\mu_2(x_k)]}, \tag{64}$$

$$M_k(u) = {}_2F_1\left(\frac{1}{2}, \Lambda_k, \Lambda_k + 1, -\frac{[\alpha_1(x_k)]^2}{[\alpha_2(x_k)]^2}u\right), \tag{65}$$

$$\Upsilon_k = \|p'_k(0)\|_2 - \frac{[\alpha_1(x_k)]^2}{2|\alpha_2(x_k)|\Lambda_k} M_k(1), \tag{66}$$

and ${}_2F_1(\nu, \mu, \mu+1, z)$ *denotes the Gauss hypergeometric function with parameters* ν, μ *and* $\mu+1$.

Proof. Let i and j be elements of $\{1, 2\}$, with $i \neq j$. If $\alpha_j(x_k) = 0$ and $\mu_i(x_k) = 0$, then from (60) we get

$$\begin{aligned} \varsigma_k(t) &= \int_0^t \sqrt{[\alpha_i(x_k)]^2}\, d\tau \\ &= |\alpha_i(x_k)|\, t\,. \end{aligned}$$

Instead, if it is still the case that $\alpha_j(x_k) = 0$, but $\mu_i(x_k) \neq 0$, then (60) yields

$$\begin{aligned} \varsigma_k(t) &= \int_0^t \sqrt{[\alpha_i(x_k)]^2 e^{-2\mu_i(x_k)\tau}}\, d\tau \\ &= |\alpha_i(x_k)| \int_0^t e^{-\mu_i(x_k)\tau}\, d\tau \\ &= \frac{|\alpha_i(x_k)|}{\mu_i(x_k)} \left[1 - e^{-\mu_i(x_k)t}\right]. \end{aligned}$$

Consider now the case in which either $\mu_1(x_k)$ or $\mu_2(x_k)$ is different from zero, say $\mu_j(x_k)$, and the other one is equal to zero, say $\mu_i(x_k)$, with i and j elements of $\{1, 2\}$ and $i \neq j$. Let both $\alpha_1(x_k)$ and $\alpha_2(x_k)$ be nonzero. From (60) we get

$$\varsigma_k(t) = \int_0^t \sqrt{[\alpha_i(x_k)]^2 + [\alpha_j(x_k)]^2 e^{-2\mu_j(x_k)\tau}}\, d\tau. \tag{67}$$

Note that the integrand in (67) is continuous in the interval $I = [0, +\infty[\in \mathbf{R}$ and the real-valued function $g(\chi) = -\frac{1}{2\mu_j(x_k)} \log\left[\frac{\chi^2 - [\alpha_i(x_k)]^2}{[\alpha_j(x_k)]^2}\right]$ is continuously differentiable in the interval $J =]|\alpha_i(x_k)|, +\infty[\subset \mathbf{R}$. If $\mu_j(x_k) > 0$, then we consider the restriction g_1 of g to the interval $J_1 = \left]|\alpha_i(x_k)|, \sqrt{[\alpha_1(x_k)]^2 + [\alpha_2(x_k)]^2}\right]$. For every $\chi \in J_1$, we have

$$\frac{\chi^2 - [\alpha_i(x_k)]^2}{[\alpha_j(x_k)]^2} \leq \frac{[\alpha_1(x_k)]^2 + [\alpha_2(x_k)]^2 - [\alpha_i(x_k)]^2}{[\alpha_j(x_k)]^2} = \frac{[\alpha_j(x_k)]^2}{[\alpha_j(x_k)]^2} = 1,$$

so that

$$\log\left[\frac{\chi^2-[\alpha_i(x_k)]^2}{[\alpha_j(x_k)]^2}\right]\le 0 \quad\Rightarrow\quad g_1(\chi)\ge 0.$$

Analogously, if $\mu_j(x_k)<0$, then we consider the restriction g_2 of g to the interval $J_2=\left[\sqrt{[\alpha_1(x_k)]^2+[\alpha_2(x_k)]^2},+\infty\right[$. For every $\chi\in J_2$, we have

$$\frac{\chi^2-[\alpha_i(x_k)]^2}{[\alpha_j(x_k)]^2}\ge\frac{[\alpha_1(x_k)]^2+[\alpha_2(x_k)]^2-[\alpha_i(x_k)]^2}{[\alpha_j(x_k)]^2}=\frac{[\alpha_j(x_k)]^2}{[\alpha_j(x_k)]^2}=1,$$

so that

$$\log\left[\frac{\chi^2-[\alpha_i(x_k)]^2}{[\alpha_j(x_k)]^2}\right]\ge 0 \quad\Rightarrow\quad g_2(\chi)\ge 0.$$

Therefore, for every $\chi\in J_\iota$ ($\iota=1,2$), we have $g_\iota(\chi)\in I$, so that the composition $\sqrt{[\alpha_i(x_k)]^2+[\alpha_j(x_k)]^2e^{-2\mu_j(x_k)g_\iota(\chi)}}$ is a real-valued function defined in all J_ι. For all $t\in[0,+\infty[$, we have $\left[\sqrt{[\alpha_i(x_k)]^2+[\alpha_j(x_k)]^2e^{-2\mu_j(x_k)t}},\sqrt{[\alpha_1(x_k)]^2+[\alpha_2(x_k)]^2}\right]\subset J_1$ when $\mu_j(x_k)>0$, and $\left[\sqrt{[\alpha_1(x_k)]^2+[\alpha_2(x_k)]^2},\sqrt{[\alpha_i(x_k)]^2+[\alpha_j(x_k)]^2e^{-2\mu_j(x_k)t}}\right]\subset J_2$ when $\mu_j(x_k)<0$. However, in both cases it is $g\left(\sqrt{[\alpha_1(x_k)]^2+[\alpha_2(x_k)]^2}\right)=g\left(\|p_k'(0)\|_2\right)=0$ and $g\left(\sqrt{[\alpha_i(x_k)]^2+[\alpha_j(x_k)]^2e^{-2\mu_j(x_k)t}}\right)=g\left(\|p_k'(t)\|_2\right)=t$ for all $t\in[0,+\infty[$. Therefore, we may apply the theorem on the integration by substitution to the integral in (67) and obtain

$$\begin{aligned}
\varsigma_k(t)=&\int_{\|p_k'(0)\|_2}^{\|p_k'(t)\|_2}\sqrt{[\alpha_i(x_k)]^2+[\alpha_j(x_k)]^2e^{-2\mu_j(x_k)g(\chi)}}\,g'(\chi)\,d\chi\\
=&\int_{\|p_k'(0)\|_2}^{\|p_k'(t)\|_2}\chi\left\{-\frac{\chi}{\mu_j(x_k)\left\{\chi^2-[\alpha_i(x_k)]^2\right\}}\right\}d\chi\\
=&\frac{1}{\mu_j(x_k)}\int_{\|p_k'(t)\|_2}^{\|p_k'(0)\|_2}\frac{\chi^2}{\chi^2-[\alpha_i(x_k)]^2}\,d\chi\\
=&\frac{1}{\mu_j(x_k)}\int_{\|p_k'(t)\|_2}^{\|p_k'(0)\|_2}\frac{\chi^2-[\alpha_i(x_k)]^2+[\alpha_i(x_k)]^2}{\chi^2-[\alpha_i(x_k)]^2}\,d\chi\\
=&\frac{1}{\mu_j(x_k)}\left[\int_{\|p_k'(t)\|_2}^{\|p_k'(0)\|_2}d\chi-[\alpha_i(x_k)]^2\int_{\|p_k'(t)\|_2}^{\|p_k'(0)\|_2}\frac{1}{[\alpha_i(x_k)]^2-\chi^2}\,d\chi\right]
\end{aligned}$$

$$=\frac{1}{\mu_j(x_k)}\left[\chi-\frac{\alpha_i(x_k)}{2}\log\left|\frac{\alpha_i(x_k)+\chi}{\alpha_i(x_k)-\chi}\right|\right]_{\|p'_k(t)\|_2}^{\|p'_k(0)\|_2}$$
$$=\frac{1}{\mu_j(x_k)}\left[\varrho_k-\|p'_k(t)\|_2+\frac{\alpha_i(x_k)}{2}\log\left|\frac{\alpha_i(x_k)+\|p'_k(t)\|_2}{\alpha_i(x_k)-\|p'_k(t)\|_2}\right|\right].$$

Now assume that $\alpha_1(x_k)$, $\alpha_2(x_k)$, $\mu_1(x_k)$ and $\mu_2(x_k)$ are not null and $\mu_1(x_k)\neq\mu_2(x_k)$. If $\mu_1(x_k)$ and $\mu_2(x_k)$ are both positive, then we define

$$0\;<\;\xi_k(t)=e^{-2[\mu_1(x_k)-\mu_2(x_k)]t}\quad\leq\;1\qquad\text{with } t\in[0,+\infty[\,,$$
$$\mu_k=\frac{\mu_2(x_k)}{2\,[\mu_1(x_k)-\mu_2(x_k)]}\;>\;0,$$
$$\beta_k=\frac{[\alpha_1(x_k)]^2}{[\alpha_2(x_k)]^2}\;>\;0.$$

We can again apply the substitution method of integration and from (60) obtain

$$\varsigma_k(t)=\frac{|\alpha_2(x_k)|}{2\,[\mu_1(x_k)-\mu_2(x_k)]}\int_{\xi_k(t)}^{1}\chi^{\mu_k-1}\sqrt{1+\beta_k\,\chi}\,d\chi\,.$$

We recall Euler's integral representation of the Gauss hypergeometric function ${}_2F_1$. If $0<\Re(b)<\Re(c)$ and $|\arg(1-z)|<\pi$, then

$${}_2F_1(a,b;c;z)=\frac{\Gamma(c)}{\Gamma(b)\,\Gamma(c-b)}\int_0^1\chi^{b-1}(1-\chi)^{c-b-1}(1-z\chi)^{-a}\,d\chi\,,$$

where ${}_2F_1(a,b;c;z)$ denotes the Gauss hypergeometric function with parameters a, b and c. Here it is understood that $\arg\chi=\arg(1-\chi)=0$ and $(1-z\chi)^{-a}$ has its principal value. For a proof, see Andrews, Askey and Roy [5]. If $\Re(\mu)>0$ and $|\arg(1+\beta\,\xi)|<\pi$, then

$$\int_0^{\xi}\chi^{\mu-1}\left(1+\beta\,\chi\right)^{-\nu}\,d\chi=\frac{\xi^{\mu}}{\mu}\,{}_2F_1\left(\nu,\mu;\mu+1;-\beta\,\xi\right),\tag{68}$$

as reported by Gradshteyn and Ryzhik [32]. Since $0<\xi_k(t)\leq 1$, by the finite additivity property for definite integrals,

$$\varsigma_k(t)=\tfrac{|\alpha_2(x_k)|}{2[\mu_1(x_k)-\mu_2(x_k)]}\left[\int_0^1\frac{\chi^{\mu_k-1}}{\left(1+\beta_k\,\chi\right)^{-\frac{1}{2}}}\,d\chi-\int_0^{\xi_k(t)}\frac{\chi^{\mu_k-1}}{\left(1+\beta_k\,\chi\right)^{-\frac{1}{2}}}\,d\chi\right].\tag{69}$$

