A TRUST REGION FRAMEWORK FOR MANAGING THE USE OF APPROXIMATION MODELS IN OPTIMIZATION [*]

NATALIA ALEXANDROV [†], J. E. DENNIS, JR. [‡], ROBERT MICHAEL LEWIS [§], AND VIRGINIA TORCZON [¶]

Abstract. This paper presents an analytically robust, globally convergent approach to managing the use of approximation models of various fidelity in optimization. By robust global behavior we mean the mathematical assurance that the iterates produced by the optimization algorithm, started at an arbitrary initial iterate, will converge to a stationary point or local optimizer for the original problem. The approach we present is based on the trust region idea from nonlinear programming and is shown to be provably convergent to a solution of the original high-fidelity problem. The proposed method for managing approximations in engineering optimization suggests ways to decide when the fidelity, and thus the cost, of the approximations might be fruitfully increased or decreased in the course of the optimization iterations. The approach is quite general. We make no assumptions on the structure of the original problem, in particular, no assumptions of convexity and separability, and place only mild requirements on the approximations. The approximations used in the framework can be of any nature appropriate to an application; for instance, they can be represented by analyses, simulations, or simple algebraic models. This paper introduces the approach and outlines the convergence analysis.

1. Introduction.

In this paper we present an approach to managing the use of approximation models in optimization that is based on the trust region approach from nonlinear programming [8, 17]. The approach we present inherits the mathematical robustness and global and local convergence properties of the classical trust region methods. By global convergence we mean the assurance that the iterates produced by an optimization algorithm working with the approximation models, started at an arbitrary initial iterate, will converge to a stationary point or local optimizer for the original problem. The local convergence rate determines the asymptotic efficiency of the method. The approach we present also suggests criteria to decide when the fidelity (and thus the cost) of the approximations might be fruitfully increased or decreased.

The use of approximations in engineering optimization motivates this work. A review of approximation models in structural optimization can be found in [2]. When many of these ideas were first formalized, for instance, in [22] and [24], the idea was to employ approximation models in conjunction with existing mathematical programming techniques to solve structural design optimization problems. However, to the best of our knowledge, prior analysis in the structural optimization community has focussed on the question of whether or not the optimization technique would converge to a solution of the problem defined by the approximation concept, rather than the

[*] Revision: 1.2 Date: 1998/01/13 22:20:41

[†] Multidisciplinary Optimization Branch, Mail Stop 159, NASA Langley Research Center, Hampton, Virginia 23681

[‡] Computational and Applied Mathematics, Rice University. Currently on leave at the Boeing Company, Seattle, Washington. Leave supported by the Dept. of Energy grant DE-FG03-95ER25257 and Air Force Office of Scientific Research grant F49620–95–1–0210.

[§] ICASE, Mail Stop 403, NASA Langley Research Center, Hampton, Virginia 23681. This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681.

[¶] Department of Computer Science, College of William & Mary, Williamsburg, Virginia 23187. This research was supported by the Air Force Office of Scientific Research grant F49620–95–1–0210 and by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

original problem (e.g., [3, 15], with an exception being [1, 18]). Because the method we propose inherits the convergence properties of the classical trust region algorithms for nonlinear optimization, we can give simple conditions which assure that the iterates produced by using suitable approximation models, starting from an arbitrary initial iterate, will converge to a stationary point or a local optimizer of the original problem. The analysis easily accommodates varying the nature of the approximation from iteration to iteration.

The trust region framework gives an adaptive method for managing the amount of optimization done with the approximation models before one has recourse to a detailed model to check the validity of the design generated by the approximation model. This regulation is based on the ability of the approximation to predict improvement in the system being optimized. Moreover, by comparing the improvement predicted by the approximation model to the improvement realized for the true system being optimized, we obtain useful information on how well the model is predicting the behavior of the system. This information can be used to suggest when a model of greater or lesser fidelity may be more suitable as well as when more or less optimization might be done on the model before the next comparison.

In this paper we consider only the case of unconstrained minimization. We do this, in part, for simplicity in presenting the trust region approach. A discussion of trust region approaches to constrained optimization, particularly the convergence theory for constrained algorithms, would require the introduction of technical machinery that would obscure the points we wish to make. Moreover, many nonlinear programming algorithms for constrained optimization—penalty methods, classical and modified barrier methods, augmented Lagrangian methods—actually proceed by solving a sequence of unconstrained optimization problems, to which the current discussion applies. The case of constrained optimization we will treat in detail elsewhere.

