

**Multilevel Algorithms for  
Nonlinear Optimization**

*Natalia Alexandrov  
J.E. Dennis, Jr.*

**CRPC-TR94466  
June, 1994**

Center for Research on Parallel Computation  
Rice University  
P.O. Box 1892  
Houston, TX 77251-1892

This work was supported in part by the National Aeronautic and Space Administration.



# MULTILEVEL ALGORITHMS FOR NONLINEAR OPTIMIZATION

Natalia Alexandrov  
ICASE  
Mail Stop 132C  
NASA Langley Research Center  
Hampton, Virginia 23681-0001  
natalia@icase.edu

J. E. Dennis, Jr.  
Department of Computational and Applied Mathematics  
Rice University  
P. O. Box 1892  
Houston, Texas 77251-1892

## ABSTRACT

Multidisciplinary design optimization (MDO) gives rise to nonlinear optimization problems characterized by a large number of constraints that naturally occur in blocks. We propose a class of multilevel optimization methods motivated by the structure and number of constraints and by the expense of the derivative computations for MDO. The algorithms are an extension to the nonlinear programming problem of the successful class of local Brown-Brent algorithms for nonlinear equations. Our extensions allow the user to partition constraints into arbitrary blocks to fit the application, and they separately process each block and the objective function, restricted to certain subspaces. The methods use trust regions as a globalization strategy, and they have been shown to be globally convergent under reasonable assumptions. The multilevel algorithms can be applied to all classes of MDO formulations. Multilevel algorithms for solving nonlinear systems of equations are a special case of the multilevel optimization methods. In this case, they can be viewed as a trust-region globalization of the Brown-Brent class.

---

Research was supported by AFOSR-89-0363, AFOSR F49620-92-J0203, DOE DE-FG05-86ER25017, NSF/CRPC CCR 9120008.

This work was supported in part by the National Aeronautic and Space Administration under NASA Contract No. NAS1-19480 while the first author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

# 1 Introduction

This work is concerned with a class of methods, called multilevel optimization algorithms, for solving the nonlinear equality constrained optimization problem, i.e.,

**Problem EQC:**

$$\begin{aligned} & \text{minimize } f(x) \\ & \text{subject to } C(x) = 0, \end{aligned}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $C : \mathbb{R}^n \rightarrow \mathbb{R}^m, m \leq n$ , are at least twice continuously differentiable.

The proposed class of algorithms can be used to solve any general nonlinear equality constrained optimization problem, but its development has been motivated by the engineering design problems that give rise to large-scale optimization formulations with constraints occurring naturally in blocks. In particular, in the multidisciplinary design optimization (MDO) environment, the sheer number of constraints, the structure of the problems, and the expense of the derivative computations necessitate the development of flexible algorithms that allow the user to partition the problem into a set of smaller problems.

While there is a number of nonlinear optimization methods that attack large problems by decomposing them into several smaller ones, these methods require the problems to have a special structure, for example, separability and convexity.

In particular, in engineering, decomposition and multilevel optimization have been used to solve large problems for some time. See [21] and [31] for a survey. The process of decomposition and multilevel formulation generally depends on identifying groups of variables and constraints that influence each other only weakly. The problem is then decomposed into such weakly coupled subproblems in various possible formulations, some hierarchic, some nonhierarchic. Recent developments in formulations can be found in [5] and [11]. Some of the approaches in [5] have been proven to be successful for many problems. In order to be more widely applicable, it requires the development of theoretical foundations.

We propose a class of multilevel optimization methods (see [2], [3]) for solving the nonlinear equality constrained optimization problem characterized by the following features:

- The constraints of the problem can be partitioned into blocks in any manner suitable to an application, or in any arbitrary manner at all. The analysis of the methods assumes certain standard smoothness and boundedness properties, but no other assumptions are made on the structure of the problem. There is no need to identify the weakly coupled groups of variables and constraints, although that may be helpful in practice. If all constraints and variables are strongly coupled, the partitioning can be done according to any other criterion useful to a particular application, for example, just the size of constraint blocks. The algorithm then solves progressively smaller dimensional subproblems to arrive at the trial step.
- The multilevel methods belong to the class of out-of-core methods. To the authors' knowledge, the multilevel algorithms are the only algorithms for general nonlinear optimization problems that require only a currently processed part of the constraints to be held in memory. Thus, theoretically, there is no limit to the size of the problem the methods can handle.

- The trial steps computed by the algorithm are required to satisfy very mild conditions, both theoretically and computationally. In fact, the substeps comprising the trial step can be computed in the subproblems using different optimization algorithms. The substeps are only required to satisfy a mild decrease condition for the subproblems and a reasonable boundedness condition—both satisfied in practice by most methods of interest. This feature is of great practical significance because in applications like MDO various constraint blocks may originate from different disciplines and may require different approaches to solving the subproblems.
- The class uses trust regions as a globalization strategy. The algorithms are proven to converge under reasonable assumptions.
- The algorithms together with their convergence theory provide a foundation for developing the algorithms and analyses of the general multilevel optimization formulations.

The proposed multilevel class of algorithms differs from the conventional algorithms in that its major iteration involves computing an approximate solution of not one model over a single restricted region, but of a sweep of models, each approximately minimized over its own restricted region. Each model approximates a block of constraints and, finally, the objective function, restricted to certain subspaces. Each model is computed at a different point. The case of a single block of constraints is included.

In the next section we introduce the foundations on which the proposed class of algorithms rests. Section 3 is devoted to the description of the class. Section 4 briefly describes current theoretical results. Section 5 concludes with a summary and discussion of current and future research.

## 2 Preliminaries

The proposed class of algorithms may be viewed as an extension of several areas of research. In this section we describe the existing algorithms and analysis schemes which serve as a foundation for the multilevel optimization methods.

### 2.1 The Local Brown-Brent Class of Methods

Theoretical origins of this research lie in the method for solving nonlinear systems of equations,  $F(x) = 0$ ,  $F : \mathbb{R}^n \rightarrow \mathbb{R}$ , introduced by Brown in [7], [8], [9]. In [6], Brent viewed Brown's method from a different perspective, which allowed Brent to propose a class of methods, among which Brown's original method was a special case. Gay [19] and Martinez [25], [26] provided further modifications and generalizations of the methods.

The following statement of the general Brown-Brent algorithm was condensed from the descriptions in Gay [19] and Dennis [12]. In these works the algorithm is described in terms of one-dimensional blocks.

Denote the components of  $F(x)$  by  $F_1(x), \dots, F_n(x)$ .

#### Algorithm 2.1 Local Brown-Brent Algorithm for Nonlinear Systems

Let  $x_c$  be the current approximation to the solution.