Since $\mu_k > 0$, $\beta_k > 0$ and $0 < \xi_k(t) \leq 1$, from (68) and (69) we conclude

$$\varsigma_k(t) = \tfrac{|\alpha_2(x_k)|}{\mu_2(x_k)} \left[{}_2F_1\left(-\tfrac{1}{2}, \mu_k; \mu_k+1; -\beta_k\right) - e^{-\mu_2(x_k)t}{}_2F_1\left(-\tfrac{1}{2}, \mu_k; \mu_k+1; -\beta_k\, \xi_k(t)\right)\right].$$

Analogously, if $\mu_2(x_k) < 0$, then instead of (68) we use

$$\int_{\xi}^{+\infty} \chi^{\mu-1}\,(1+\beta\,\chi)^{-\nu}\; d\chi = \frac{\xi^{\mu-\nu}}{\beta^{\nu}(\nu-\mu)}\;{}_2F_1\left(\nu, \nu-\mu; \nu-\mu+1; -\frac{1}{\beta\,\xi}\right),$$

where $\Re(\nu) > \Re(\mu)$. ■

From equation (59) we get $\varsigma_k'(t) = \|y_k'(t)\|_2$ which, together with (54) and (58), implies $\varsigma_k'(t) > 0\,\forall t \in [0, +\infty[$. Hence function $\varsigma_k(t)$ is strictly increasing in the interval $[0, +\infty[$. The range of $\varsigma_k(t)$ is obviously an interval, say I_k: if Γ_k is bounded, then $I_k = [0, L_k]$, where $L_k = \lim_{t\to+\infty} \varsigma_k(t)$ is the finite length of Γ_k, otherwise $I_k = [0, +\infty[$. All this implies that function $\varsigma_k(t)$ is invertible, its inverse is defined in interval I_k, has range $[0, +\infty[$, is strictly increasing and has a continuous first derivative.

Therefore, in the parametric equation $y_k(t) = x_k + p_k(t)$ of Γ_k we can put t equal to the inverse of $\varsigma_k(t)$, so we obtain a new regular parametric representation of Γ_k with base interval I_k, namely,

$$y_k(\varsigma) = x_k + p_k(t(\varsigma)), \tag{70}$$

which is the *parametric representation of* Γ_k *as a function of the arc.*

Finally, since $p_k(t)$ is twice-continuously differentiable in $[0, +\infty[$, we can define, for all t in $[0, +\infty[$, the *algebraic curvature* $\kappa_k(t)$ of curve Γ_k at $x_k + p_k(t)$, relative to the parametric representation (51).

Theorem 2.12 *Let* Γ_k *denote a curve in* $\mathbf{R}^n$ *and equation (51) be a regular parametric representation of* Γ_k*, with base interval* $[0, +\infty[$*. If the positive orientation fixed on* Γ_k *coincides with the positive orientation induced on* Γ_k *by the parametric representation (51), then the algebraic curvature of* Γ_k*, relative to this parametric* *representation, is the continuous real-valued function* $\kappa_k : [0, +\infty[\subseteq \mathbf{R} \to \mathbf{R}$ *defined as follows:*

$$\kappa_k(t) = \frac{\alpha_1(x_k)\,\alpha_2(x_k)\,[\mu_1(x_k) - \mu_2(x_k)]\,e^{-[\mu_1(x_k)+\mu_2(x_k)]t}}{\left\{[\alpha_1(x_k)]^2 e^{-2\mu_1(x_k)t} + [\alpha_2(x_k)]^2 e^{-2\mu_2(x_k)t}\right\}^{\frac{3}{2}}} \qquad t \in [0, +\infty[\,,$$

and $\kappa_k(t)$ *is nonnegative for all* t *in the base interval.*

Proof. Observe that

$$\begin{aligned}
&\big(\varphi_k'(t)\big)_1\big(\varphi_k''(t)\big)_2 - \big(\varphi_k''(t)\big)_1\big(\varphi_k'(t)\big)_2 \\
&\quad = \left[-\sum_{i=1}^{2} e^{-\mu_i(x_k)t}\big(u_i(x_k)\big)_1\alpha_i(x_k)\right]\left[\sum_{i=1}^{2} \mu_i(x_k)e^{-\mu_i(x_k)t}\big(u_i(x_k)\big)_2\alpha_i(x_k)\right] \\
&\quad\quad - \left[\sum_{i=1}^{2} \mu_i(x_k)e^{-\mu_i(x_k)t}\big(u_i(x_k)\big)_1\alpha_i(x_k)\right]\left[-\sum_{i=1}^{2} e^{-\mu_i(x_k)t}\big(u_i(x_k)\big)_2\alpha_i(x_k)\right] \\
&\quad = \alpha_1(x_k)\,\alpha_2(x_k)[\mu_1(x_k) - \mu_2(x_k)]e^{-[\mu_1(x_k)+\mu_2(x_k)]t}\big(u_1(x_k)\big)_1\big(u_2(x_k)\big)_2 \\
&\quad\quad - \alpha_1(x_k)\,\alpha_2(x_k)[\mu_1(x_k) - \mu_2(x_k)]e^{-[\mu_1(x_k)+\mu_2(x_k)]t}\big(u_1(x_k)\big)_2\big(u_2(x_k)\big)_1 \\
&\quad = \alpha_1(x_k)\,\alpha_2(x_k)[\mu_1(x_k) - \mu_2(x_k)]e^{-[\mu_1(x_k)+\mu_2(x_k)]t} \\
&\quad\quad \left[\big(u_1(x_k)\big)_1\big(u_2(x_k)\big)_2 - \big(u_1(x_k)\big)_2\big(u_2(x_k)\big)_1\right] \\
&\quad = \alpha_1(x_k)\,\alpha_2(x_k)[\mu_1(x_k) - \mu_2(x_k)]e^{-[\mu_1(x_k)+\mu_2(x_k)]t},
\end{aligned}$$

where $\big(u_1(x_k)\big)_1\big(u_2(x_k)\big)_2 - \big(u_1(x_k)\big)_2\big(u_2(x_k)\big)_1 = 1$ because $u_1(x_k)$ and $u_2(x_k)$ are orthonormal. Now the desired result follows from the above equation and the fact that the algebraic curvature of Γ_k is equal to

$$\kappa_k(t) = \frac{1}{\sqrt[3]{\|y_k(t)\|_2}}\left[\big(\varphi_k'(t)\big)_1\big(\varphi_k''(t)\big)_2 - \big(\varphi_k''(t)\big)_1\big(\varphi_k'(t)\big)_2\right].$$

■

Chapter 3

THE ALGORITHM

This chapter describes an algorithm that uses the search curves defined in section 2.2. Starting at a point $x_0 \in D$, the algorithm generates a sequence $(x_k)_{k=0}^{+\infty}$ of iterates in $\mathbf{R}^n$ by determining, at each iterate x_k, a search direction p_k as

$$p_k = p_k(t_k), \tag{1}$$

for some $t_k \in]0, +\infty[$, where $p_k(t)$ is defined by

$$p_k(t) = \begin{cases} (s_k, d_k)\ \vartheta_k(t) & \text{if } x_k \text{ is not a stationary point} \\ d_k\, t & \text{if } x_k \text{ is stationary and not positive semidefinite,} \end{cases} \tag{2}$$

and then by using the relationship

$$x_{k+1} = \begin{cases} x_k + p_k & \text{if } x_k \text{ is not stationary or is not positive semidefinite} \\ x_k & \text{otherwise} \end{cases} \tag{3}$$

for all $k \in \{0, 1, \ldots\}$.

3.1 Convergence theory of the algorithm

It is easy to show that the algorithm we propose is a *well defined* descent method, namely, that there exists a sequence $(t_k)_{k=0}^{+\infty}$ of positive real numbers such that the following property holds:

$$F(x_{k+1}) < F(x_k) \qquad \forall k \in \{0, 1, \ldots\}, \tag{4}$$

unless $x_k = x^*$, where $\nabla F(x^*) = 0$ and $\nabla^2 F(x^*)$ positive semidefinite. Towards this end, we first need to show the following lemma.

Lemma 3.1 *Let $f : I \subseteq \mathbf{R} \to \mathbf{R}$ be a real-valued function continuously-differentiable in the closed interval I. If $\bar{t} \in I$ and $f'(\bar{t}) \neq 0$, then for any real number s such that $sf'(\bar{t}) < 0$ there is a positive constant τ for which I includes the open interval whose extremes are $\bar{t}$ and $\bar{t} + s\tau$, and*

$$f(\bar{t} + st) < f(\bar{t}) \qquad \forall t \in]0, \tau[.$$

Proof. Since f' is continuous in I, $\bar{t} \in I$ and $f'(\bar{t}) \neq 0$, by the theorem on the permanence of sign, there exists a neighborhood $N_\delta(\bar{t}) \subset I$ in which f' is non null and has the same sign of $f'(\bar{t})$. Therefore, for any real number s such that $sf'(\bar{t}) < 0$, we have $sf'(t) < 0$ for all $t \in N_\delta(\bar{t})$. Take $\tau = \frac{\delta}{|s|} > 0$ and, for all $t \in]0, \tau[$, let us apply Lagrange's Theorem to the function f in the intervals $[\bar{t} + st, \bar{t}]$ when $s < 0$, and in the intervals $[\bar{t}, \bar{t} + st]$ when $s > 0$. Let us consider, for instance, the former case. There exists at least a point $\xi \in]\bar{t} + st, \bar{t}[\subset N_\delta(\bar{t})$ such that

$$\begin{aligned} & f(\bar{t}) - f(\bar{t} + st) = f'(\xi)(\bar{t} - \bar{t} - st) \\ \Leftrightarrow \quad & f(\bar{t} + st) - f(\bar{t}) = sf'(\xi)t < 0 \\ \Leftrightarrow \quad & f(\bar{t} + st) < f(\bar{t}) \qquad \forall t \in]0, \tau[. \end{aligned}$$

An analogous argument can be used in the case $s > 0$. ■

Using Lemma 3.1, we now prove the following theorem.