Let $x = (x^1, \cdots, x^n)$ denote the design variables, and suppose that one has a model of high physical fidelity but high computational cost, as well as an approximate model of lower physical fidelity but lower computational cost. Let the associated performance measures (merit/cost/objective functions) be $f(x)$ and $a(x)$ and their sensitivities (with respect to the design variables) be $\nabla f(x)$ and $\nabla a(x)$:

$$\begin{array}{lcl} x \rightarrow \boxed{\text{High-fidelity (“true”) model}} \rightarrow & & f(x),\ \nabla f(x) \\ x \qquad \rightarrow \boxed{\text{Approximate model}} \rightarrow & & a(x),\ \nabla a(x). \end{array}$$

Fig.1.1 describes a conceptual scheme for using approximation models in the context of optimization. One occasionally uses information from the high-fidelity model to check designs generated using a model of lower fidelity but of lower computational cost. One then takes a number of optimization iterations using this simpler, cheaper approximation model. At the end of this optimization phase, one has recourse to the high-fidelity model to recalibrate the lower-fidelity model and then continues optimization using the simplified model.

In order to make such a scheme robust—that is, to be assured that we are converging to a design that is likely to yield at least a local optimum for the original, high-fidelity problem—we must address the following questions.

- What does one do when the design derived from optimization using the approximation model a fails to produce improvement in the true objective f?
- More generally, how can one use information about the predictive value of a (or lack thereof) to regulate the amount of optimization done using a before recourse to the high-fidelity model?

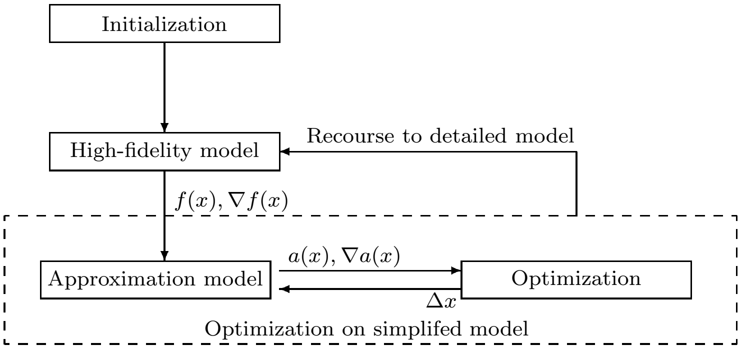

FIG. 1.1. *Conceptual optimization algorithm using approximation models.*

In addition, for reasons of efficiency, one also seeks guidance in regards to the following:

- When might it be appropriate to either change or refine the model to improve the progress of the optimization? When can the quality (and, presumably, cost) of the approximation model be reduced?

The trust region mechanism gives a systematic response to both poor and incorrect prediction on the part of the approximation model while not being so conservative as to retard progress when the approximation model does a good job of predicting improvement in the high-fidelity model. Furthermore, the trust region mechanism gives us a measure of how well the model is predicting improvement in the system and thus suggests criteria for changing or updating the model based on this measure of the predictive abilities of the model.

The approach we present is a confluence of a theoretically and practically attractive nonlinear programming methodology with a widely used engineering practice. The exclusive focus in nonlinear optimization has been on the use of local models, almost always quadratic, with various approximations to first- and second-order sensitivities. Our contribution here is the observation that the trust region framework fits naturally with the idea of using non-quadratic approximations found in engineering optimization. We require the approximations only to satisfy a few mild conditions, discussed in §3. Mathematically, the trust region framework for managing approximations is a straightforward extension of the classical trust region theory.

In §2 we present the relevant features of classical trust region algorithms. In §3 we apply the classical trust region approach to manage the use of the more general approximations available in engineering applications. In §4 we discuss how the convergence theory for classical trust region algorithms remains valid for the new approach using general approximation models. In §5 we discuss how the information used in the course of the the classical trust region approach might be used to decide when it might be appropriate to either change or refine the model to improve the progress of the optimization. Examples of approximations that can be used in the trust region framework we present are discussed in the Appendix.

2. The classical trust region approach in optimization. One of the goals of modern nonlinear programming algorithms is robust global behavior. By robust global behavior we mean the mathematical assurance that the iterates produced by an optimization algorithm, started at an arbitrary initial iterate, will converge to a stationary point or local optimizer for the problem. This robustness is achieved by globalization strategies such as trust regions, line searches, and continuation methods.

The classical trust region idea is to regulate the length of the steps taken in an iterative optimization process based on how well the current quadratic Taylor series model of f is found to predict improvement in f. This leads to an adaptive method for adjusting the size of the steps taken based on how well the local quadratic models are predicting decrease in f.

At iteration k, one begins by building a quadratic model q_k of the objective function:

$$f(x_k + s) \approx q_k(x_k + s) = f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s. \tag{2.1}$$

We use here the notation s to denote the prospective *step* Δx in the design variables, g_k to denote an approximation of the gradient $\nabla f(x_k)$, and the term B_k to denote a model of the second derivatives (curvature) of f at x_k. The convergence analysis of trust region methods in the unconstrained case places very mild requirements on the information used to construct the approximation (2.1). The gradient approximation g can be of limited accuracy [5], while the B_k need only remain uniformly bounded in norm. Typically B_k is the Hessian of f, calculated analytically or via finite-differences, or a quasi-Newton Hessian built up using a secant update such as BFGS or DFP [8].