**Outer Loop:** Do until convergence:

$$y_0 = x_c$$

$$H_0 = \mathbb{R}^n$$

**Inner Loop:** Do  $k = 1, n$

1. Form the linearization,  $L_k$  about  $y_{k-1}$  of  $F_k$  restricted to  $\bigcap_{i=0}^{k-1} H_i$ .

$L_k = 0$  defines  $H_k$ , an  $(n - k)$ -dimensional hyperplane in  $\mathbb{R}^n$ .

2. Move from  $y_{k-1} \in \bigcap_{i=0}^{k-1} H_i$  to  $y_k \in \bigcap_{i=0}^k H_i$ .

**End Inner Loop**

$$x_c = y_n$$

**End Outer Loop**

The point  $y_n$  of intersection of all the hyperplanes is the point where all the linearizations vanish. The way in which the steps 1-2 of the inner loop are actually done determines the particular kind of Brown-Brent method. In Brent's method,  $s_k = y_k - y_{k-1}$  is the shortest  $\ell_2$  norm step from  $y_{k-1}$  to  $H_k$ . In Brown's method,  $s_k$  is the shortest  $\ell_2$  norm step from  $y_{k-1}$  to  $H_k$  parallel the  $k$ -th coordinate axis.

When applied to a linear system of equations, i.e., when  $F(x) = Ax - b$ , Brown's method is equivalent to Gaussian elimination with pivoting about the maximum row element of the reduced matrix [7], while Brent's method is equivalent to factoring  $A$  into a product of a lower triangular matrix and an orthogonal matrix [6]. It can be shown, based on [33], that there exists a Brown-Brent analog for any matrix decomposition in the linear case.

Brown [7], [9], Brown and Dennis [10], Brent [6], and Gay [19] established local quadratic convergence of variants of the algorithm, both for analytic and finite difference derivatives. To the authors' knowledge, there had been no theoretically supported global extensions of Brown-Brent methods until [2].

## 2.2 Trust-Region Methods

Consider the following unconstrained minimization problem.

**Problem UNC:**

$$\begin{aligned} &\text{minimize } f(x) \\ &x \in \mathbb{R}^n, \end{aligned}$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuously differential. Given  $x_c$ , the current approximation to the solution, a trust-region algorithm for solving the problem finds a trial step by solving the following trust-region subproblem approximately:

$$\begin{aligned} &\text{minimize } f(x_c) + \nabla f(x_c)^T s + \frac{1}{2} s^T H_c s \\ &\text{subject to } \|s\| \leq \delta_c, \end{aligned} \tag{1}$$

where  $f, \delta_c \in \mathbb{R}$ ,  $\nabla f, s \in \mathbb{R}^n$ ,  $H_c = H_c^T \in \mathbb{R}^{n \times n}$  is the Hessian of  $f$  or an approximation to it,  $\delta_c > 0$  is the trust-region radius, and  $\|\cdot\|$  denotes the  $\ell_2$  norm. The idea is to accept the trial step when the quadratic model adequately predicts the behavior of the function, and to recompute the step in a smaller region if it does not.

The trust-region approach to the problem of solving systems of nonlinear equations is just a special case of the approach to the problem above; namely, for nonlinear equations, the objective function  $f(x)$  is taken to be  $\|F(x)\|_2^2$ .

Detailed treatment of the trust-region approach to unconstrained optimization and nonlinear equations can be found in Dennis and Schnabel [14], Sorensen [32], Moré [27], Moré and Sorensen [28], Powell [29], and Shultz, Schnabel and Byrd[30].

For the equality constrained optimization problem, the successive quadratic programming (SQP) algorithm is used commonly. Its step is found by computing a minimum of the quadratic model of the Lagrangian at the current point, subject to linearized constraints. A trust-region algorithm based on SQP adds the trust-region constraint to the subproblem and additional constraints designed to ensure that the trust-region constraint and the linearized constraints are consistent.

### 2.2.1 Merit Functions

In order to evaluate a trial step, trust-region algorithms use merit functions, which are functions related to the problem in such a way that the improvement in the merit function signifies progress toward the solution of the problem.

For unconstrained minimization, a natural choice for a merit function is the objective function itself. Let

$$\phi(s) = f(x_c) + \nabla f(x_c)^T s + \frac{1}{2} s^T H_c s \quad (2)$$

denote the quadratic model of the merit function. We define two related functions.

**The actual reduction** is defined as

$$ared_c(s_c) = f(x_c) - f(x_c + s_c), \quad (3)$$

and **the predicted reduction** is defined as

$$\begin{aligned} pred_c(s_c) &= \phi(0) - \phi(s_c) \\ &= -\nabla f(x_c)^T(s_c) - \frac{1}{2} s_c^T H_c s_c, \end{aligned} \quad (4)$$

so that the predicted reduction in the merit function is an approximation to the actual reduction in the merit function.

The standard way to evaluate the trial step in trust-region methods is to consider the ratio of the actual reduction to the predicted reduction. A value lower than a small predetermined value causes the step to be rejected. Otherwise the step is accepted.

For nonlinear systems of equations, the norm of the residuals serves as a merit function. For the constrained optimization, the merit function is some expression that involves both the objective function and the constraints.

We shall see that conventional merit functions prove to be inadequate for multilevel algorithms.

### 2.2.2 Fraction of Cauchy Decrease

To assure global convergence of a trust-region algorithm for problem UNC, the trial step is required to satisfy a **fraction of Cauchy decrease** condition. This mild condition means that the trial

step,  $s_c$ , must predict at least a fraction of the decrease predicted by the Cauchy step, which is the steepest descent step for the model within the trust region. We must have for some fixed  $\kappa > 0$

$$\phi(s_c) - \phi(0) \leq \kappa[\phi(s_c^{CP}) - \phi(0)], \quad (5)$$

where

$$s_c^{CP} = -\alpha_c^{CP} \nabla f(x_c) \text{ with } \alpha_c^{CP} = \begin{cases} \frac{\|\nabla f(x_c)\|^2}{\nabla f(x_c)^T H_c \nabla f(x_c)} & \text{if } \frac{\|\nabla f(x_c)\|^3}{\nabla f(x_c)^T H_c \nabla f(x_c)} \leq \delta_c \\ \frac{\delta_c}{\|\nabla f(x_c)\|} & \text{otherwise.} \end{cases}$$

See Dennis and Schnabel [14], pp. 139–141, for details on the Cauchy point.

The fraction of Cauchy decrease property implies a weaker condition which has a more convenient form and is frequently used as a technical lemma in the global convergence proofs.