Theorem 3.2 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on the open subset D of $\mathbf{R}^n$. Moreover, let $p_k(t)$ be defined by (2) for all $k \in \{0, 1, \ldots\}$. If $x_0 \in D$, then there exists a sequence $(t_k)_{k=0}^{+\infty}$ of positive real* *numbers that defines $(p_k)_{k=0}^{+\infty}$ according to (1), generates a sequence $(x_k)_{k=0}^{+\infty}$ in D according to (3), and is such that property (4) holds.*

Proof. Without loss of generality, we assume that $x_k \neq x^*$ for all $k \in \{0, 1, \ldots\}$. We shall prove the theorem by induction on the iteration number. At x_0, let $J \subseteq [0, +\infty[$ be a closed interval such that $x_0 + p_0(t) \in D$ for all $t \in J$. Since $x_0 \in D$, D is an open set, $p_0(0) = 0$ and p_0 is continuous in $[0, +\infty[$, we have that $J \neq \varnothing$ and $\{0\}$ is a proper subset of J. Consider the real-valued function $f : J \subseteq \mathbf{R} \to \mathbf{R}$ defined by

$$f(t) = F(x_0 + p_0(t)) \qquad t \in J.$$

Since $p_0 \in C^\infty([0, +\infty[)$ and F is twice-continuously differentiable in D, by the theorem on the differentiability of composition, we have that f is twice-continuously differentiable in J and

$$\begin{aligned} f'(t) &= \nabla F(x_0 + p_0(t))^T p_0'(t) \\ f''(t) &= p_0'(t)^T \nabla^2 F(x_0 + p_0(t)) p_0'(t) + \nabla F(x_0 + p_0(t))^T p_0''(t) \end{aligned} \qquad \forall t \in J. \tag{5}$$

Let $\bar{t} = 0$, so that

$$f(\bar{t}) = f(0) = F(x_0 + p_0(0)) = F(x_0).$$

If x_0 is a stationary point, then

$$\begin{aligned} f'(\bar{t}) =& f'(0) = \nabla F(x_0)^T p_0'(0) = 0, \\ f''(\bar{t}) =& f''(0) = p_0'(t)^T \nabla^2 F(x_0 + p_0(t)) p_0'(t) \\ =& \frac{\sum\limits_{i \in \mathcal{W}_k} v_i(x_k)^T}{\sqrt{\operatorname{card}(\mathcal{W}_k)}} \nabla^2 F(x_k) \frac{\sum\limits_{i \in \mathcal{W}_k} v_i(x_k)}{\sqrt{\operatorname{card}(\mathcal{W}_k)}} \\ =& \lambda_n(x_k) < 0. \end{aligned}$$

Since f'' is continuous in $[0, +\infty[$ and $f''(0) < 0$, by the theorem on the permanence of sign, there exists a positive real number δ such that $[0, +\infty[\subset J$ and $f''(t) < 0$ for all $t \in [0, \delta[$. From Taylor's formula of order two of function f, relative to point

$\bar{t} = 0$, with the remainder in Lagrange form, we have

$$\begin{aligned} f(t) &= f(0) + f'(0)\,\delta + \frac{1}{2} f''(\xi)\,\delta^2 \\ &= f(0) + \frac{1}{2} f''(\xi)\,\delta^2, \end{aligned}$$

where $t \in]0, \delta[$ and $\xi \in]0, t[$, so that $f''(\xi) < 0$. Therefore, $f(t) - f(0) < 0$ for all $t \in]0, \delta[$. We now take $t_0 \in]0, \delta[$ and $p_0 = p_0(t_0)$, so that $x_1 = x_0 + p_0$ *is an element of* D and $F(x_1) < F(x_0)$.

If x_0 is not a stationary point, then from (5) we have

$$\begin{aligned} f'(\bar{t}) =& f'(0) = \nabla F(x_0)^T p_0'(0) \\ =& \nabla F(x_0)^T \left[-\nabla F(x_0)\right] \\ =& - \left\|\nabla F(x_0)\right\|_2^2 < 0. \end{aligned}$$

If we let $s = 1$, then $sf'(\bar{t}) < 0$ and, by Lemma 3.1, there exists a positive constant τ for which $[0, \tau[\subseteq J$ and

$$\begin{aligned} & f(\bar{t} + t) < f(\bar{t}) && \forall t \in]0, \tau[\\ \Leftrightarrow \quad & f(t) < f(0) && \forall t \in]0, \tau[\\ \Leftrightarrow \quad & F(x_0 + p_0(t)) < F(x_0) && \forall t \in]0, \tau[. \end{aligned}$$

We now take $t_0 \in]0, \tau[$ and $p_0 = p_0(t_0)$, so that $x_1 = x_0 + p_0$ *is an element of* D and

$$F(x_1) < F(x_0).$$

Assume now that there exists a positive real number t_{k-1} which defines p_{k-1} and x_k as $p_{k-1} = p_{k-1}(t_{k-1})$ and $x_k = x_{k-1} + p_{k-1}$, so that $x_k \in D$ (and $F(x_k) < F(x_{k-1})$). We can repeat the same argument given for the base case and show that there exists a positive real number t_k which defines p_k and x_{k+1} according to (1) and (3), so that $x_{k+1} \in D$ and $F(x_{k+1}) < F(x_k)$. ■

It is well known that condition (4) is not sufficient in itself to ensure that the sequence $(x_k)_{k=0}^{+\infty}$ converges to a minimizer of F. If condition (4) holds and F is bounded below, then clearly the sequence $(F(x_k))_{k=0}^{+\infty}$ converges. However, we must ensure that the sequence converges to $F(x^*)$. What we basically require is that the reduction $F(x_{k+1}) - F(x_k)$ tends to zero *only if* x_k tends to x^*.

To define the sequence $(x_k)_{k=0}^{+\infty}$ by (3), we use the parametric representations of the search curves as functions of the arc, as we have defined them in section 2.3, namely,

$$y_k(\varsigma) = x_k + p_k(t(\varsigma)) \qquad \varsigma \in I_k, \tag{6}$$

where I_k is either $[0, L_k]$ or $[0, +\infty[$. If $\mu \in]0,1[$ and $\eta \in [\mu, 1[$, then we define $\Xi_k(\mu, \eta)$ as the set of points $\varsigma \in I_k$ such that $x_k + p_k(t(\varsigma)) \in D$ and

$$F\big(x_k + p_k\,(t(\varsigma))\big) \le F(x_k) + \mu\left[-\varsigma\,\|\nabla F(x_k)\|_2 + \frac{1}{2}\,\varsigma^2 \min\{\lambda_n(x_k), 0\}\right], \tag{7}$$

$$\left|\nabla F\big(x_k + p_k\,(t(\varsigma))\big)^T \frac{p_k'\,(t(\varsigma))}{\|p_k'\,(t(\varsigma))\|_2}\right| \le -\eta\Big[-\|\nabla F(x_k)\|_2 + \varsigma \min\{\lambda_n(x_k), 0\}\Big]. \tag{8}$$

The following lemma will show that conditions (7) and (8) can be satisfied whenever a descent pair of F exists at a point $x_k \in D$.

Lemma 3.3 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$, and (s_k, d_k) be the nonascent pair of F defined by (2.26) or (2.27) at a point $x_k \in D$ that is not a stationary point of F or is not a positive semidefinite point of F. Moreover, let $x_k + p_k(t(\varsigma))$ be the search curve from x_k defined by (6), where $p_k(t)$ is defined by (2) and $t(\varsigma)$ is the inverse function of the arc length parameter of $x_k + p_k(t)$ with base point x_k. Assume that x_k is such that the level set $\Omega(F(x_k))$ is compact. If $0 < \mu \le \eta < 1$, then the set $\Xi_k(\mu, \eta)$ defined by (7) and (8) contains a nontrivial interval.*

Proof. As in the proof of Theorem 3.2, let $J_k \subseteq I_k$ denote a closed interval, containing zero, such that $x_k + p_k(t(\varsigma)) \in D$ for all $\varsigma \in J_k$. Moreover, consider the real-valued function $f_k : J_k \subseteq \mathbf{R} \to \mathbf{R}$ defined by

$$f_k(\varsigma) = F\big(x_k + p_k\,(t(\varsigma))\big) \qquad \varsigma \in J_k.$$

Recall that $\varsigma_k(t)$ is differentiable in $[0, +\infty[$, $\varsigma_k'(t) = \|p_k'(t)\|_2 \neq 0$ for all $t \in [0, +\infty[$, and the inverse $t(\varsigma)$ is continuous in I_k. Therefore, $t(\varsigma)$ is differentiable in I_k and

$$t'(\varsigma) = \frac{1}{\varsigma_k'(t(\varsigma))} = \frac{1}{\|p_k'(t(\varsigma))\|_2}.$$

Hence

$$f_k'(\varsigma) = \nabla F\big(x_k + p_k\,(t(\varsigma))\big)^T \frac{p_k'(t(\varsigma))}{\|p_k'(t(\varsigma))\|_2} \tag{9}$$

and

$$f_k'(0) = \nabla F(x_k)^T \frac{s_k}{\|s_k\|_2} = -\|\nabla F(x_k)\|_2. \tag{10}$$

Define β as

$$\beta = \sup_{\varsigma \in J_k} \left\{\varsigma : \varsigma > 0,\ f(\varsigma) \leq f(0)\right\}.$$

From the definition of $p_k(t)$, we have either $f_k'(0) < 0$ or $f_k'(0) = 0$ and $f_k''(0) < 0$. This, together with the continuity of f_k and the compactness property of $\Omega(f_k(0))$, implies that $\beta > 0$, it is finite and $f_k(0) = f_k(\beta)$. Thus, for all $\varsigma \geq 0$,

$$f_k(\beta) \geq f_k(0) + \mu\left[\varsigma\, f_k'(0) + \frac{1}{2}\,\varsigma^2 \min\left\{\lambda_n(x_k), 0\right\}\right], \tag{11}$$

because $\mu > 0$, $f_k'(0) \leq 0$ and $\min\{\lambda_n(x_k), 0\} \leq 0$. Consider now the real-valued function $h_k : J_k \subseteq \mathbf{R} \to \mathbf{R}$ defined by

$$h_k(\varsigma) = f_k(\varsigma) - f_k(0) - \mu\left[\varsigma\, f_k'(0) + \frac{1}{2}\,\varsigma^2 \min\left\{\lambda_n(x_k), 0\right\}\right]. \tag{12}$$

Condition (11) implies $h_k(\beta) \geq 0$. Note also that $h_k(0) = 0$ and either $h_k'(0) < 0$ or $h_k'(0) = 0$ and $h_k''(0) < 0$. This, together with the continuity of h_k, implies the existence of a point $\hat{\beta} \in]0, \beta]$ such that $h_k(\hat{\beta}) = 0$ and $h_k(\varsigma) < 0$ for all $\varsigma \in]0, \hat{\beta}[$. By Rolle's theorem, there exists a point $\xi \in]0, \hat{\beta}[$ such that $h_k'(\xi) = 0$. Hence, from (12),

$$f_k'(\xi) = \mu\left[f_k'(0) + \xi \min\left\{\lambda_n(x_k), 0\right\}\right],$$

which shows that $f_k'(\xi) < 0$ and implies

$$|f_k'(\xi)| = -\mu\left[f_k'(0) + \xi \min\left\{\lambda_n(x_k), 0\right\}\right].$$

Since $\mu \leq \eta$, the above equality implies

$$|f_k'(\xi)| \leq -\eta \left[f_k'(0) + \xi \min \{\lambda_n(x_k), 0\} \right].$$

Thus ξ satisfies (8). Moreover, conditions $\xi < \hat{\beta}$ and $h_k(\varsigma) < 0$ for all $\varsigma \in]0, \hat{\beta}[$ imply that $h_k(\xi) < 0$ and, therefore,

$$f_k(\xi) \leq f_k(0) + \mu \left[\xi\, f_k'(0) + \frac{1}{2}\, \xi^2 \min \{\lambda_n(x_k), 0\} \right].$$

Thus ξ also satisfies (7). Finally, continuity of f_k' shows that $\Xi_k(\mu, \eta)$ must contain an interval of points. ∎

To prove convergence of the proposed algorithm, we will also need the following lemma; for a proof, see Ortega and Rheinboldt [46].