The trust region algorithm proceeds by building and minimizing quadratic models of the form (2.1). However, in general such a quadratic model is known to be a good approximation only in a neighborhood of x_k. Consequently, we restrict the step we take to a region in which we trust the quadratic model to approximate f well, whence the name "trust region." This is done by adding a constraint on the length of the step allowed, resulting in the trust region subproblem:

$$\begin{array}{ll} \text{minimize} & q_k(x_k + s) \\ \text{subject to} & \| \, s \, \| \leq \delta_k. \end{array} \tag{2.2}$$

Note that the the direction of the step may vary with δ_k. In practice, one need not solve the problem (2.2) exactly for the step s_k; there is a relaxed condition on how much decrease s_k must produce in the quadratic model q_k in order to insure robust behavior [26, 17]. We discuss this criterion, the Fraction of Cauchy Decrease condition, in §4.

Although in this algorithm we use an ℓ_2 trust region, that is, the bound on the step is expressed using the Euclidean norm, this assumes an even scaling of all the components of x. In practice, the variables are scaled to improve performance; this leads to a trust region of the form $\| \, \Lambda s \, \| \leq \delta_k$, where Λ is a symmetric positive definite matrix. Other choices of norm are possible to define the trust region as well. For instance, one can use the ℓ_∞ norm, which is more appropriate when solving problems with bounds on the design variables [11].

One then decides whether to accept the prospective step s:

$$x_{k+1} = \begin{cases} x_k + s & \text{if } f(x_k + s) < f(x_k) \\ x_k & \text{otherwise.} \end{cases} \tag{2.3}$$

The trust radius δ_k is similar in purpose to a move limit [30]. However, the two are distinguished by the way in which they are updated. Move limits are set and updated in a manner based on the intuition of the user. From the point of view of mathematical analysis this is ad hoc. While the use of move limits in this way can be successful, to the authors' knowledge no proof of convergence for optimization algorithms that use move limits exists. On the other hand, in trust region algorithms, the "move limit"—the trust radius—is expanded and contracted in a systematic way for which one can prove global and local convergence results.

In particular, after each optimization iteration, the trust radius is updated in an adaptive way based on the predictive quality of the quadratic model used to generate steps, according to the following principles:

1. If the model did a very good job of predicting the actual improvement of f or if there was even more improvement than predicted in f, then increase δ_k and allow a longer step at the next optimization iteration $k+1$, since the model has proven its utility in finding improvement in f over the current trust region.
2. However, if the model did a bad job of predicting the improvement in f, either because f actually increased with the step s, or because f did decrease but not nearly as much as predicted by the quadratic model, decrease the size of the trust region used in the next optimization iteration. Calculus assures us that the quadratic model is good if we remain sufficiently close to x_k.
3. Finally, if the model did an acceptable but not especially noteworthy job of predicting the improvement in f, leave the size of the trust region alone.

Numerically, one chooses positive constants $r_1 < r_2 < 1$ and $c_1 < 1$, $c_2 > 1$ that regulate the expansion and contraction of the trust region. One compares the actual and predicted decrease,

$$r = \frac{f(x_k) - f(x_k + s)}{f(x_k) - q_k(x_k + s)},$$

and updates the trust radius as follows:

$$\delta_{k+1} = \begin{cases} c_1 \parallel s \parallel & \text{if } r < r_1 \\ \min\{c_2 \parallel \delta_k \parallel, \Delta^*\} & \text{if } r > r_2 \\ \parallel s \parallel & \text{otherwise,} \end{cases} \tag{2.4}$$

where Δ^* is an upper bound on the trust radius. Typical values for r_1 and r_2 are $r_1 = 0.10$ and $r_2 = 0.75$ [8].

Note that we do not reduce the trust region if the quadratic model under-predicts improvement in f. In the context of optimization, we focus on predicting descent, as opposed to the more general question of the overall quality of the approximation. The latter issue is important, however, in constrained problems if one wishes to insure feasibility of the iterates produced by an optimization algorithm.

The classical trust region algorithm is summarized in Figure 2.1. We have omitted stopping criteria; for a discussion of stopping criteria for trust region methods, see [8, 12].