**Lemma 2.1** *Let  $s_c$  satisfy (5). Then*

$$\phi(0) - \phi(s_c) \geq \frac{\kappa}{2} \|\nabla f(x_c)\| \min \left\{ \frac{\|\nabla f(x_c)\|}{\|H_c\|}, \delta_c \right\}. \quad (6)$$

**References:** Powell [29]; Moré [27].

Either (5) or (6) is necessary to establish global convergence theoretically.

### 2.2.3 Global Convergence Results

Powell's global convergence theorem [29] for any unconstrained minimization trust-region algorithm serves as a prototype for most trust-region related convergence results.

**Theorem 2.1** *Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be continuously differentiable and bounded below on the level set  $\{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$ . Assume that  $\{H_i\}$  are uniformly bounded above. Let  $\{x_i\}$  be the sequence of iterates generated by a trust-region algorithm that satisfies (5) or (6). Then*

$$\liminf_{i \rightarrow \infty} \|\nabla f(x_i)\| = 0.$$

Detailed treatment of the unconstrained minimization theory and practice can be found in Moré [27], Moré and Sorensen [28], Sorensen [32], and Shultz, Schnabel and Byrd [30].

### 2.2.4 Tangent-Space Methods for Constrained Optimization

The multilevel methods proposed here may be viewed as a generalization of an approach to nonlinear programming known as the null space or generalized elimination approach (see Fletcher [18]).

Different authors refer to different methods as “null space methods”, but the general idea of a null space method for equality constrained minimization is to reduce the dimension of the problem by first taking the step intended to solve the constraint equations, and then to minimize the model of the function restricted to the null space of the linearized constraints. The resulting minimization problem is of a lower dimension than the original one.

A well-known local method of this type is the GRG (Generalized Reduced Gradient) algorithm. Details of GRG and other null space methods can be found in Lasdon [23], Fletcher [18], Avriel [4], and Gill, Murray and Wright [20].



A class of global trust-region algorithms that use the same general principle of reducing the problem's dimension is known as the class of tangent space methods. The tangent space approach was introduced to avoid the possibility of inconsistency of the constrained trust-region subproblem.

Recent work on these methods can be found in Maciel [24] and Dennis, El-Alem and Maciel [13]. The main feature of the class is that the trial step is computed as a sum of two substeps, the first of which is made toward the linearized constraints in the direction orthogonal to the null space of the constraint Jacobian, while the second is made to minimize the model of the Lagrangian in the null space of the linearized constraints. The function and derivative information is computed at a single point  $x_c$ .

The multilevel methods proposed here generalize the tangent space methods in the sense that their trial steps are sums of not two substeps but of as many substeps as there are constraint blocks together with a substep on the model of the objective function with the model information computed at the points resulting from taking the substeps one-by-one.

### 3 Multilevel Algorithms for Nonlinear Optimization

In this section we present the class of multilevel optimization algorithms for the nonlinear equality constrained minimization problem. Since the time of its introduction in [2], the class has undergone changes. In [2], the globalization and extension to constrained optimization only of local Brent's method was proposed. Recent developments (see [3]) have extended the results to provide globalization and extension to constrained optimization of the entire local Brown-Brent class.

#### 3.1 Notation

Due to arbitrary blocking of the constraints, the notation becomes cumbersome. To ease the reading effort, we omit the subscripts and superscripts where possible. Here is an explanation of the notation conventions.

Unless specified otherwise, all norms are  $\ell_2$  norms.

From here on, we assume that the equality constraints of problem EQC are partitioned into  $M$  blocks of arbitrary size and composition. Let the constraints of the first block be numbered from  $n_1 = 1$  to  $n_2 - 1$ ; the constraints of the second block—from  $n_2$  to  $n_3 - 1$ ; and so on, until the constraints of the last block are numbered from  $n_{M-1}$  to  $n_M = m$ .

The algorithms will be formally considered to have an outer loop, in which we make the decision about the acceptability of the step, and the inner loop, in which we solve a sequence of minimization subproblems. The sum of the substeps produced as solutions of these subproblems yields the total trial step. The outer loop counter is  $i$ ; the inner loop counter is  $k$ . Thus  $k$  corresponds to the block number of constraints. If the subscript  $k$  is used with a constant, that constant refers to the properties of the  $k$ -th block of constraints, independent of the iterates. Note that the term "inner loop" is formal. The purpose of the inner loop is to compute a basis for the null space of the Jacobian of our constraint system, but step-by-step, using information at different points, instead of the simultaneous computation of, say, the Newton's method.

We denote the sequence of points generated by the outer loop of the algorithm by  $\{x_i\}$  when we consider the iterates as members of the sequence in convergence analysis. Otherwise, we use  $x_c$ ,  $x_-$ , and  $x_+$  to denote the current, the previous, and the next iterates, respectively.

We denote the sequence of points generated within each inner loop by  $y_k$ ,  $k = 0, \dots, M + 1$ , when we need not consider the outer loop iteration number. Most of the time we shall be discussing entities within a single iteration. Otherwise, we use subscripts and superscripts. For example,  $y_k^i$  or  $y_k^c$  denote the inner  $k$ -th iterate within the  $i$ -th outer loop. Note that  $y_0 = x_c$  and  $y_{M+1} = x_+$ .

The substep produced by solving the  $k$ -th subproblem of the inner loop is denoted by  $s_k$ ,  $k = 1, \dots, M + 1$ . The sum  $s_1 + \dots + s_{M+1}$  yields the total trial step  $\hat{s}_c$ . Again, we use subscripts, e.g.,  $\hat{s}_i$ , to denote the total step as a part of the sequence of steps produced by the algorithm.

We denote the radius of the trust region for subproblem  $k$ , centered at  $y_{k-1}$ , by  $\delta_k$ ,  $k = 1, \dots, M + 1$ . The radius of the total trust region, centered at  $x_c = y_0$  is  $\hat{\delta}_c$  or  $\hat{\delta}_i$ .

We denote the projector onto the intersection of null spaces of  $\nabla C_1(x), \dots, \nabla C_k(x)$  by  $P_k$ .

Again, when we omit superscripts, we refer to the objects within a single outer loop. For example,  $C_k(y_{k-1})$  refers to  $C_k(y_{k-1}^i)$  or  $C_k(y_{k-1}^c)$ .

Additional notation will be introduced as needed.

### 3.2 General Description

The general class of multilevel algorithms can be described in the following way. The constraint system of the problem is partitioned into  $M$  arbitrary blocks. In practice, this block decomposition is obvious in most cases. At the current approximation to a solution of problem EQC,  $x_c$ , we set  $y_0 = x_c$ . The trial step is computed as follows.