Lemma 3.4 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function continuously differentiable on an open subset D of $\mathbf{R}^n$, and assume that point $x_0 \in D$ is such that the level set $\Omega(F(x_0))$ is compact. In addition, assume that F has a finite number of stationary points in $\Omega(F(x_0))$. If the sequence $(x_k)_{k=0}^{+\infty}$ is in $\Omega(F(x_0))$,*

$$\lim_{k \to +\infty} \|x_{k+1} - x_k\|_2 = 0 \qquad \textit{and} \qquad \lim_{k \to +\infty} \|\nabla F(x_k)\|_2 = 0,$$

then $(x_k)_{k=0}^{+\infty}$ converges to some point $x^* \in \Omega(F(x_0))$ with $\nabla F(x^*) = 0$.

Using Lemma 3.3 and Lemma 3.4, we now prove the following theorem.

Theorem 3.5 *Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$, and assume that point $x_0 \in D$ is such that the level set $\Omega(F(x_0))$ is compact. In addition, assume that F has a finite number of stationary points in $\Omega(F(x_0))$. Let the sequences $(x_k)_{k=0}^{+\infty}$, $\big((s_k, d_k)\big)_{k=0}^{+\infty}$, $(\varsigma_k)_{k=0}^{+\infty}$ and $(p_k)_{k=0}^{+\infty}$ be such that*

$$x_{k+1} = \begin{cases} x_k + p_k(t(\varsigma_k)) & \textit{if } x_k \textit{ is not stationary or is not positive semidefinite} \\ x_k & \textit{otherwise} \end{cases}, \tag{13}$$

(s_k, d_k) is the nonascent pair of F defined by (2.26) or (2.27) at a point $x_k \in D$ that is not a stationary point of F or is not a positive semidefinite point of F, $t(\varsigma)$ is the inverse function of the arc length parameter of $x_k + p_k(t)$ with base point x_k, $p_k(t)$ is defined by (2) and $\varsigma_k > 0$ is an element of the set $\Xi_k(\mu, \eta)$ defined by conditions (7) and (8) with $0 < \mu \leq \eta < 1$. Then the following results hold:

1. *the sequence $(x_k)_{k=0}^{+\infty}$ is well defined and lies in $\Omega(F(x_0))$;*

2. *either there exists a finite index l such that $\nabla F(x_l) = 0$ and $\nabla^2 F(x_l)$ is positive semidefinite, or the sequence $(x_k)_{k=0}^{+\infty}$ converges to a point $x^* \in D$ such that $\nabla F(x^*) = 0$ and $\nabla^2 F(x^*)$ is positive semidefinite.*

Proof. To prove *1*, observe that, if $x_k \in \Omega(F(x_0))$ then the level set $\Omega(F(x_k))$ is also compact and, if x_k is not a stationary point of F or is not a positive semidefinite point of F, Lemma 3.3 ensures the existence of a suitable positive $\varsigma_k \in \Xi_k(\mu, \eta)$ such that $x_k + p_k(t(\varsigma_k)) \in \Omega(F(x_k)) \subseteq \Omega(F(x_0))$. Since $x_0 \in \Omega(F(x_0))$, by induction on the iteration number we conclude that x_k is well defined and is an element of $\Omega(F(x_0))$ for all $k \in \{0, 1, \ldots\}$. In addition, the sequence $\big(F(x_k)\big)_{k=0}^{+\infty}$ is strictly decreasing and hence the compactness of $\Omega(F(x_0))$ implies that it converges.

To prove *2*, observe that the theorem holds trivially if $\nabla F(x_l) = 0$ and $\nabla^2 F(x_l)$ is positive semidefinite for any finite index l. Moreover, since F has a finite number of stationary points in $\Omega(F(x_0))$, without loss of generality we can consider only the case when $\nabla F(x_k) \neq 0$ for all $k \in \{0, 1, \ldots\}$. As in the proof of Lemma 3.3, let $J_k \subseteq I_k$ denote a closed interval, containing zero, such that $x_k + p_k(t(\varsigma)) \in D$ for all $\varsigma \in J_k$. Consider also the real-valued functions $f_k : J_k \subseteq \mathbf{R} \to \mathbf{R}$ defined, for all $k \in \{0, 1, \ldots\}$, by

$$f_k(\varsigma) = F\big(x_k + p_k\,(t(\varsigma))\big) \qquad \varsigma \in J_k. \tag{14}$$

Since $\varsigma_k \in \Xi_k(\mu, \eta)$, from (7), (10) and (14) we have

$$f_k(\varsigma_k) - f_k(0) \leq \mu \left[\varsigma_k\, f_k'(0) + \frac{1}{2}\,\varsigma_k^2 \min\left\{\lambda_n(x_k), 0\right\}\right] \leq 0. \tag{15}$$

Convergence of the sequence $\big(F(x_k)\big)_{k=0}^{+\infty}$ implies that $\big(f_k(\varsigma_k) - f_k(0)\big)_{k=0}^{+\infty}$ converges to zero and hence, from (15) and the fact that $\mu > 0$, we have

$$\lim_{k\to+\infty} \varsigma_k\, f_k'(0) = 0, \tag{16}$$

$$\lim_{k\to+\infty} \varsigma_k^2 \min\{\lambda_n(x_k), 0\} = 0. \tag{17}$$

Since $\varsigma_k \in \Xi_k(\mu, \eta)$, from (8), (9) and (10) we have

$$f_k'(\varsigma_k) \geq \eta \Big[f_k'(0) + \varsigma_k \min\{\lambda_n(x_k), 0\} \Big]. \tag{18}$$

In (16) assume that, as k goes to $+\infty$, ς_k tends to zero but $f_k'(0)$ *does not*. Hence there is a real number $\epsilon > 0$ and an infinite subsequence of $\big(f_k'(0)\big)_{k=0}^{+\infty}$, with indexes $(i_k)_{k=0}^{+\infty}$, such that

$$-f_{i_k}'(0) \geq \epsilon \qquad \forall k \in \{0, 1, \ldots\}. \tag{19}$$

Thus $f_{i_k}'(0) + \varsigma_{i_k} \min\{\lambda_n(x_{i_k}), 0\} < 0$ for all $k \in \{0, 1, \ldots\}$, and from (18) we obtain

$$\frac{f_{i_k}'(\varsigma_{i_k})}{f_{i_k}'(0) + \varsigma_{i_k} \min\{\lambda_n(x_{i_k}), 0\}} \leq \eta \qquad \forall k \in \{0, 1, \ldots\}. \tag{20}$$

Since $\nabla^2 F(x_k)$ is continuous in the compact set $\Omega(F(x_0))$, by (17) we have

$$\lim_{k\to+\infty} \varsigma_{i_k} \min\{\lambda_n(x_{i_k}), 0\} = 0. \tag{21}$$

Moreover, since ς_k tends to zero, continuity of f_k' and (19) imply

$$\lim_{k\to+\infty} f_{i_k}'(\varsigma_{i_k}) = \lim_{k\to+\infty} f_{i_k}'(0) \neq 0. \tag{22}$$

Hence, if we take the limit of both sides of inequality (20), equations (21) and (22) imply

$$1 \leq \eta,$$

which contradicts the assumption $\eta < 1$. Therefore, it must be the case that, as k goes to $+\infty$, $f_k(0) = -\|\nabla F(x_k)\|_2$ tends to zero, so that

$$\lim_{k\to+\infty} \nabla F(x_k) = 0. \tag{23}$$

In (17) assume that, as k goes to $+\infty$, ς_k tends to zero but $\min\{\lambda_n(x_k), 0\}$ *does not*. Hence there is a real number $\epsilon > 0$ and an infinite subsequence of $(\min\{\lambda_n(x_k), 0\})_{k=0}^{+\infty}$, with indexes $(j_k)_{k=0}^{+\infty}$, such that

$$-\lambda_n(x_{j_k}) \geq \epsilon \qquad \forall k \in \{0, 1, \ldots\}. \tag{24}$$

This, together with the inequalities $f'_{j_k}(0) \leq 0$ and $\varsigma_k > 0$ for all $k \in \{0, 1, \ldots\}$, implies that $f'_{j_k}(0) + \varsigma_{j_k} \min\{\lambda_n(x_{j_k}), 0\} < 0$ for all $k \in \{0, 1, \ldots\}$. Thus from (18) we obtain (20)

$$\frac{f'_{j_k}(\varsigma_{j_k})}{f'_{j_k}(0) + \varsigma_{j_k}\,\lambda_n(x_{j_k})} \leq \eta \qquad \forall k \in \{0, 1, \ldots\}. \tag{25}$$

Since both numerator and denominator on the left-hand side of inequality (25) tend to zero, and their first derivatives are convergent, we can take the limit of both sides and apply l'Hôspital's theorem, which yields

$$\lim_{k \to +\infty} \frac{f''_{j_k}(\varsigma_{j_k})}{\lambda_n(x_{j_k})} \leq \eta. \tag{26}$$

If ς_k tends to zero, then continuity of f''_k implies

$$\lim_{k \to +\infty} f''_{j_k}(\varsigma_{j_k}) = \lim_{k \to +\infty} f''_{j_k}(0). \tag{27}$$

We differentiate (9) to obtain

$$\begin{aligned} f''_k(\varsigma) = & \frac{p'_k(t(\varsigma))^T}{\|p'_k(t(\varsigma))\|_2} \nabla^2 F\big(x_k + p_k\,(t(\varsigma))\big) \frac{p'_k(t(\varsigma))}{\|p'_k(t(\varsigma))\|_2} \\ & + \nabla F\big(x_k + p_k\,(t(\varsigma))\big)^T \left[\frac{p''_k(t(\varsigma))}{\|p'_k(t(\varsigma))\|_2} - \frac{p'_k(t(\varsigma)) p'_k(t(\varsigma))^T p''_k(t(\varsigma))}{\|p'_k(t(\varsigma))\|_2^3} \right]. \end{aligned}$$

Since $p'_k(t(0)) = s_k$, letting $\varsigma = 0$ in the above formula gives

$$\begin{aligned} f''_k(0) &= \frac{s_k^T}{\|s_k\|_2} \nabla^2 F(x_k) \frac{s_k}{\|s_k\|_2} - \frac{s_k^T p''_k(t(0))}{\|s_k\|_2} + \frac{\|s_k\|_2^2\, s_k^T p''_k(t(0))}{\|s_k\|_2^3} \\ &= \frac{s_k^T}{\|s_k\|_2^2} \nabla^2 F(x_k) s_k. \end{aligned} \tag{28}$$

Hence from (26), (27) and (28) we obtain

$$\lim_{k\to+\infty} \frac{s_{j_k}^T \nabla^2 F(x_{j_k}) s_{j_k}}{\|s_{j_k}\|_2^2 \, \lambda_n(x_{j_k})} \le \eta. \tag{29}$$