3. A trust region approach with generalized approximation models. In the engineering optimization literature, the quadratic model (2.1) based on the Taylor series for f is sometimes called a *formal approximation*. Alternative models used in engineering practice can produce approximations that are better than the quadratic model over a larger neighborhood. These approximations are usually based on some knowledge of the problem and thus are specific to the application whereas quadratic models are always locally applicable. Many such problem-specific approximations can be found in the structural optimization literature; we discuss some examples that satisfy the conditions of the trust region framework in the Appendix.

```
Choose $x_0 \in \mathbb{R}^n$ and $\delta_0 > 0$.
For $k = 0, 1, \ldots$ until convergence do {
    Find an approximate solution $s_k$ to the subproblem:
        minimize    $q_k(x_k + s)$
        subject to  $\| s \| \le \delta_k$.
    Compare the actual and predicted decrease:
        $r = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - q_k(x_k + s_k)}$.
    Update $x_k$ according to (2.3) and $\delta_k$ according to (2.4).
}
```

Fig. 2.1. *The classical trust region algorithm for unconstrained minimization.*

We place the following two requirements on the approximation model used at each optimization iteration:

$$a_k(x_k) = f(x_k) \tag{3.1}$$

$$\nabla a_k(x_k) = \nabla f(x_k). \tag{3.2}$$

If the model a_k and its first derivatives at x_k agree with those of the actual objective f, we call the approximation a *first-order model.* If, in addition,

$$\nabla^2 a_k(x_k) = \nabla^2 f(x_k), \tag{3.3}$$

we call the approximation a *second-order model*, though these are not the primary focus of this paper since it is typically not the case that second-order derivatives of f are available.

In the case of unconstrained minimization we actually can weaken the condition (3.2) and develop an approach based on inexact gradients along the lines of [5], but for simplicity we will not pursue that approach here. One can also develop an approach using *zero-order models* that satisfy only the condition (3.1) [7].

The conditions (3.1)–(3.2) are not especially restrictive if exact or approximate sensitivity information for f is available. In the Appendix we discuss examples of approximations that satisfy (3.1)–(3.2) by construction.

The conditions (3.1) and (3.2) guarantee that sufficiently close to x_k, the approximation a_k is a good model of f. It is then clear how the trust region approach provides a mechanism to regulate the use of a in an optimization iteration: if the approximation model a is not a good predictor of the improvement of f for a long step, we decrease δ and fall back on a region in which a is an increasingly good model of f. On the other hand, if a is doing a good job of approximating the behavior of f, then we do not decrease the length of the steps we take, and thereby avoid unnecessarily restricting the progress of the optimization.

The trust region algorithm for unconstrained minimization using general first-order approximation models is then given in Figure 3.1, again, omitting the stopping criteria. In §4 we explain precisely what we mean by finding an approximate solution s_k to the subproblem.

Note that the approximation need not be fixed across all iterations; it can vary with the iteration, so we denote the approximation by a_k rather than a. One can change the approximation model if necessary to reflect the current region of the design space (for the case of quadratic models, see [4]). Furthermore, though only a single level of approximation is described in Figure 3.1, nothing precludes additional

```
Choose $x_0 \in I\!R^n$, $\Delta_0 > 0$.
For $k = 0, 1, \ldots$ until convergence do {
    Choose $a_k$ that satisfies $a_k(x_k) = f(x_k)$ and $\nabla a_k(x_k) = \nabla f(x_k)$.
    Find an approximate solution $s_k$ (for example, as in Figure 4.1)
    to the subproblem:
        minimize     $a_k(x_k + s)$
        subject to   $\| s \| \leq \Delta_k$.
    Compare the actual and predicted decrease in $f$:
        $r = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - a_k(x_k + s_k)}$.
    Update $x_k$ according to (2.3) and $\Delta_k$ according to (2.4).
}
```

FIG. 3.1. *A trust region algorithm using general approximation models.*

levels of approximation in solving the subproblem. The framework we propose can accommodate many levels of approximation which vary in fidelity but also, presumably, in computational cost.

4. Robustness and convergence theory. Obviously, the practical performance of any algorithm along the lines of Figure 1.1 depends on the quality of the approximation models and their ability to predict the behavior of f. However, we can make some general mathematical statements about the analytical robustness of the trust region framework for using approximation models in optimization.

In the conceptual algorithm given in Figure 1.1, if an optimization iterate is unsuccessful, the user would respond by improving the fidelity of the model and/or "doing less optimization". The trust region approach focuses on the latter option, which is accomplished by reducing the trust radius. This incidentally has the effect of "improving the model" insofar as attention is restricted to smaller regions in which the approximation is increasingly better. This strategy enables us to establish the robust behavior of the approach under the reasonably mild matching conditions (3.1) and (3.2), as we now discuss.

The algorithm in Figure 3.1 is the classical trust region procedure for unconstrained minimization with the distinction that the trust region subproblem

$$\begin{array}{ll} \text{minimize} & a_k(x_k + s) \\ \text{subject to} & \| s \| \leq \Delta_k \end{array} \tag{4.1}$$

involves a general first-order approximation model a_k instead of the conventional quadratic model q_k given by (2.1).