We find an approximate minimizer,  $s_1$ , of the quadratic Gauss-Newton model about  $y_0$  of the first block of constraints in the trust region of radius  $\delta_1$ . The step is required to satisfy a fraction of Cauchy decrease condition for this model and a mild boundedness condition discussed in the next subsection. The step is taken to yield the point  $y_1 = y_0 + s_1$ .

We then find an approximate minimizer of the quadratic model of the second block of constraints, restricted to the null space of the Jacobian of the first block. This model is built using **the information at the new point**. It is important to emphasize that all the function and derivative information for the second block is computed at the new point  $y_1$ . The next step,  $s_2$ , bounded by its own trust-region, is obtained to satisfy a fraction of Cauchy decrease condition for this restricted model of the second block. The step is taken to yield the point  $y_2$ .

The process of computing steps that satisfy sufficient predicted decrease for the restricted models of progressively smaller dimensions continues. Again, **the model for each block is built by using the function and derivative information at the most recently computed point**. The final step on the constraints,  $s_M$ , is obtained to produce sufficient predicted decrease in the quadratic model, at  $y_{M-1}$ , of the last block of constraints, restricted to the intersection of the null spaces of the Jacobians of all previous blocks.

When all the constraint blocks have been processed,  $n - m$  degrees of freedom still remain. The remaining variables are used in building a model of the objective function, so that the final substep,  $s_{M+1}$ , is obtained to produce sufficient predicted decrease in the quadratic model at  $y_M$  of the objective function, restricted to the intersection of the null spaces of the Jacobians of all constraint blocks. The final step is taken to yield the next major iterate, i.e., the next approximation to a solution of problem EQC. Thus, the total trial step from  $x_c$  to  $x_+$  is the sum of the substeps in the inner sweep, i.e.,  $\hat{s}_c = s_1 + \dots + s_{M+1}$ .

Unless the convergence criterion is met, the total trial step is evaluated, and the algorithm returns to process again the first block of constraints in a trust region determined by the success or failure of the trial step.

### 3.2.1 Computing the Substeps

During the constraint elimination stage, the substeps solve the following subproblems:

$$\begin{aligned} & \text{minimize } \frac{1}{2} \|C_k(y_{k-1}) + \nabla C_k(y_{k-1})^T s\|_2^2 \\ & \text{subject to } \nabla C_j(y_{j-1})^T s = 0, j = 1, \dots, k-1, \\ & \text{and possibly an additional constraint} \\ & \text{on the step direction,} \\ & \text{and } \|s\|_2 \leq \delta_k, \end{aligned}$$

for  $k = 1, \dots, M$ . (Note that for  $k = 1$  there is no null space constraint.) Then the objective function subproblem is:

$$\begin{aligned} & \text{minimize } f(y_M) + \nabla f(y_M)^T s + \frac{1}{2} s^T H(y_M) s \\ & \text{subject to } \nabla C_j(y_{j-1})^T s = 0, j = 1, \dots, M, \\ & \text{and possibly an additional constraint} \\ & \text{on the step direction,} \\ & \text{and } \|s\|_2 \leq \delta_{M+1}. \end{aligned}$$

If there is no additional constraint on the direction of the step, the subproblems produce a trust-region generalization of the local Brent step. A constraint requiring that the step be parallel to some coordinate hyperplane would be a generalization of the local Brown step. In practice, there is no explicit constraint for the generalization of the Brown step; rather it is computed implicitly (see [3]).

Let  $Q_{k-1}$  be a matrix the columns of which form a basis for the intersection of the null spaces of  $\nabla C_1(y_0), \dots, \nabla C_{k-1}(y_{k-2})$ . A change of variables,  $v = Q_{k-1}s$ , converts the constrained subproblems to unconstrained ones.

For relatively small problems, the null space bases can be computed by using the QR decomposition to find the basis for null space of  $\nabla C_1(y_0)$ , and then by updating the decomposition for subsequent subproblems to find a basis for the null space intersections. For larger problems, the QR decomposition becomes prohibitively expensive. In that case, reduced basis projectors can be used. More details about the null space basis computations can be found, for example, in [3] and [20].

There are various methods for solving large-scale trust region subproblems. We are holding much hope for the method recently developed by D. Sorensen of Rice University.

However, once the subproblems with null space constraints are converted into unconstrained trust-region subproblems, the steps may be chosen in any manner, as long as they satisfy two mild conditions.

1. As mentioned earlier, if there are no additional constraints on the subproblem  $k$ , its solution, a Levenberg-Marquard step for the reduced problem, produces a generalization of the Brent

step. That is, the substep is orthogonal to the linearized constraint hyperplane, for all blocks numbered  $k + 1, \dots, M$ . However, we do not require that the substeps be orthogonal. We require that each substep satisfies

$$\|s_k\| \leq \Lambda_k \|C_k(y_{k-1})\| \quad (7)$$

for some positive constant  $\Lambda_k$  that depends only on the properties of that particular constraint block but is independent of the iteration. Other conditions are possible to assure global convergence. However, this condition, first formalized in [13], is reasonable in that it is enforced automatically by any algorithms of interest for computing linearly feasible points. For instance, this is easily shown for the extensions of both Brown and Brent steps (see [3]).

2. We also require for each substep to satisfy a fraction of Cauchy decrease condition for the particular subproblem that substep solves. This is also a very mild condition—it is satisfied by all reasonable methods. Note that we do not place any conditions on the total trial step—only on the substeps.

It is easy to show that if  $s_k^{B-B}$  is an unconstrained Brown or Brent substep (or any substep out of the local Brown-Brent class), we can claim the following:

*If  $\|s_k^{B-B}\| \leq \delta_k$ , then let  $s_k = s_k^{B-B}$ . Otherwise let*

$$s_k = \frac{\delta_k * s_k^{B-B}}{\|s_k^{B-B}\|}. \quad (8)$$

*Then  $s_k$  satisfies the fraction of Cauchy decrease condition on subproblem  $k$ . The proof is given in [3].*

Thus, we see that simply truncating the unconstrained Brown or Brent substep to the size of the trust region will produce sufficient predicted decrease in the models of the constraint blocks.

### 3.2.2 The Merit Function and Its Model

Merit functions used to evaluate the progress of single-block trust-region algorithms consist of some combination of the objective function and the constraints. One common merit function is the  $\ell_2$  penalty function  $f(x) + \rho \|C(x)\|_2^2$ , where  $\rho$  is the penalty parameter.

In the process of the multilevel algorithm development, it has become apparent that conventional merit functions are inadequate for measuring progress of the multilevel methods, because a conventional merit function does not take into account the order in which minimization proceeds.