Since $\lambda_n(x_{j_k})$ is the smallest eigenvalue of $\nabla^2 F(x_{j_k})$, in the notation of Chapter 2 we may write

$$\begin{aligned}
\frac{s_{j_k}^T \nabla^2 F(x_{j_k}) s_{j_k}}{\|s_{j_k}\|_2^2 \, \lambda_n(x_{j_k})} &= \frac{s_{j_k}^T \sum_{i=1}^{n} \lambda_i(x_{j_k}) v_i(x_{j_k}) v_i(x_{j_k})^T s_{j_k}}{\|s_{j_k}\|_2^2 \, \lambda_n(x_{j_k})} \\
&= \frac{\sum_{i=1}^{n} \lambda_i(x_{j_k}) \left[s_{j_k}^T v_i(x_{j_k}) \right]^2}{\|s_{j_k}\|_2^2 \, \lambda_n(x_{j_k})} \\
&\ge \frac{\lambda_n(x_{j_k}) \sum_{i=1}^{n} \left[s_{j_k}^T v_i(x_{j_k}) \right]^2}{\|s_{j_k}\|_2^2 \, \lambda_n(x_{j_k})} \\
&= \frac{\sum_{i=1}^{n} \left[s_{j_k}^T v_i(x_{j_k}) \right]^2}{\|s_{j_k}\|_2^2} = 1.
\end{aligned}$$

If we use the above inequality in (29), then it follows

$$1 \le \lim_{k\to+\infty} \frac{s_{j_k}^T \nabla^2 F(x_{j_k}) s_{j_k}}{\|s_{j_k}\|_2^2 \, \lambda_n(x_{j_k})} \le \eta,$$

which again contradicts the assumption $\eta < 1$. Therefore, it must be the case that, as k goes to $+\infty$, $\min\{\lambda_n(x_k), 0\}$ tends to zero, so that

$$\lim_{k\to+\infty} \nabla^2 F(x_k) \quad \text{is positive semidefinite.}$$

Finally, from the definition of $\vartheta_k(t)$ we get that convergence to zero of the sequences $\left(\nabla F(x_k)\right)_{k=0}^{+\infty}$ implies

$$\lim_{k\to+\infty} \vartheta_k(t(\varsigma_k)) = 0. \tag{30}$$

From (2) and (13) we have

$$\|x_{k+1} - x_k\|_2 = \|p_k(t(\varsigma_k))\|_2 \le \left(\|\nabla F(x_k)\|_2, \|d_k\|_2\right) \vartheta_k(t(\varsigma_k)).$$

Hence (23) and (30) imply

$$\lim_{k\to+\infty} \|x_{k+1} - x_k\|_2 = 0.$$

Therefore, Lemma 3.4 applies and we obtain that the sequence $(x_k)_{k=0}^{+\infty}$ converges to some point $x^* \in \Omega(F(x_0))$ with $\nabla F(x^*) = 0$. Since $\min\{\lambda_n(x_k), 0\}$ tends to zero as k goes to $+\infty$, we also have that $\nabla^2 F(x^*)$ must be positive semidefinite. ■

We want to remark that the original method of gradients converges only to a stationary point. Instead, with the introduction of search criteria (7) and (8) we were able to prove convergence of the subspace method to a stationary point where the Hessian matrix is positive semidefinite.

3.2 Statement of the algorithm

We now combine all the ideas of the previous sections into an algorithm for the solution of unconstrained optimization problems. Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be a real-valued function twice-continuously differentiable on an open subset D of $\mathbf{R}^n$. Let x_0 be a point of D, and let μ and η be two positive real numbers such that $0 < \mu \leq \eta < 1$. Starting at x_0, with $k = 0$, we first evaluate the function value $F(x_k)$, the gradient vector $\nabla F(x_k)$ and the Hessian matrix $\nabla^2 F(x_k)$ at x_k. While the termination criteria are not met, we compute the nonascent pair (s_k, d_k) of F at x_k, then the matrix G_k and its eigensystem $\mu_1(x_k), \mu_2(x_k), u_1(x_k)$ and $u_2(x_k)$. This is the information that identifies the curve Γ_k.

Once we have Γ_k represented by $x_k + p_k(t(\varsigma))$ as a function of the arc, we seek a point along it that satisfies the search criteria described in section 3.1, namely, a point $x_k + p_k(t(\varsigma_k))$ such that $\varsigma_k \in \Xi_k(\mu, \eta)$. Our first task in finding a point along Γ_k that satisfies the search criteria is to select an initial trial point. The scheme for selecting this point is an important factor in the performance of the algorithm, both its speed and its robustness. A good scheme will produce initial points that themselves satisfy the search criteria (7) and (8) a large percentage of the time and that when they do not are not far from points that do.

For a bounded curve, the point at the end of the curve has arc length parameter equal to the length L_k of the curve. If the Hessian matrix $\nabla^2 F(x_k)$ is positive definite, then the step to $x_k + p_k(t(L_k))$ is the Newton step. As we have seen in Chapter 1, this point has long been recognized as a good initial trial point in other unconstrained optimization algorithms, especially close to a solution. Therefore, we use this step as the initial trial point for bounded curves. For unbounded curves, we use the point with arc length parameter equal to $-\mu_2(x_k)^{-1}$ subject to the safeguard that, if $|\mu_2(x_k)|$ is not sufficiently large, we use the point with arc length parameter equal to $1 - \mu_2(x_k)$. The just-mentioned points are usually, but not always, good initial trial points. We need to constrain them somewhat as a safeguard for those cases in which they are not.

We search for a point with an arc length parameter in the search interval $[0, L_k]$ for a bounded curve, or in the search interval $[0, +\infty[$ for an unbounded curve. A trial point divides the search interval into two subintervals; if the curve is bounded and the trial point is at arc length parameter L_k, then one of the subinterval is empty. Based upon information of the function at the trial point, either it is accepted as the next point in the iteration or one of the two subintervals is discarded and the trial point becomes the new upper bound or the new lower bound of the search interval. The process continues until either a trial point is accepted or a maximum number of trials occur, in which case an error condition is returned and the algorithm stops.

At the end of the curve-search, once we have accepted the point with arc length parameter ς_k, we set $p_k = p_k(t(\varsigma_k))$ and move to point $x_{k+1} = x_k + p_k$. Algorithm 1 on the next page expresses formally the complete method that we have specified.

3.3 Summary and future work

In this dissertation we presented a subspace method based on a differential equation approach to solve unconstrained optimization problems. The method was similar in spirit to the original method of gradients. However, unlike the original method of gradients, it required the solution of a system of only two ordinary differential equations.

Algorithm 1 (Subspace Method)

Let $F : D \subseteq \mathbf{R}^n \to \mathbf{R}$ be twice-continuously differentiable on the open set D;
Let $x_0 \in D$;
Let μ, $\eta \in \mathbf{R}$ such that $0 < \mu \leq \eta < 1$;
$k = 0$;
Evaluate $F(x_k)$, $\nabla F(x_k)$ and $\nabla^2 F(x_k)$;
while (*termination criteria are not met*) **do**
  Compute the nonascent pair (s_k, d_k) of F at x_k;
  Compute W_k and N_k;
  $G_k = W_k^T \nabla^2 F(x_k) W_k$;
  Compute $\mu_1(x_k)$, $\mu_2(x_k)$, $u_1(x_k)$ and $u_2(x_k)$;
  if Γ_k is bounded **then**
    $\varsigma_k = L_k$;
  else
    if $\mu_2(x_k)$ is sufficiently large **then**
      $\varsigma_k = -\mu_2(x_k)^{-1}$;
    else
      $\varsigma_k = 1 - \mu_2(x_k)$;
    end if
  end if
  while $\varsigma_k \notin \Xi_k(\mu, \eta)$ **do**
    Use cubic or quadratic safeguarded interpolation to find a trial point ς_k;
    Compute $\varphi_k(t(\varsigma_k))$;
    $p_k = (s_k, d_k)\, N_k \varphi_k(t(\varsigma_k))$;
    Evaluate $F(x_k + p_k)$ and $\nabla F(x_k + p_k)$;
  end while
  $x_{k+1} = x_k + p_k$;
  $k = k + 1$;
  Evaluate $\nabla^2 F(x_k)$;
end while

This system was definable, made sense and was solvable regardless the nature of the Hessian matrix of the objective function. Its solution was unaffected by possible poor scaling of the optimization problem. Under mild assumptions the method was shown to converge to a point satisfying the first and second-order necessary conditions for unconstrained optimality.

A MATLAB implementation of the method was provided. It has been tested and have been proven to be be very reliable and efficient in comparison to leading alternative routines. It is at least as robust as the algorithm that solves the system of n ordinary differential equations. Test problem documentation, software documentation and numerical results have been gathered in the appendices.

The method can be extended to constrained optimization problems almost as simply as linesearch methods. In the case of linear equality constraints, the system of differential equations is applied to the space spanned by the columns of Z which form a basis for the set of vectors orthogonal to the rows of the constraint matrix. This means that the gradient ∇F and the Hessian matrix $\nabla^2 F$ are replaced by the projected gradient $Z^T \nabla F$ and the projected Hessian matrix $Z^T \nabla^2 F Z$, respectively.

In the case of simple bounds in which at each iteration we have only a subset of the variables active, in determining whether to move off a constraint, only first-order multipliers need be computed, since the incremental step is the projected steepest descent. This means multiple constraints can safely be deleted. Theorem 2.12 showed that a search curve either lies on a constraint, or intersects it at most in two points, because its curvature never changes sign in the base interval.

The case of nonlinear constraints depends on the type of algorithm. For example, a MINOS-type algorithm would be relatively straightforward to implement, since we are solving a sequence of linearly constrained problems and that case has been covered. Likewise nonlinear equality constraint problem are relatively easy to deal with since, like the linear case, we can consider the curve lying on the space spanned by a linearization of the constraints. However, the current iterate will not lie on the constraints so there is a component of the search stepping to the constraint.

Therefore, we have a well-defined curve lying in a plane and a step to the plane, and there are several possible ways of combining them. The more difficult case is determining the equivalent of an SQP method.

Finally, since the method of this dissertation is an outgrowth of a method proposed to solve variational problems, it would be instructive to compare the performance of our method with that of methods specifically designed to solve variational problems associated with partial differential equations.

Appendix A

TEST PROBLEM DOCUMENTATION

The algorithm in Chapter 3 was tested with 37 test problems taken from Albert G. Buckley's test set [16]. This is a compilation from many sources of 417 test problems from 83 test functions; some have been developed by researchers for testing algorithms, others are real problems from practitioners. Following the terminology adopted by Behrman [6], here we use the term *test function* to mean a functional form that may have a variable size or have parameters that may be varied, and we use the term *test problem* to mean a test function with specified size and parameters and a specified initial point.

From Buckley's test set Behrman selected for implementation 142 test problems from all 83 test functions. Each of the 142 test problems was implemented in Fortran with the aid of Mathematica and ADIFOR. Mathematica was used to generate the objective functions and the gradient vectors for all the problems and the Hessian matrices for problems of size $n \leq 20$. ADIFOR was used to generate the Hessian-vector products for all problems and the Hessian matrices for problems of size $n > 20$.