The trust region algorithm for general approximation models in Figure 3.1 requires only that this subproblem be solved approximately. Here "approximately" means that the solution of each iteration, s_k, can be obtained in any manner suitable to the application, as long as it satisfies a condition, known as the Fraction of Cauchy Decrease (FCD) condition, concerning the change in the the model a_k from the point x_k to the point $x_k + s_k$. Let $g(x) = \nabla f(x)$. We state the FCD condition in the following form: there exist $\beta > 0$ and $C > 0$, independent of k, for which the step s_k satisfies

$$f(x_k) - a_k(x_k + s_k) \geq \beta \| g(x_k) \| \min\left(\Delta_k, \frac{\| g(x_k) \|}{C} \right). \tag{4.2}$$

Roughly speaking, (4.2) says that we require the approximation to predict some fraction of the improvement in f that is predicted by the minimum of the linear model of f restricted to the trust region.

Moreover, a consequence of (4.2) is that we need not strictly enforce the trust region constraint. The length of the step s_k is acceptable if $\| \, s \, \| \leq \alpha\Delta_k$ for $\alpha \geq 1$ independent of k. See [17] for a more general discussion of the FCD condition.

The FCD condition (4.2) is very mild and, typically, trust region algorithms automatically satisfy this condition by design. In Figure 4.1 we give an algorithm for solving the subproblem (4.1) that satisfies the FCD condition. This algorithm for solving the subproblem is itself based on the classical trust region approach using a local quadratic approximation.

```
Given $x_k \in I\!R^n$,, $\Delta_k > 0$, $\tau \in (0,1)$, and $\alpha \geq 1$, set $y_0 = x_k$, $\delta_0 = \tau\Delta_k$, $v_0 = 0$.
For $j = 0, 1, \ldots$, while $\| \, s \, \| \leq \alpha\Delta_k$ and at least until $v_j \neq 0$ do {
    Construct a quadratic model $q_j(y_j + p) = a_k(y_j) + \nabla a_k(y_j)^T p + \frac{1}{2} p^T B_j p$,
    where $B_j$ approximates the second order information for $a_k$ at $y_j$.
    Find an approximate solution $p_j$ to
        minimize      $q_j(y_j + p)$
        subject to    $\| \, p \, \| \leq \delta_j$
                      $\| \, y_j + p \, \| \leq \Delta_k$
    that satisfies FCD for $a_k$ from $y_j$.
    Compare the actual and predicted decrease in $a_k$:
        $r = \dfrac{a_k(y_j) - a_k(y_j + p_j)}{a_k(y_j) - q_j(y_j + p_j)}$.
    Update $y_j$ according to (4.3) and $\delta_j$ according to (2.4).
    Set $v_{j+1} = v_j + (y_{j+1} - y_j)$.
}
Set $s_k = v_j$.
```

FIG. 4.1. *A trust region algorithm for computing s_k approximately.*

The algorithm in Figure 4.1 for computing the step s_k is a sequence of classical trust region iterations applied to the approximation model a_k as the unconstrained objective. Algorithms for the exact and approximate solution of the subproblem

$$\begin{array}{ll} \text{minimize} & q_j(y_j + p) \\ \text{subject to} & \| \, p \, \| \leq \delta_j \\ & \| \, y_j + p \, \| \leq \Delta_k \end{array}$$

are discussed in [16].

We use a slightly more stringent rule for updating y_j than we used for updating x_k in (2.3). We choose $\mu > 0$, independent of k, j, and update as follows:

$$\text{(4.3)} \qquad \begin{array}{l} \text{If } y_j = x_k \text{, then } y_{j+1} = \begin{cases} y_j + p_j & \text{if } r > \mu \\ y_j & \text{otherwise.} \end{cases} \\ \text{If } y_j \neq x_k \text{, then } y_{j+1} = \begin{cases} y_j + p_j & \text{if } r > 0 \\ y_j & \text{otherwise.} \end{cases} \end{array}$$

This rule, a slight modification of one frequently encountered in both the theoretical analysis and practical implementation of trust region methods, insures that the solution of the subproblem (4.1) satisfies the FCD condition for f at x_k: we do not accept a step from x_k until $r > \mu > 0$ (we are guaranteed that eventually we will find such a step since, if we do not, we reduce the trust region, and ultimately a successful step will be found). Let p_N be that first acceptable step. Since the step generated by the classical trust region algorithm applied to a_k satisfies the FCD condition for a_k at x_k, and $r > \mu$, we have

$$\begin{aligned} & a_k(x_k) - a_k(x_k + p_N) \geq \mu \left(a_k(x_k) - q_N(x_k + p_N) \right) \\ & \geq \mu\beta \parallel \nabla a_k(x_k) \parallel \min \left(\delta_N, \frac{\parallel \nabla a_k(x_k) \parallel}{C} \right), \end{aligned}$$

which, in light of (3.1) and (3.2), yields

$$f(x_k) - a_k(x_k + p_N) \geq \mu\beta \parallel \nabla f(x_k) \parallel \min \left(\delta_N, \frac{\parallel \nabla f(x_k) \parallel}{C} \right),$$