The difficulty can be summarized as follows:

- The result of the  $k$ -th minimization subproblem predicts decrease for the  $k$ -th component from point  $y_{k-1}$  to point  $y_k$ . It predicts no change for all previous blocks. However, there is no prediction at all about how  $s_1 + \dots + s_k$  changes and likely increases the norms of the blocks numbered  $k + 1, \dots, M$ . Neither does any substep, except  $s_{M+1}$  predict the behavior of the objective function.

This observation brought us to the conclusion that the merit function must take into account the multilevel structure of the scheme. Consider the following modified  $\ell_2$  penalty function:

$$\begin{aligned}\bar{\mathcal{P}}(x; \rho_1, \dots, \rho_M) &= f(x) + \rho_M(\|C_M(x)\|^2 \\ &\quad \rho_{M-1}(\|C_{M-1}(x)\|^2 + \rho_{M-2}(\|C_{M-2}(x)\|^2 + \dots + \rho_2(\|C_2(x)\|^2 + \rho_1\|C_1(x)\|^2)))) \\ &= f(x) + \sum_{k=1}^M \left( \prod_{j=k}^M \rho_j \right) \|C_k(x)\|^2,\end{aligned}$$

where  $\rho_k \geq 1$ ,  $k = 1, \dots, M$ . where  $\rho_k \geq 1$ ,  $k = 1, \dots, M$ . The initial choice  $\rho_k = 1$  is arbitrary and scale-dependent. The only requirement is that  $\rho_k \geq 1$ . For theoretical purposes, the problem is assumed to be well-scaled.

The new merit function penalizes for the possible predicted increase in the constraint blocks  $k, \dots, M$ , or in the objective function that may have occurred during inner loop iterations  $1, \dots, k-1$ .

At  $y_{M+1} = x_+ = x_c + \hat{s}_c$ , we model each  $\|C_k(x_+)\|^2$  by  $\|C_k(y_{k-1}) + \nabla C_k(y_{k-1})s_k\|^2$ , and so we model the merit function at  $x_+$  by

$$\begin{aligned}\mathcal{M}_c(s_1, \dots, s_{M+1}; \rho_1^c, \dots, \rho_M^c) &= f(y_M) + \nabla f(y_M)^T s_{M+1} + \frac{1}{2} s_{M+1}^T H(y_M) s_{M+1} \\ &\quad + \|C_M(y_{M-1}) + \nabla C_M(y_{M-1})^T s_M\|^2 + \rho_{M-1}^c (\|C_{M-1}(y_{M-2}) + \nabla C_{M-1}(y_{M-2})^T s_{M-1}\|^2 \\ &\quad + \rho_{M-2}^c (\|C_{M-2}(y_{M-3}) + \nabla C_{M-2}(y_{M-3})^T s_{M-2}\|^2 + \dots \\ &\quad + \rho_2^c (\|C_2(y_1) + \nabla C_2(y_1)^T s_2\|^2 + \rho_1^c \|C_1(y_0) + \nabla C_1(y_0)^T s_1\|^2))) \\ &= f(y_M) + \nabla f(y_M)^T s_{M+1} + \frac{1}{2} s_{M+1}^T H(y_M) s_{M+1} \\ &\quad + \sum_{k=1}^M \left( \prod_{j=k}^M \rho_j \right) \|C_k(y_{k-1}) + \nabla C_k(y_{k-1})^T s_k\|^2.\end{aligned}$$

We define the actual reduction as the difference between the merit function values at  $x_c$  and  $x_+$ , and we define the predicted reduction as the difference between the value of the merit function at  $x_c$  and the value of the model at  $x_+$ .

### 3.2.3 Updating the Penalty Parameters

This penalty parameter updating scheme for multilevel methods generalizes the scheme proposed in El-Alem [15], [16]. It ensures that our merit function has an essential property, namely, that unless an iterate is optimal, the predicted reduction should always be positive. We use the following procedure:

**Algorithm 3.1** Penalty Parameter Updating Algorithm (*Done on completion of each inner sweep of minimization problems.*)

Denote the set  $\{s_1, \dots, s_k\}$  by  $S_k$  and denote the set  $\{\rho_1, \dots, \rho_k\}$  by  $\rho_k$ .

At the beginning of a multilevel algorithm, set  $\rho_1^- = \dots = \rho_M^- = 1$  and choose  $\beta \in (0, 1)$ .

1. Compute  $C_{pred1}(s_1) = \|C_1(y_0)\|^2 - \|C_1(y_0) + \nabla C_1(y_0)^T s_1\|^2$ .
2. **Do**  $k = 1, M$

Update  $\rho_k$ .

Compute

$$Cpred_{k+1}(S_{k+1}; \rho_{k-1}^c, \rho_k^-) = \\ \frac{1}{2} [\|C_{k+1}(y_0)\|^2 - \|C_{k+1}(y_k) + \nabla C_{k+1}(y_k)^T s_{k+1}\|^2] \\ + \rho_k^- Cpred_k(S_k; \rho_{k-1}^c).$$

if  $Cpred_{k+1}(S_{k+1}; \rho_{k-1}^c, \rho_k^-) \geq$

$\frac{\rho_k^-}{2} Cpred_k(S_k; \rho_{k-1}^c)$  then

$$\rho_k^c = \rho_k^-.$$

$$Cpred_{k+1}(S_{k+1}; \rho_{k-1}^c, \rho_k^c) = \\ Cpred_{k+1}(S_{k+1}; \rho_{k-1}^c, \rho_k^-).$$

else

$$\rho_k^c = \rho_k^- + \beta,$$

$$\text{where } \rho_k^- = \frac{2[\|C_{k+1}(y_k) + \nabla C_{k+1}(y_k)^T s_{k+1}\|^2 - \|C_{k+1}(y_0)\|^2]}{Cpred_k(S_k; \rho_{k-1}^c)}.$$

Compute  $Cpred_{k+1}(S_{k+1}; \rho_{k-1}^c, \rho_k^c)$ .

end if

end Do

3. Update  $\rho_M$ .

Compute

$$pred(S_M; \rho_{M-2}^c, \rho_{M-1}^-) = \\ [f(y_0) - \phi_M(s_M)] + \rho_{M-1}^- Cpred_M(S_M; \rho_{M-1}^c).$$

if  $pred(S_M; \rho_{M-2}^c, \rho_{M-1}^-) \geq$

$\frac{\rho_{M-1}^-}{2} Cpred_M(S_M; \rho_{M-1}^c)$  then

$$\rho_M^c = \rho_{M-1}^-.$$

$$pred(S_M; \rho_{M-2}^c, \rho_{M-1}^c) = pred(S_M; \rho_{M-2}^c, \rho_{M-1}^-).$$

else

$$\rho_M^c = \rho_{M-1}^- + \beta,$$

$$\text{where } \rho_{M-1}^- = \frac{2[\phi_M(s_M) - f(y_0)]}{Cpred_M(S_M; \rho_{M-1}^c)}.$$