Behrman provided two versions of each test problem, of the form prob.f and prob.ad.f, which he called the ANSI and ADIFOR versions, respectively. The ANSI version is in ANSI-standard Fortran 77. The ADIFOR version contains ADIFOR-generated code using the following three extensions to the ANSI standard: use of the underscore in identifier, the DO . . . END DO statement and the symbolic names longer than 6 characters. Since the Fortran compiler we used could compile the ADIFOR version, we used it instead of the ANSI version because the ADIFOR version has additional functionality.

For further information on Behrman's implementation of the test problems, including information on how to use them, see Appendix A in [6].

Appendix B

SOFTWARE DOCUMENTATION

We provide a MATLAB implementation of Algorithm 1 on page 70 for solving unconstrained optimization problems. The routine is called SUMEBADEA, for **su**bspace **me**thod **ba**sed on a **d**ifferential **e**quation **a**pproach.

We also provide MEX-files to call, from MATLAB, the Fortran implementations of the test problems. In this appendix we include as an example the MEX-file called EVA, for **eva**luation. EVA is a gateway subroutine that interfaces MATLAB with the ADIFOR versions of the Fortran computational routines that were provided by Behrman to implement 142 unconstrained minimization test problems from Buckley's test set. The four computational routines called by EVA are

```
SUBROUTINE CF( N, X, F, INFO )
SUBROUTINE CGRAD( N, X, GRAD, INFO )
SUBROUTINE CHESSV( N, X, V, HESSV, INFO )
SUBROUTINE CHESS( N, X, HESS, LDHESS, INFO )
```

which calculate the value of the objective function, the gradient, the Hessian-vector product and the Hessian matrix, respectively.

Finally, we provide a MEX-file to call, from MATLAB, a Fortran implementation of the Gauss hypergeometric function, which is needed to compute the arc length parameter of most search curves. In this appendix we included the MEX-file called

F21, which is a gateway subroutine that interfaces MATLAB with a Fortran computational routine that was provided by Zhang and Jin [63]. The computational routine called by it is

```
SUBROUTINE HYGFX ( A, B, C, X, HF )
```

which computes the Gauss hypergeometric function ${}_2F_1(a, b, c, x)$ with real parameters a, b and c, such that $c \neq 0, -1, -2, \ldots$, and real arguments $x < 1$.

For further information on MEX-files, including information on how to use them, see the MATLAB *Application Program Interface Guide* [38].

```
c     SUBROUTINE EVA
c
c     THIS IS THE GATEWAY ROUTINE OF A FORTRAN MEX-FILE FOR MATLAB.
c
c     Gateway  subroutine  that  interfaces MATLAB  with the ADIFOR
c     versions  of  the  FORTRAN  computational  routines that were
c     provided  by  William  Behrman to implement 142 unconstrained
c     minimization test problems from Albert G. Buckley's test set.
c
c     The four computational routines called by this gateway are
c        SUBROUTINE CF( N, X, F, INFO )
c        SUBROUTINE CGRAD( N, X, GRAD, INFO )
c        SUBROUTINE CHESSV( N, X, V, HESSV, INFO )
c        SUBROUTINE CHESS( N, X, HESS, LDHESS, INFO )
c     which  calculate  the  value  of  the objective function, the
c     gradient, the Hessian-vector product and the Hessian  matrix,
c     respectively.
c
c
c     NOTE: if your FORTRAN compiler does not support the %val
c           construct, use mxCopy__ routines to pass the value
c           of the  variable, rather  than  the address of the
c           variable, to the computational routines.
c
c=====================================================================

c     THE GATEWAY SUBROUTINE

      SUBROUTINE mexFunction(nlhs, plhs, nrhs, prhs)

       IMPLICIT NONE
```

```
c-----------------------------------------------------------------------
c      (INTEGER) Replace INTEGER*8 by INTEGER if not on the DEC Alpha
c      or the SGI 64-bit platforms.

       INTEGER*8 plhs(*), prhs(*)
       INTEGER*8 mxGetPr, mxCreateFull, mxGetString
c-----------------------------------------------------------------------

       INTEGER nlhs, nrhs, mxGetM, mxGetN

c      KEEP THE ABOVE SUBROUTINE, ARGUMENT, AND FUNCTION DECLARATIONS
c      FOR USE IN ALL YOUR FORTRAN MEX-FILES.

c=======================================================================

c-----------------------------------------------------------------------
c      (INTEGER) Replace INTEGER*8 by INTEGER if not on the DEC Alpha
c      or the SGI 64-bit platforms.

       INTEGER*8 pX, pOUT, pV
c-----------------------------------------------------------------------

       INTEGER mxIsString, mxIsNumeric
       INTEGER m_in, n_in, size, strlen, status, sizeV
       INTEGER IpINFO
       CHARACTER*2 Sname

c CHECK FOR PROPER NUMBER OF ARGUMENTS.
       IF (nlhs .NE. 1)
     +    CALL mexErrMsgTxt('One output argument required.')
       IF (nrhs .LT. 2)
```

```
     +    CALL mexErrMsgTxt('At least two input arguments required.')

c CHECK THAT INPUT #1 IS A STRING.
      IF (mxIsString(prhs(1)) .EQ. 0)
     +    CALL mexErrMsgTxt('Input #1 must a string.')

c CHECK THAT INPUT #2 IS NUMERIC.
      IF (mxIsNumeric(prhs(2)) .EQ. 0)
     +    CALL mexErrMsgTxt('Input #2 must be numeric.')

c GET THE DIMENSIONS OF INPUT #1 ARGUMENT.
      m_in   = mxGetM(prhs(1))
      strlen = m_in * mxGetN(prhs(1))

c CHECK THAT INPUT #1 IS A ROW VECTOR.
      IF (m_in .NE. 1)
     +    CALL mexErrMsgTxt('Input #1 must be a row vector.')

c GET THE DIMENSIONS OF INPUT #2 ARGUMENT.
      m_in = mxGetM(prhs(2))
      n_in = mxGetN(prhs(2))
      size = m_in * n_in

c CHECK THAT INPUT #2 IS A VECTOR.
      IF (MIN(m_in,n_in) .NE. 1)
     +    CALL mexErrMsgTxt('Input #2 must be a vector.')

c GET THE STRING CONTENT (DEREFERENCE THE INPUT INTEGER).
      status = mxGetString(prhs(1), Sname, strlen)
```

```
c CHECK THAT THE STRING HAS BEEN CREATED SUCCESSFULLY.
      IF (status .NE. 0)
     +    CALL mexErrMsgTxt('String of input #1 is too long.')

c---------------------------------------------------------------------
c IMPLEMENT A "CASE" STRUCTURE ON THE BASIS OF THE VALUE OF INPUT #1.
c---------------------------------------------------------------------

c CALCULATE THE VALUE OF THE OBJECTIVE FUNCTION AT POINT X.
      IF (Sname .EQ. 'F') THEN
c      CHECK FOR PROPER NUMBER OF INPUT ARGUMENTS.
         IF (nrhs .NE. 2)
     +      CALL mexErrMsgTxt('Two input arguments required.')
c      CREATE A SCALAR FOR THE RETURN ARGUMENT.
         plhs(1) = mxCreateFull(1, 1, 0)
c      ASSIGN POINTERS TO THE VARIOUS PARAMETERS.
         pX   = mxGetPr(prhs(2))
         pOUT = mxGetPr(plhs(1))
c      CALL THE COMPUTATIONAL SUBROUTINE.
         CALL CF(size, %val(pX), %val(pOUT), IpINFO)

c CALCULATE THE GRADIENT OF THE OBJECTIVE FUNCTION AT POINT X.
      ELSEIF (Sname .EQ. 'g') THEN
c      CHECK FOR PROPER NUMBER OF INPUT ARGUMENTS.
         IF (nrhs .NE. 2)
     +      CALL mexErrMsgTxt('Two input arguments required.')
c      CREATE A VECTOR FOR THE RETURN ARGUMENT.
         plhs(1) = mxCreateFull(m_in, n_in, 0)
c      ASSIGN POINTERS TO THE VARIOUS PARAMETERS.
         pX   = mxGetPr(prhs(2))
         pOUT = mxGetPr(plhs(1))
```

```
c       CALL THE COMPUTATIONAL SUBROUTINE.
           CALL CGRAD(size, %val(pX), %val(pOUT), IpINFO)

c CALCULATE THE HESSIAN MATRIX OF THE OBJECTIVE FUNCTION AT POINT X
c MULTIPLIED BY VECTOR V.
        ELSEIF (Sname .EQ. 'Hv') THEN
c       CHECK FOR PROPER NUMBER OF INPUT ARGUMENTS.
           IF (nrhs .NE. 3)
     +        CALL mexErrMsgTxt('Three input arguments required.')
c       CHECK THAT INPUT #3 IS NUMERIC.
           IF (mxIsNumeric(prhs(3)) .EQ. 0)
     +        CALL mexErrMsgTxt('Input #3 must be numeric.')
c       GET THE DIMENSIONS OF INPUT #3 ARGUMENT.
           n_in  = mxGetN(prhs(3))
           sizeV = mxGetM(prhs(3)) * n_in
c       CHECK THAT INPUT #3 IS A COLUMN VECTOR.
           IF (n_in .NE. 1)
     +        CALL mexErrMsgTxt('Input #3 must be a column vector.')
c       CHECK THAT INPUT VECTOR #2 AND INPUT VECTOR #3 HAVE THE SAME
c       DIMENSION.
           IF (size .NE. sizeV)
     +        CALL mexErrMsgTxt('Input vector #2 and input vector' //
     +                          ' #3 must have the same dimension.')
c       CREATE A VECTOR FOR THE RETURN ARGUMENT.
           plhs(1) = mxCreateFull(sizeV, 1, 0)
c       ASSIGN POINTERS TO THE VARIOUS PARAMETERS.
           pX   = mxGetPr(prhs(2))
           pV   = mxGetPr(prhs(3))
           pOUT = mxGetPr(plhs(1))
c       CALL THE COMPUTATIONAL SUBROUTINE.
           CALL CHESSV(size, %val(pX), %val(pV), %val(pOUT), IpINFO)
```

```
c CALCULATE THE HESSIAN MATRIX OF THE OBJECTIVE FUNCTION AT POINT X.
      ELSEIF (Sname .EQ. 'H') THEN
c       CHECK FOR PROPER NUMBER OF INPUT ARGUMENTS.
         IF (nrhs .NE. 2)
     +      CALL mexErrMsgTxt('Two input arguments required.')
c       CREATE A MATRIX FOR THE RETURN ARGUMENT.
         plhs(1) = mxCreateFull(size, size, 0)
c       ASSIGN POINTERS TO THE VARIOUS PARAMETERS.
         pX   = mxGetPr(prhs(2))
         pOUT = mxGetPr(plhs(1))
c       CALL THE COMPUTATIONAL SUBROUTINE.
         CALL CHESS(size, %val(pX), %val(pOUT), size, IpINFO)

c BREAK THE EXECUTION WHEN THE VALUE OF INPUT #1 IS IMPROPER.
      ELSE
         CALL mexErrMsgTxt('Improper value of input #1.')
      ENDIF
c CHECK THE INFORMATION ON SUBROUTINE EXECUTION.
      IF (IpINFO .NE. 0) THEN
         WRITE(*,*)
    5    FORMAT (A10,I2)
         WRITE(*,5) '   INFO =', IpINFO
         WRITE(*,*) '   Error in the execution of the',
     +             ' computational subroutine.'
         CALL mexErrMsgTxt('Invalid dimension of the input vector.')
      ENDIF

c RETURN THE OUTPUT ARGUMENTS TO MATLAB AND END THE EXECUTION OF THE
c MEX-FILE.
      RETURN
     END
```

```
c     SUBROUTINE F21
c
c     THIS IS THE GATEWAY ROUTINE OF A FORTRAN MEX-FILE FOR MATLAB.
c
c     Gateway  subroutine  that  interfaces  MATLAB  with  a  FORTRAN
c     computational  routine that was  provided  by Shanjie Zhang and
c     Jianming Jin in the disk that accompanies the book
c                   COMPUTATION OF SPECIAL FUNCTIONS
c               Copyright 1996 by John Wiley & Sons, Inc.
c
c     The computational routine called by this gateway is
c               SUBROUTINE HYGFX ( A, B, C, X, HF )
c     which computes the Gauss hypergeometric function 2_F_1(a,b,c,x)
c     with real  parameters  a, b, c, such  that  c  <>  0,-1,-2,...,
c     and real arguments x < 1.
c
c
c     NOTE: if your FORTRAN compiler does not support the %val
c           construct, use mxCopy_ routines to pass the values
c           of the variables, rather than the addresses of the
c           variables, to the computational routines.
c
c=====================================================================


c     THE GATEWAY SUBROUTINE

      SUBROUTINE mexFunction(nlhs, plhs, nrhs, prhs)

       IMPLICIT NONE
```

```
c-------------------------------------------------------------------
c      (INTEGER) Replace INTEGER*8 by INTEGER if not on the DEC Alpha
c      or the SGI 64-bit platforms.

       INTEGER*8 plhs(*), prhs(*)
       INTEGER*8 mxGetPr, mxCreateFull, mxGetString
c-------------------------------------------------------------------

       INTEGER nlhs, nrhs, mxGetM, mxGetN

c      KEEP THE ABOVE SUBROUTINE, ARGUMENT, AND FUNCTION DECLARATIONS
c      FOR USE IN ALL YOUR FORTRAN MEX-FILES.

c===================================================================

c-------------------------------------------------------------------
c      (INTEGER) Replace INTEGER*8 by INTEGER if not on the DEC Alpha
c      or the SGI 64-bit platforms.

       INTEGER*8 pa, pb, pc, px, pF
c-------------------------------------------------------------------

       INTEGER mxIsNumeric, mxIsComplex

c CHECK FOR PROPER NUMBER OF ARGUMENTS.
       IF (nlhs .NE. 1)
     +    CALL mexErrMsgTxt('One output argument required.')
       IF (nrhs .NE. 4)
     +    CALL mexErrMsgTxt('Four input arguments required.')
```

```
c CHECK THAT INPUT ARGUMENTS ARE NUMERIC.
      IF (mxIsNumeric(prhs(1)) .EQ. 0)
     +    CALL mexErrMsgTxt('Input #1 must be numeric.')
      IF (mxIsNumeric(prhs(2)) .EQ. 0)
     +    CALL mexErrMsgTxt('Input #2 must be numeric.')
      IF (mxIsNumeric(prhs(3)) .EQ. 0)
     +    CALL mexErrMsgTxt('Input #3 must be numeric.')
      IF (mxIsNumeric(prhs(4)) .EQ. 0)
     +    CALL mexErrMsgTxt('Input #4 must be numeric.')

c CHECK THAT INPUT ARGUMENTS ARE REAL.
      IF (mxIsComplex(prhs(1)) .EQ. 1)
     +    CALL mexErrMsgTxt('Input #1 must be real.')
      IF (mxIsComplex(prhs(2)) .EQ. 1)
     +    CALL mexErrMsgTxt('Input #2 must be real.')
      IF (mxIsComplex(prhs(3)) .EQ. 1)
     +    CALL mexErrMsgTxt('Input #3 must be real.')
      IF (mxIsComplex(prhs(4)) .EQ. 1)
     +    CALL mexErrMsgTxt('Input #4 must be real.')

c CHECK THAT INPUT ARGUMENTS ARE SCALARS.
      IF ((mxGetM(prhs(1)) .NE. 1) .OR. (mxGetN(prhs(1)) .NE. 1))
     +    CALL mexErrMsgTxt('Input #1 must be a scalar.')
      IF ((mxGetM(prhs(2)) .NE. 1) .OR. (mxGetN(prhs(2)) .NE. 1))
     +    CALL mexErrMsgTxt('Input #2 must be a scalar.')
      IF ((mxGetM(prhs(3)) .NE. 1) .OR. (mxGetN(prhs(3)) .NE. 1))
     +    CALL mexErrMsgTxt('Input #3 must be a scalar.')
      IF ((mxGetM(prhs(4)) .NE. 1) .OR. (mxGetN(prhs(4)) .NE. 1))
     +    CALL mexErrMsgTxt('Input #4 must be a scalar.')
```

```
c CREATE A SCALAR FOR THE RETURN ARGUMENT.
       plhs(1) = mxCreateFull(1, 1, 0)

c ASSIGN POINTERS TO THE VARIOUS PARAMETERS.
       pa = mxGetPr(prhs(1))
       pb = mxGetPr(prhs(2))
       pc = mxGetPr(prhs(3))
       px = mxGetPr(prhs(4))
       pF = mxGetPr(plhs(1))

c CALL THE COMPUTATIONAL SUBROUTINE.
       CALL HYGFX(%val(pa), %val(pb), %val(pc), %val(px), %val(pF))

c RETURN THE OUTPUT ARGUMENTS TO MATLAB AND END THE EXECUTION OF THE
c MEX-FILE.
       RETURN
      END
```