If we place the additional requirement that the Hessians $\nabla^2 a_k(x+s)$ of the approximations a_k are bounded for all s such that $\parallel s \parallel \leq \Delta_k$ uniformly in k, then we can be assured that there exists γ independent of k for which $\delta_N \geq \gamma \Delta_k$, so we arrive at an FCD condition for a_k as an approximation of f at x_k:

$$f(x_k) - a_k(x_k + p_N) \geq \gamma\mu\beta \parallel \nabla f(x_k) \parallel \min \left(\Delta_k, \frac{\parallel \nabla f(x_k) \parallel}{C} \right).$$

Since any steps after step N only produce further decrease in a_k, the step generated by the algorithm in Figure 4.1 produces a fraction of Cauchy decrease for a_k as an approximation of f at x_k.

Because the convergence analysis of our algorithm is virtually identical to the analysis of classical trust region methods, we give only an outline here. Powell's global convergence theorem [21] is a powerful result that provides simple conditions for analyzing all trust region algorithms. The theorem states that if f is bounded below, uniformly continuously differentiable, and the Hessian approximations B_k in (2.1) are uniformly bounded, then the sequence of iterates generated by a classical trust-region algorithm whose steps satisfy a FCD condition satisfies

$$\liminf_{k \to \infty} \parallel \nabla f(x_k) \parallel = 0.$$

If one uses a step acceptance criterion of the form (4.3) instead of (2.3) in the classical algorithm, one has

$$\lim_{k \to \infty} \parallel \nabla f(x_k) \parallel = 0.$$

Since the algorithm that we propose to find steps s_k satisfies a FCD condition for the models a_k (Figure 4.1), similar results hold for our algorithm under the hypothesis that the Hessians $\nabla^2 a_k(x+s)$ are bounded uniformly in k for all s such that $\parallel s \parallel \leq \Delta_k$.

With second-order models one can devise algorithms with convergence properties like those found in [25, 26]. These classes of algorithms insure convergence to points at which the Hessian of f is positive semi-definite, the second-order necessary condition for minimality, and convergence to a local minimizer if f is locally convex around such a point.

Unlike the trust region approach, the line-search strategy [8, 13] does not generalize in a straightforward way to non-quadratic approximation models. If the current iterate is point A, and a line-search method applied to the approximation a_k visits first the point B and then C, we know that the direction from A to B is a descent direction for f, and that the direction from B to C is a descent direction for a_k, but there is no guarantee that the direction from A to C is a descent direction for f. Thus any backtracking mechanism associated with a line-search method will not necessarily produce improvement as one takes shorter steps if the search direction is decided by minimizing an approximation model, since there is no guarantee that such a search direction points in a direction of descent in a neighborhood of x_k. One would need to store the intermediate iterates in the approximation of a_k and perform a search along the piecewise linear path defined by those points.

5. Relationship to model management. The principles that underlie the update of the trust radius in the classical trust region method were presented in §2. We now observe that these principles may suggest ideas for deciding when a model of greater or lesser fidelity may be more suitable. These observations are based on the fact that in the classical trust region approach, the approximation model was always quadratic, and the only way of "improving" the model was to reduce the trust radius, thereby restricting attention to a region where the quadratic model is a better approximation. In the case of general approximation models, however, we may have the option of changing the nature of the model, in addition to changing the trust radius.

Recall that in the classical trust region algorithm given in Figure 2.1, if the model did a very good job of predicting improvement in f, then the update rule for δ_k increases δ_k, thus allowing longer steps at the next optimization iteration $k + 1$, since the model has proven itself to be good over the current trust region. In the case of general approximation models, we also have the option of changing the nature of the model. If the model is doing an exceptionally good job of producing improvement in the system, then we may be able to use a different model, perhaps one with less physical accuracy but lower computational cost.

In the classical algorithm, if the model did a bad job of predicting improvement in f, either because f actually increased with the step s or because f did decrease but not nearly as much as predicted by the quadratic model, we decrease the size of the trust region used in the next optimization iteration, since this is our only mechanism for "improving" the quadratic model, and we know that this model is good if we remain sufficiently close to x_k. In the setting of general approximation models, the need to reduce the trust region also indicates that progress may be more successful with a different choice of models—again, a model management decision.

Finally, in the classical algorithm we leave the size of the trust region alone if the model did an acceptable but not especially noteworthy job of predicting the improvement in f. For general approximation models, this suggests that the model remain the same since its predictive abilities have just proven to be sufficient and there is no strong reason to believe that a less faithful model will be sufficient for the purposes of the optimization.