Compute  $pred(S_M; \rho_{M-2}^c, \rho_{M-1}^c)$ .

end if

End

Note that without updating the penalty parameters we can be assured of the positive predicted reduction from  $x_c$  only for the first block of constraints, i.e., only  $Cpred_1(s_1)$  is definitely positive without additional considerations. To ensure that  $Cpred_2(s_1, s_2; \rho_1)$  is positive, we may have to increase  $\rho_1$ . Now that  $Cpred_2(s_1, s_2; \rho_1)$  is positive, we use it to ensure that the next partial predicted reduction is positive, and so on. So, for each each substep  $s_k$ , the predicted reduction accumulated by the step  $s_1 + \dots + s_k$  is at least a fraction of the predicted decrease accumulated by the step  $s_1 + \dots + s_{k-1}$ .

Thus the predicted reduction of the first block is the most heavily penalized one.

It should be emphasized that the step computation is completely independent of the penalty parameter computation.

### 3.2.4 Step Evaluation and Trust-Region Radii Updating

Although there are various schemes of evaluating the trial step and updating the trust region radii, for the sake of simplicity in discussion, we adopt the following strategy:

- The total trial step is evaluated outside the inner loop.
- All individual trust region radii are equal and are updated simultaneously by the same factor.

Other strategies for practical implementations are discussed in [1]. We would like to emphasize that the simultaneous expansion or contraction of the trust region radii is not a technical requirement.

The algorithm for evaluating the step and updating the trust region radii follows.

#### Algorithm 3.2 Step Evaluation / Trust Region Update

Given  $\delta_k > 0, k = 1, \dots, M$  (or  $k = 1, \dots, M + 1$  for optimization),  $\delta_{max} > 0, \delta_{min} > 0, 0 < \eta_1 < \eta_2 < 1, \alpha_1 \in (0, 1], \alpha_2 > 1, x_c \in \mathbb{R}^n, a_{red}, p_{red}$ ,

```

Compute  $r = \frac{a_{red}}{p_{red}}$ .
if  $r < \eta_1$  then (step not accepted)
     $\delta_k = \alpha_1 * \delta_k$ .
else if  $r \geq \eta_2$  then (step accepted)
     $\delta_k = \min\{\delta_{max}, \max\{\delta_{min}, \alpha_2 * \delta_k\}\}$ .
     $x_c = x_+$ .
else ( step accepted)
     $\delta_k = \max\{\delta_{min}, \delta_k\}$ .
     $x_c = x_+$ .
end if

```

We note that if the step is not accepted, the trust region radii are decreased without any safeguard. However, if the step is accepted, the next trust region radius is set to be no smaller than a predetermined positive value  $\delta_{min}$ . This strategy is extremely important in the global convergence theory. It ensures that the trust region radius is bounded away from zero and hence that the penalty parameters are bounded from above. This technique was introduced in [22].

### 3.2.5 The Stopping Criteria

We use the first order necessary conditions for problem EQC to terminate the algorithm and require that

$$\begin{aligned}
 \|C_1(y_0)\| &\leq \epsilon_{tol} \\
 \|C_2(y_1)\| &\leq \epsilon_{tol} \\
 &\dots \\
 \|C_M(y_{M-1})\| &\leq \epsilon_{tol} \\
 \|P_M^T \nabla f(y_M)\| &\leq \epsilon_{tol}
 \end{aligned} \tag{9}$$

hold simultaneously.

Since

$$\|s_k\| = O(\|C_k(y_{k-1})\|),$$

if  $\|C_k(y_{k-1})\|$  is small,  $\|s_k\|$  will be small and the inner loop iterates  $y_k$  will be close to each other, and in the limit we can show ([2], [3]) that at least a subsequence of the generated sequence of the outer loop iterates will converge to a stationary point of problem EQC.

The tolerance parameters  $\epsilon_{tol}$  need not be the same, but for convenience, they are taken to be the same throughout the convergence analysis.

The reason for requiring such a stopping criterion is theoretical and practical. The conventional test for the entire norm of the constraint residual being close to 0 does not differentiate between the individual  $\|C_k(y_{k-1})\|$ . It is essential for the convergence proof to determine how close to feasibility an iterate must be in order for the penalty parameters not to be increased. This is a measure of feasibility versus optimality. The conventional stopping criterion allows only the total feasibility to be measured and thus to determine when  $\rho_M$  does not have to be increased. But even if  $\rho_M$  is not increased,  $\rho_1, \dots, \rho_{M-1}$  may have to be increased because of the relative sizes of the component block norms. The conventional criterion does not allow us to measure **relative feasibility** of one block of constraints with respect to the others.

In practice, we do not wish to evaluate the residuals at the same point just for the sake of the stopping criterion.

Other stopping criteria are possible, but the one above is the most natural one.

### 3.2.6 The Statement of the Algorithm

The formal description of the algorithm follows.

Let the constraints be partitioned into  $M$  blocks.

#### Algorithm 3.3 Multilevel Algorithm for Equality Constrained Optimization

Given  $\delta_k > 0, k = 1, \dots, M, \delta_{max} > 0, \delta_{min} > 0, 0 < \eta_1 < \eta_2 < 1, \alpha_1 \in (0, 1], \alpha_2 > 1, x_c \in \mathbb{R}^n$ .

**Outer Loop:** Do until convergence:

$y_0 = x_c$ .

**Compute the trial step.**

**Inner Loop:** Do  $k = 1, M$

If  $y_{k-1}$  is not feasible then

    Compute  $s_k$  that satisfies a fraction of Cauchy decrease

    condition on  $\frac{1}{2}\|C_k(y_{k-1}) + \nabla C_k(y_{k-1})s\|_2^2$  restricted to

    the intersection of the null spaces of  $\nabla C_j(y_{j-1})^T s = 0, j = 1, \dots, k-1,$

    and  $\|s_k\| \leq \Lambda_k \|C_k(y_{k-1})\|$  (satisfied automatically).

$y_k = y_{k-1} + s_k$ .

End if

**End Inner Loop**

Compute  $s_{M+1}$  to satisfy the fraction of Cauchy decrease

condition on the subproblem: minimize  $\phi_M(s_M)$  restricted to



the intersection of the null spaces of  $J_j(y_{j-1})s = 0, j = 1, \dots, M$ ,  
and  $\|s\|_2 \leq \delta_M$ .

$$y_{M+1} = y_M + s_{M+1}.$$

$$x_+ = y_{M+1}.$$

**The trial step is:**  $\hat{s}_c = s_1 + \dots + s_{M+1}$ .

**Update the penalty parameters**

**Evaluate the step and update the trust region radius**

If the step is accepted, set  $x_c = x_+$ .