Appendix C

NUMERICAL RESULTS

In this section we give the results of the numerical testing of SUMEBADEA, which was performed on a Silicon Graphics Origin 2000 running IRIX64 release 6.5.6m. The set of termination criteria that were applied within the algorithm to indicate a successful termination is the following:

$$\begin{aligned}
&\|\nabla F(x_k)\|_2 \le \sqrt[3]{\tau_F}\big(1 + |F(x_k)|\big),\\
&\lambda_n(x_k) \ge \tau_F,\\
&F(x_{k-1}) - F(x_k) < \tau_F\big(1 + |F(x_k)|\big),\\
&\|x_{k-1} - x_k\|_2 < \sqrt{\tau_F}\big(1 + \|x_k\|_2\big).
\end{aligned}$$

We chose τ_F equal to 10^{-12}. Table C.1 on page 89 contains the performance of routine SUMEBADEA on each test problem in terms of number of iterations l until termination and the 2-norm of the gradient vector $\nabla F(x_l)$.

We also compared the performance of our routine SUMEBADEA against that of Behrman's implementation of a gradient flow method, and against that of two leading commercial routines that also use first and second derivatives. For a gradient flow method, we used the routine UMINH. For a linesearch method, we used the routine E04LBF from the NAG library. For a trust-region method, we used the routine DMNH from the AT&T PORT library. The test problems were run on these routines using the routines' default or suggested parameter values.

With the limit value $F(x^*)$ of the objective function known for each test problem, we were also able to use the reduction in error

$$\varepsilon_k = \left| \frac{F(x_k) - F(x^*)}{F(x_0) - F(x^*)} \right|$$

as a uniform measure of the routines' convergence. We recorded the number of iterations required by each routine to achieve tolerance level $\varepsilon_k = 10^{-12}$.

Table C.1 also includes a column with the rank, first, second, third or fourth, of SUMEBADEA on a given test problem. In the last column there is the ratio of SUMEBADEA's performance on the problem to that of the best routine's performance.

Table C.1: Number of iterations l until termination, 2-norm of the gradient vector at termination, and place finishes for number of iterations, tolerance = 10^{-12}.

Function	Problem	Size	l	$\|\nabla F(x_l)\|_2$	Rank	Ratio
ARGAUS	ARGAUS	3	3	2.47E-16	1	1
ARGQDN	ARGQDN50	5	2	0.00E+00	1	1
ARTRIG	ARTRIG10	10	11	7.45E-12	2	1.11
AVRIEL	AVRIEL3	2	2	4.44E-16	1	1
BARD70	BARD70	3	12	4.98E-16	2	1.14
BEAL58	BEAL58KO	2	11	2.90E-15	2	1.12
BOOTH	BOOTH	2	2	3.55E-15	1	1
BOX66	BOX662HL	2	13	1.33E-16	2	1.38
BRKMCC	BRKMCC	2	4	4.22E-15	1	1
BROWND	BROWND	4	9	3.45E-11	2	1.14
BROY7D	BROY7D	60	28	2.22E-10	3	2.45
BRWNAL	BRWNL100	100	6	1.09E-11	1	1
BRYBND	BRYBND18	100	8	1.39E-12	1	1
CLUSTR	CLUSTR	2	14	3.00E-18	2	1.5
CRGLVY	CRGLY500	500	17	2.65E-14	1	1
DIXON	DIXON	10	2	0.00E+00	1	1
EXTRSN	EXTRA100	100	25	0.00E+00	2	1.05
FRDRTH	FRDRTHB3	50	7	1.75E-12	1	1
GOTTFR	GOTTFR	2	20	1.17E-16	2	1.62
HILBRT	HILBRT12	12	5	2.06E-23	1	1
HIMM1	HIMM1	2	1	0.00E+00	1	1
HIMM25	HIMM25	2	1	0.00E+00	1	1
MANCIN	MANCIN50	50	4	5.36E-09	1	1
NONDIA	NONDIA20	20	26	8.66E-13	3	1.26
PENAL1	PEN1LN1	50	45	2.87E-17	1	1
PENAL1	PEN1LN2	100	45	5.04E-16	1	1
QUARTC	QUARTC	25	33	9.07E-13	1	1

SCHMVT	SCHMV500	500	4	4.51E-15	1	1
TDQUAD	TDQ500	500	2	9.86E-32	1	1
TOINT	PSPTOINT	50	12	5.61E-14	1	1
TRIDIA	TRLN100	100	2	6.86E-13	1	1
WATSON	WATSON6	6	16	9.56E-12	3	1.36
WOODS	WOODS	4	50	6.60E-14	3	1.3
WOODS	WOODS80	80	47	3.18E-13	2	1.22
XTX	XTX16	16	2	1.41E-31	1	1
XTX	XTX2	2	2	4.97E-32	1	1
ZANGWL	ZANGWL1	3	2	4.19E-29	1	1

Bibliography

[1] Filippo Aluffi-Pentini, Valerio Parisi and Francesco Zirilli. A differential-equations algorithm for nonlinear equations. *ACM Transactions on Mathematical Software*, vol. 10, no. 3, September 1984, pp. 299–316.

[2] Filippo Aluffi-Pentini, Valerio Parisi and Francesco Zirilli. Algorithm 617. DAFNE: a differential-equations algorithm for nonlinear equations. *ACM Transactions on Mathematical Software*, vol. 10, no. 3, September 1984, pp. 317–324.