6. Conclusion. We have presented a trust region approach to the use of approximation models in optimization with provable robustness properties in terms of its convergence to stationary points for the original problem. Moreover the trust region idea of monitoring the improvement predicted by the approximation model to the improvement realized for the true system being optimized may prove useful in suggesting when a model of greater or lesser accuracy may be more suitable. The practical significance of the latter point will be the subject of future investigation.

Acknowledgments. The authors wish to thank Jean-Francois Barthelemy of the NASA Langley Research Center and Greg Shubin and Paul Frank of the Boeing Company for their helpful comments.

Appendix: Examples of first-order approximation models. In this section we briefly review some examples of approximation models that satisfy the first-order requirements (3.1) and (3.2).

Algebraic approximation models. Early work in the development of approximation concepts for structural optimization [22] concentrated on using linear approximations so that mathematical programming techniques such as sequential linear programming (SLP) could be employed. Linear approximations are one instance of the more general form

$$a_k(x) = f(x_k) + \sum_{i=1}^{n} g_i(x_k)(x^i - x^i_k)\phi_i(x^i, x^i_k), \tag{6.1}$$

where $x = (x^1, \cdots, x^n)$ and $g_i = \partial_{x^i} f$. Any approximation of the form (6.1) necessarily satisfies $a_k(x_k) = f(x_k)$. We also see that

$$\partial_{x^i} a_k(x)\Big|_{x=x_k} = g_i \phi_i(x^i_k, x^i_k),$$

so $\nabla a_k(x_k) = \nabla f(x_k)$ if and only if $\phi_i(x^i_k, x^i_k) = 1$.

The choice $\phi_i(x^i, x^i_k) = 1$ yields the first-order Taylor series approximation, the simplest form of (6.1). An alternative approximation seen in structural optimization comes from introducing reciprocal variables into the formulation of the problem. This transformation is based on the observation that a significant class of constraints in structural engineering can be transformed from nonlinear to linear equations by using the reciprocals of the sizing type design variables (at the expense of introducing nonlinearity into the objective function). This leads to the *reciprocal approximation*,

$$a_k(x) = f(x_k) + \sum_{i=1}^{n} g_i(x_k)(x^i - x^i_k)\frac{x^i_k}{x^i}, \tag{6.2}$$

where $\phi_i(x^i, x^i_k) = x^i_k / x^i$.

Early computational results employing these two approximations [28] suggested that the use of such linearizations, in particular, a first-order Taylor series approximation, could be computationally effective (and here it is noted that there is also a significant computational savings to be enjoyed), but that such approximations were not always accurate. It was also observed that a significant problem with the use of reciprocal variables is that the approximation becomes unbounded if any one of the variables approaches zero.

Subsequent work along these lines was aimed at greater accuracy and numerical stability in the approximation model without requiring the costly calculation of higher-order derivatives. Examples of such work include the *modified reciprocal approximation* [14],

$$a_k(x) = f(x_k) + \sum_{i=1}^{n} g_i(x_k)(x^i - x^i_k)\frac{x^i_k + c^i}{x^i + c^i}, \tag{6.3}$$

with $\phi_i(x^i, x_k^1) = (x_k^i + c^i)/(x^i + c^i)$ (where the values of the c^i's are typically small compared to representative values of the corresponding x^i's), and the *conservative approximation* [27],

$$a_k(x) = f(x_k) + \sum_{i=1}^{n} g_i(x_k)(x^i - x_k^i)\phi_i(x^i, x_k^i), \tag{6.4}$$

where

$$\phi_i(x^i, x_k^i) = \begin{cases} 1 & \text{if } x_k^i g_i(x_k) \geq 0 \\ x_k^i/x^i & \text{otherwise.} \end{cases}$$

The conservative approximation has the attractive feature of leading to a convex programming problem and thus is amenable to solution by nonlinear programming techniques that take advantage of the dual problem. This observation has led to the development of a range of convex approximation strategies (in particular, [23, 3, 10]; see [2] for further references).

A slightly different line of inquiry noted that the reciprocal and conservative approximations destroy the linearity of the problem and thus the possibility of using SLP. However, the *posynomial approximation* [9],

$$a_k(x) = f(x_k) \prod_{i=1}^{n} \left(\frac{x^i}{x_k^i} \right)^{\alpha^i},$$

with

$$\alpha^i = \frac{1}{f(x_k)} g_i(x_k),$$

can be treated using geometric programming techniques. This approach is studied in [15, 19, 20, 29]. This approach is noteworthy in that it has an attendant convergence analysis; [1, 18] show that under appropriate conditions, geometric programming techniques, when applied to a posynomial approximation of the original problem, converge to a stationary point of the original problem.