**End Outer Loop**

We should note that there is an option to eliminate only a subset of constraints via the described procedure. In this case, the rest of the constraints and the objective function would be restricted to the intersection of the null spaces of the Jacobians of the processed constraints, and the resulting reduced optimization problem would be solved by a chosen method. The discussion of this approach is left for later work.

## 4 Global Convergence Results

In this section we give a summary of the global convergence theory for multilevel algorithms.

### 4.1 Basic Ingredients of a Global Convergence Proof

Our proof contains the general ingredients of a global convergence analysis for a trust-region method. The first three are requires for a typical analysis of an unconstrained minimization algorithm.

1. The trial step must be shown to satisfy a sufficient predicted decrease condition, usually the FCD condition. Our algorithm assumes that the substeps satisfy the FCD condition on the subproblems. It remains for us to show that the total step from  $x_c$  to  $x_+$  satisfies a suitable decrease condition.
2. The difference between the actual and predicted reduction must be bounded above by at least a constant multiple of the square of the total step norm plus multiples of higher powers of the step norm. This is easily shown multilevel algorithms.
3. The algorithm must be shown to be well-defined, i.e., we must prove that the ratio of the actual reduction to predicted reduction can be made greater than a given  $\eta_1 \in (0, 1)$  after a finite number of trial step computations. Given 2, it is easy to show that as the trust region radius approaches zero, the ratio of the actual reduction to predicted reduction approaches one. For the algorithm to be well-defined we must show that the ratio of the predicted to actual reduction approaches one faster than the trust region radius goes to zero. This is easily established for our algorithm.

An algorithm for constrained optimization that uses penalty parameters in its merit function requires the fourth ingredient.

4. The penalty parameter in the merit function must be shown to be bounded. The technique is to prove that the product of the penalty parameter and the trust region radius is bounded by a constant independent of the iterates. The sequence of the trust region radii is then shown to be bounded away from zero. Here a crucial role is played by the trust region updating technique introduced in [22]: after a successful iteration and before starting the next iteration, the trust region radius is set to be no smaller than a pre-defined value. This way of updating allows us to prove that the sequence of penalty parameters is bounded from above.

The method for updating the penalty parameters ensures that the sequence of penalty parameters is nondecreasing \*, which, together with its boundedness, allows us to conclude that the penalty parameter sequence converges and, moreover, remains constant after a finite number of increases. This fact is used in the global convergence theorem.

## 4.2 Assumptions

We make the following assumptions on the problem and the sequence of steps and iterates:

- $f, C$  are at least twice continuously differentiable.
- The gradient of the constraints has full rank. This is a strong assumption, but it is a standard practice to require it for the sake of convergence proofs. Practical experience suggests that the breakdown of this assumption does not necessarily diminish the efficacy of our algorithm. Not assuming full rank would allow us to prove a slightly weaker convergence result.
- $f(x), \nabla f(x), \nabla^2 f(x), H_M, C(x), \nabla C(x), \nabla C_k(x), \nabla^2 C_j(x), j = 1, \dots, m,$   
 $\{[P_{k-1}^T \nabla C_k(x)]^T [P_{k-1}^T \nabla C_k(x)]\}^{-1}, k = 1, \dots, M,$  are all uniformly bounded in norm for all  $x$  in the domain of interest.

Since we require that the Hessian of the objective function be only bounded, we can even take it to be 0. Of course, such an approximation would lower the effectiveness of the algorithm.

## 4.3 Summary of the Proof

In this subsection we provide an overview of steps in the convergence proof. The details can be found in [3].

- We show that under our assumptions, the norm of any intermediate sum of the substeps is bounded by a constant times the norm of the total trial step.
- Several technical results provide workable expressions of the FCD (fraction of Cauchy decrease) condition similar to the one used for unconstrained optimization.
- A standard result provides an upper bound on the error between actual reduction and predicted reduction.

---

\*The global convergence theory for algorithms with nonmonotone penalty parameters has been investigated by Mahmoud El-Alem [17].

- By virtue of the penalty parameter updating scheme, the multilevel algorithms have the property that if an iterate is feasible, the penalty parameters are not increased. We show that if the iterates are sufficiently close to feasibility, the penalty parameters are not increased either. This result is crucial to the proof of convergence, giving a sufficient condition for the penalty parameters not to be increased.
- Next we establish an upper bound on the product of the penalty parameters with the trust region radii. This result allows us to conclude that the radii are bounded below if the penalty parameters increase. The penalty parameter sequences are shown to be nondecreasing, which, together with their boundedness from above, allows us to conclude that the penalty parameters tend to a limit, and, moreover, stay constant after a finite number of outer iterations. The limit is shown to exist, but its explicit expression is not known.
- We have shown that the total trust region radius is bounded away from zero if any of the penalty parameters are increased. Now we show that radius is always bounded away from zero. The trust region updating strategy ensures that it is bounded from above.
- The next result guarantees that the algorithm is well defined, i.e., that after a finite number of outer loop iterations an acceptable step  $\hat{s}_c$  with

$$\frac{ared}{pred} \geq \eta_1$$

will be found.

- In the global convergence result, we show that if the objective function is bounded below, then the sequence of iterates generated by a multilevel algorithm has a subsequence convergent to a stationary point of the equality constrained minimization problem.
- As a corollary, we can now conclude that the multilevel algorithm for nonlinear equations is also globally convergent.

## 5 Discussion and Concluding Remarks

We have described a broad new class of multilevel algorithms for solving the nonlinear equations problem and the equality constrained optimization problem. The class can be considered as a globalization and an extension of the local class of algorithms of Brown and Brent for solving nonlinear systems of equations.

The main practical appeal of the multilevel algorithms is that in the case of equality constrained optimization, they allow the user to partition the constraint system arbitrarily, to fit the application, and to process the blocks of constraints separately. In their finite-difference derivative form, they require fewer function evaluations than the Newton's method.

The multilevel class is characterized by requiring very mild conditions to be imposed on the trial steps. All reasonable algorithms satisfy these conditions automatically.

We have established global convergence theory for the entire class. The theory implies convergence of the nonlinear equations solver, which, to the author's knowledge, is the first theoretically supported method for globalizing Brown-Brent methods. The global convergence theory was made

possible by the introduction of the new merit function that takes into account the order of the constraint processing. The nested penalty parameters are updated by an extension of the scheme proposed by El-Alem [15].