[3] Antonio Ambrosetti. Problemi variazionali in analisi non lineare. *Bollettino della Unione Matematica Italiana,* Serie VII, vol. 2-A, no. 2, June 1988, pp. 169–188.

[4] Antonio Ambrosetti and Paul H. Rabinowitz. Dual variational methods in critical point theory and applications. *Journal of Functional Analysis*, vol. 14, no. 4, December 1973, pp. 349–381.

[5] George E. Andrews, Richard Askey and Ranjan Roy. *Special Functions*, vol. 71 of *Encyclopedia of Mathematics and Its Applications.* Cambridge University Press, Cambridge, U.K., 1999.

[6] William Behrman. *An Efficient Gradient Flow Method for Unconstrained Optimization.* Ph. D. Dissertation, Program in Scientific Computing and Computational Mathematics, Stanford University, Stanford, California, June 1998.

[7] Melvyn Stuart Berger. *Nonlinearity and Functional Analysis: Lectures on Nonlinear Problems in Mathematical Analysis.* Academic Press, Inc., New York, 1977.

[8] Paul T. Boggs. An algorithm, based on singular perturbation theory, for ill-conditioned minimization problems. *SIAM Journal on Numerical Analysis*, vol. 14, no. 5, October 1977, pp. 830–843.

[9] Erik G. Boman. *Infeasibility and Negative Curvature in Optimization.* Ph. D. Dissertation, Program in Scientific Computing and Computational Mathematics, Stanford University, Stanford, California, February 1999.

[10] C. A. Botsaris. Differential gradient methods. *Journal of Mathematical Analysis and Applications*, vol. 63, no. 1, March 1978, pp. 177–198.

[11] C. A. Botsaris. A curvilinear optimization method based upon iterative estimation of the eigensystem of the Hessian matrix. *Journal of Mathematical Analysis and Applications*, vol. 63, no. 2, April 1978, pp. 396–411.

[12] C. A. Botsaris. A class of methods for unconstrained minimization based on stable numerical integration techniques. *Journal of Mathematical Analysis and Applications*, vol. 63, no. 3, May 1978, pp. 729–749.

[13] C. A. Botsaris and David H. Jacobson. A Newton-type curvilinear search method for optimization. *Journal of Mathematical Analysis and Applications*, vol. 54, no. 1, April 1976, pp. 217–229.

[14] Felix E. Browder. Infinite dimensional manifolds and non-linear elliptic eigenvalue problems. *Annals of Mathematics,* Second Series, vol. 82, no. 3, November 1965, pp. 459–477.

[15] A. A. Brown and M. C. Bartholomew-Biggs. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *Journal of Optimization Theory and Applications*, vol. 62, no. 2, August 1989, pp. 211–224.

[16] Albert G. Buckley. Test functions for unconstrained minimization. *Technical Report*, Department of Mathematics, Royal Roads Military College, FMO Victoria, British Columbia, Canada, November 1994.

[17] Henri Paul Cartan. *Differential Calculus*. Hermann, Paris, France, 1983.

[18] Richard Warren Cottle, Franco Giannessi and Jacques-Louis Lions, editors. *Variational Inequalities and Complementarity Problems: Theory and Applications*. John Wiley & Sons, Ltd., Chichester, U.K., 1980.

[19] Richard Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, vol. 49, no. 1, January 1943, pp. 1–23.

[20] Richard Courant. *Calculus of Variations and Supplementary Notes and Exercises, 1945-1946: Revised and amended by J. Moser*. New York University, Courant Institute of Mathematical Sciences, New York, 1962.

[21] Arthur Erdélyi, editor. Wilhelm Magnus, Fritz Oberhettinger and Francesco Giacomo Tricomi, research associates. *Higher Transcendental Functions. Based, in part, on notes left by Harry Bateman, and compiled by the staff of the Bateman Manuscript Project.* Bateman Manuscript Project, McGraw-Hill Book Company, Inc., New York, 1953-1955.

[22] Arthur Erdélyi, editor. Wilhelm Magnus, Fritz Oberhettinger and Francesco Giacomo Tricomi, research associates. *Tables of Integral Transforms. Based, in part, on notes left by Harry Bateman, and compiled by the staff of the Bateman Manuscript Project.* Bateman Manuscript Project, McGraw-Hill Book Company, Inc., New York, 1954.

[23] Anthony V. Fiacco and Garth Philip McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley and Sons, Inc., New York, 1968.

[24] Roger Fletcher and T. L. Freeman. A modified Newton method for minimization. *Journal of Optimization Theory and Applications*, vol. 23, no. 3, 1977, pp. 357–372.

[25] Anders Forsgren, Philip E. Gill and Walter Murray. Computing modified Newton directions using a partial Cholesky factorization. *Technical Report SOL 93-1*, Department of Operations Research, Stanford University, Stanford, California, March 1993.

[26] Anders Forsgren and Walter Murray. Newton methods for large-scale linear equality-constrained minimization. *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 2, April 1993, pp. 560–587.

[27] Philip E. Gill and Walter Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Mathematical Programming*, vol. 7, no. 3, December 1974, pp. 311–350.

[28] Philip E. Gill, Walter Murray and Margaret H. Wright. *Practical Optimization*. Academic Press Limited, London, U.K., 1981.

[29] Donald Goldfarb. Curvilinear path steplength algorithms for minimization which use directions of negative curvature. *Mathematical Programming*, vol. 18, no. 1, January 1980, pp. 31–40.

[30] Gene Howard Golub and Charles F. Van Loan. *Matrix Computations*. Third edition. The Johns Hopkins University Press, Baltimore, Maryland, 1996.

[31] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The LaTeX Companion*. Addison Wesley Longman, Inc., Reading, Massachusetts, 1994.

[32] Izrail′ Solomonovich Gradshteyn and Iosif Moiseevich Ryzhik; Alan Jeffrey, editor. *Table of Integrals, Series, and Products*. Fifth edition. Academic Press, Inc., Boston, Massachusetts, 1994.

[33] Jacques Hadamard. Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées. *Mémoires présentés par divers savants à l'Académie des Sciences de l'Institut National de France, Series 2*, vol. 33, no. 4, 1908, pp. 1–128.

[34] Shmuel Kaniel and Achiya Dax. A modified Newton's method for unconstrained minimization. *SIAM Journal on Numerical Analysis*, vol. 16, no. 2, 1979, pp. 324–331.

[35] David Kinderlehrer and Guido Stampacchia. *An Introduction to Variational Inequalities and Their Applications.* Academic Press, Inc., New York, 1980.

[36] Leslie Lamport. *LaTeX: A Document Preparation System - User's Guide and Reference Manual.* Second edition. Addison Wesley Longman, Inc., Reading, Massachusetts, 1994.

[37] David G. Luenberger. *Optimization by Vector Space Methods.* John Wiley & Sons, Inc., New York, 1969.

[38] MATLAB®. *Application Program Interface Guide: Version 5.* The MathWorks, Inc., Natick, Massachusetts, 1998.

[39] Garth Philip McCormick. A modification of Armijo's step-size rule for negative curvature. *Mathematical Programming*, vol. 13, no. 1, August 1977, pp. 111–115.

[40] Jorge J. Moré. Recent developments in algorithms and software for trust region methods. Achim Bachem, Martin Grötschel and Bernhard H. Korte, editors, *Mathematical Programming: The State of the Art - Bonn 1982.* Springer-Verlag, Berlin, Germany, 1983, pp. 258–287.

[41] Jorge J. Moré and Danny C. Sorenson. On the use of directions of negative curvature in a modified Newton method. *Mathematical Programming*, vol. 16, no. 1, January 1979, pp. 1–20.

[42] Jorge J. Moré and Danny C. Sorenson. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 3, January 1983, pp. 553–572.

[43] Hiro Mukai and Èlijah Polak. A second-order method for unconstrained optimization. *Journal of Optimization Theory and Applications*, vol. 26, no. 4, December 1978, pp. 501–513.

[44] Stephen G. Nash and Ariela Sofer. *Linear and Nonlinear Programming.* The McGraw-Hill Companies, Inc., New York, 1996.

[45] Jorge Nocedal and Stephen Joseph Wright. *Numerical Optimization.* Springer-Verlag New York, Inc., New York, 1999.

[46] James M. Ortega and Werner C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 2000.

[47] Richard S. Palais. Morse theory on Hilbert manifolds. *Topology: An International Journal of Mathematics*, vol. 2, no. 4, December 1963, pp. 299–340.

[48] Richard S. Palais. Lusternik-Schnirelman theory on Banach manifolds. *Topology: An International Journal of Mathematics*, vol. 5, no. 2, May 1966, pp. 115–132.

[49] Richard S. Palais and Stephen Smale. A generalized Morse theory. *Bulletin of the American Mathematical Society*, vol. 70, no. 1, January 1964, pp. 165–172.

[50] Stefan Schäffler and Hubert Warsitz. A trajectory-following method for unconstrained optimization. *Journal of Optimization Theory and Applications*, vol. 67, no. 1, October 1990, pp. 133–140.

[51] Gerald A. Shultz, Robert B. Schnabel and Richard H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong

global convergence properties. *SIAM Journal on Numerical Analysis*, vol. 22, no. 1, February 1985, pp. 47–67.

[52] Stephen Smale. Morse theory and a non-linear generalization of the Dirichlet problem. *Annals of Mathematics,* Second Series, vol. 80, no. 2, September 1964, pp. 382–396.

[53] Guido Stampacchia. Variational inequalities. *Actes du Congrès International des Mathématiciens, 1970*, vol. 2 (D). Publiés sous la direction du Comité d'Organisation du Congrès, Gauthier-Villars Éditeur, quai des Grands-Augustins, n° 55, Paris, France, 1971, pp. 877–883.

[54] Gilbert Strang. *Linear Algebra and Its Applications.* Third edition. Harcourt Brace Jovanovich College Publishers, San Diego, California, 1988.

[55] Michael Struwe. *Variational Methods: Applications to Nonlinear Partial Differential Equations and Hamiltonian Systems.* Second, revised and substantially expanded edition. Springer-Verlag, Berlin and Heidelberg, Germany, 1996.

[56] Jean-Philippe Vial and Israel Zang. Unconstrained optimization by approximation of the gradient path. *Mathematics of Operations Research*, vol. 2, no. 3, August 1977, pp. 253–265.

[57] Israel Zang. A new arc algorithm for unconstrained optimization. *Mathematical Programming*, vol. 15, no. 1, July 1978, pp. 36–52.

[58] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications I: Fixed-Point Theorems.* Springer-Verlag New York, Inc., New York, 1986.

[59] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications II/A: Linear Monotone Operator.* Springer-Verlag New York, Inc., New York, 1990.

[60] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications II/B: Nonlinear Monotone Operator.* Springer-Verlag New York, Inc., New York, 1990.

[61] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications III: Variational Methods and Optimization.* Springer-Verlag New York, Inc., New York, 1985.

[62] Eberhard Zeidler. *Nonlinear Functional Analysis and its Applications IV: Applications to Mathematical Physics.* Springer-Verlag New York, Inc., New York, 1988.

[63] Shanjie Zhang and Jianming Jin. *Computation of Special Functions.* John Wiley & Sons, Inc., New York, 1996.