The algorithms are expected to be applicable to the problem of the multidisciplinary design optimization and to serve as a foundation for the study of the general multilevel optimization problem.

We would like to mention one more application. The design of complex engineering systems is by nature a multicriteria optimization problem. The design projects are distinguished by very large numbers of variables, constraints, and expensive analyses. To solve the problem, it is necessary to break it into disciplines, each of which produces its own optimal design. The discipline designs are then incorporated into a total design. The multilevel methods proposed here would allow researchers to integrate constraints obtained from different sources.

To solve the multicriteria optimization problem, it is necessary to decide when an iterate is optimal. One of the approaches to optimality is the statement of the multicriteria problem as a multilevel optimization problem, i.e., the problem of minimizing a function on a feasible set, which is an optimal set for another function, and so on. In such an approach, the user places priorities on the optimization problems that are to be solved sequentially. We believe that the multilevel algorithms proposed here will serve as a beginning for a detailed study of the general multilevel optimization problem.

Directions of research in progress include local convergence rates, implementation, extensive testing on applications, incorporation of bound and inequality constraints, and extensions to general nonlinear bilevel and multilevel optimization.

## References

- [1] Natalia Alexandrov. On implementation of multilevel algorithms for nonlinear equations and equality constrained optimization. In progress.
- [2] Natalia Alexandrov. *Multilevel Algorithms for Nonlinear Equations and Equality Constrained Optimization*. PhD thesis, Rice University, Department of Computational and Applied Mathematics, May 1993. Technical Report TR93-20.
- [3] Natalia Alexandrov and J. E. Dennis, Jr. A class of general trust-region multilevel algorithms for systems of nonlinear equations: Global convergence theory. In preparation.
- [4] M. Avriel. *Nonlinear Programming*. Prentice-Hall, 1976.
- [5] R. J. Balling and J. Sobieszczanski-Sobieski. Optimization of coupled systems optimization: A critical overview of approaches. In *Proceedings of the 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-94-4330, Panama City, Florida, September 7-9 1994. In preparation.
- [6] Richard P. Brent. Some efficient algorithms for solving systems of nonlinear equations. *SIAM J. Numerical Analysis*, 10(2):327—344, April 1973.

- [7] Kenneth M. Brown. *A Quadratically Convergent Method for Solving Simultaneous Nonlinear Equations*. PhD thesis, Purdue University, 1966.
- [8] Kenneth M. Brown. Algorithm 316. Solution of simultaneous nonlinear equations. *Communications ACM*, 10:728—729, 1967.
- [9] Kenneth M. Brown. A quadratically convergent Newton-like method based on Gaussian elimination. *SIAM J. Numerical Analysis*, 6(4):560—569, December 1969.
- [10] Kenneth M. Brown and J. E. Dennis, Jr. On the second order convergence of Brown's derivative-free method for solving simultaneous nonlinear equations. Research Report 71-7, Yale University, Department of Computer Science, New Haven, Connecticut, June 1971.
- [11] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem formulation for multidisciplinary optimization. Technical Report 93-34, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251-1892, 1993.
- [12] J. E. Dennis, Jr. Non-linear least squares and equations. In *The State-of-the-Art in Numerical Analysis*, 1976.
- [13] J. E. Dennis, Jr., Mahmoud El-Alem, and Maria Cristina Maciel. A global convergence theory for general trust-region-based algorithms for equality constrained optimization. Technical Report TR92-28, Rice University, Department of Computational and Applied Mathematics, Houston, Texas 77251, September 1992.
- [14] J. E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jersey, 1983.
- [15] M. M. El-Alem. *A Global Convergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*. PhD thesis, Rice University, Department of Mathematical Sciences, 1988.
- [16] M. M. El-Alem. A global convergence theory for the Celis-Dennis-Tapia trust region algorithm for constrained optimization. *SIAM J. Numerical Analysis*, 28:266—290, 1991.
- [17] M. M. El-Alem. Robust trust-region algorithm with non-monotonic penalty parameter scheme for constrained optimization. Technical Report TR92-30, Rice University, Department of Computational and Applied Mathematics, Houston, Texas 77251, September 1992.
- [18] R. Fletcher. *Practical Methods of Optimization*. Wiley, 1989.
- [19] David M. Gay. *Brown's Method and Some Generalizations, with Applications to Minimization Problems*. PhD thesis, Cornell University, 1975.
- [20] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, 1981.
- [21] Raphael T. Haftka and Zafer Gürdal. *Elements of Structural Optimization*. Academic Press, Kluwer Academic Publishers, 1993. Third revised and expanded edition.

- [22] M. El Hallabi and R. A. Tapia. A global convergence theory for arbitrary norm trust-region methods for nonlinear equations. Technical Report TR87-25, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251-1892, 1987.
- [23] L. S. Lasdon and A. D. Waren. Generalized reduced gradient software for linearly and nonlinearly constrained problems. In H. J. Greenberg, editor, *Design and Implementation of Optimization Software*, pages 335—362. Sijthoff and Noordhoff, Netherlands, 1978.
- [24] María Cristina Maciel. *A Global Convergence Theory for a General Class of Trust Region Algorithms for Equality Constrained Optimization*. PhD thesis, Rice University, Department of Computational and Applied Mathematics, 1993.
- [25] José Mario Martínez. Generalization of the methods of Brent and Brown for solving nonlinear simultaneous equations. *SIAM J. Numerical Analysis*, 1(3):434—448, June 1979.
- [26] José Mario Martínez. Solving nonlinear simultaneous equations with a generalization of Brent's method. *BIT*, 20:501—510, 1980.
- [27] J. J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: The State of the Art*. Springer-Verlag, 1983.
- [28] J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM J. Sci. Stat. Comput.*, 4(3):553—572, September 1983.
- [29] M. J. D. Powell. Convergence properties of a class of minimization algorithms. In O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, editors, *Nonlinear Programming 2*. Academic Press, 1975.
- [30] 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 J. Numerical Analysis*, 22(1):47—67, February 1985.
- [31] Jaroslaw Sobieszczanski-Sobieski, Benjamin James, and Augustine Dovi. Structural optimization by multilevel decomposition. In *Proceedings of the AIAA/ASME/ASCE/AHS 24th Structures, Structural Dynamics and Materials Conference*, Lake Tahoe, Nevada, May 2-4 1983. AIAA Paper No. 83-0832-CP.
- [32] D. C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numerical Analysis*, 19(2):409—426, April 1982.
- [33] G. W. Stewart. Conjugate direction methods for solving systems of linear equations. *Numer. Math.*, 21:285-297, 1973.