Finally, we briefly mention second-order approximation models of the form

$$\begin{aligned} a_k(x) = {} & f(x_k) + \sum_{i=1}^{n} g_i(x_k)(x^i - x_k^i)\phi_i(x^i, x_k^i) \\ & + \sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij}(x_k)(x^i - x_k^i)(x^j - x_k^j)\psi_{ij}(x^i, x_k^i, x^j, x_k^j), \end{aligned} \tag{6.5}$$

where $\phi_i(x_k^i, x_k^i) = 1$ and $\psi_{ij}(x_k^i, x_k^i, x_k^j, x_k^j) = 1$. The *reciprocal quadratic approximation* is an instance of this type of model:

$$\begin{aligned} a_k(x) = {} & f(x_k) + \sum_{i=1}^{n} g_i(x_k)(x^i - x_k^i)\frac{x_k^i}{x^i} \\ & + \sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij}(x_k)(x^i - x_k^i)(x^j - x_k^j)\frac{x_k^i}{x^i}\frac{x_k^j}{x^j} \end{aligned}$$

$$\phi_i(x^i, x_k^1) = \frac{x_k^i}{x^i} \qquad \psi_{ij}(x^i, x_k^i, x^j, x_k^j) = \frac{x_k^i}{x^i}\frac{x_k^j}{x^j}.$$

The β-correlation approach. The β-correlation method to approximate modeling presented in [6] is a generic approach to correcting a lower-fidelity model f_{lo}, say, one of lower physical fidelity, by scaling. Unlike the models of the preceding section, this approach is not based on any specific mathematical form of the response f.

One defines the scale factor β to be

$$\beta(x) = \frac{f(x)}{f_{\mathrm{lo}}(x)}.$$

Given the current design x_k, one builds a first-order model β_k of β about x_k:

$$\beta_c(x) = \beta(x_k) + \nabla\beta(x_k)^T(x - x_k).$$

The local model of β is then used to scale the lower-fidelity model in order to derive a better approximation of f:

$$f(x) = \beta(x)a(x) \approx \beta_k(x)f_{\mathrm{lo}}(x).$$

It is straightforward to verify that the approximation

$$a_k(x) = \beta_k(x)f_{\mathrm{lo}}(x)$$

satisfies (3.1) and (3.2).

The β-correlation method can be extended to produce second-order approximation models by using a second-order model of β,

$$\tilde{\beta}_k(x) = \beta(x_k) + \nabla\beta(x_k)^T(x - x_k) + \frac{1}{2}(x - x_k)^T\nabla^2\beta(x_k)(x - x_k),$$

and the approximation $\tilde{a_k}(x) = \tilde{\beta}_k(x)f_{\mathrm{lo}}(x)$.

REFERENCES

[1] M. Avriel and A. C. Williams, *Complementary geometric programming*, SIAM Journal on Applied Mathematics, 19 (1970), pp. 125–141.

[2] J.-F. M. Barthelemy and R. T. Haftka, *Approximation concepts for optimum structural design—a review*, Structural Optimization, 5 (1993), pp. 129–144.

[3] V. Braibant and C. Fleury, *An approximation-concepts approach to shape optimal design*, Computer Methods in Applied Mechanics and Engineering, 53 (1985), pp. 119–148.

[4] R. G. Carter, *Multi-model algorithms for optimization*, PhD thesis, Rice University, Department of Computational and Applied Mathematics, Houston, TX, 1986.

[5] R. G. Carter, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM Journal on Numerical Analysis, 28 (1991), pp. 251–265.

[6] K. J. Chang, R. T. Haftka, G. L. Giles, and P.-J. Kao, *Sensitivity-based scaling for approximating structural response*, Journal of Aircraft, 30 (1993), pp. 283–288.

[7] J. E. Dennis and V. Torczon, *Managing approximation models in optimization*, in Multidisciplinary Design Optimization: State-of-the-Art, N. M. Alexandrov and M. Y. Hussaini, eds., SIAM, Philadelphia, 1997.

[8] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[9] R. J. Duffin, E. L. Peterson, and C. M. Zener, *Geometric Programming*, John Wiley & Sons, New York, 1967.

[10] C. Fleury and V. Braibant, *Structural optimization: A new dual method using mixed variables*, International Journal for Numerical Methods in Engineering, 23 (1986), pp. 409–428.

[11] A. Friedlander, J. M. Martínez, and S. A. Santos, *A new trust region algorithm for bound constrained minimization*, tech. rep., Department of Applied Mathematics, University of Campinas, CP 6065, 13081 Campinas SP, Brasil, April 1996.

[12] D. M. Gay, *On convergence testing in model/trust-region algorithms for unconstrained optimization*, Tech. Rep. Computing Science Technical Report No. 104, Bell Laboratories, September 1982.

[13] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[14] R. T. Haftka and C. P. Shore, *Approximation methods for combined thermal/structural design.* NASA TP-1428, June 1979.

[15] P. Hajela, *Geometric programming strategies in large-scale structural synthesis*, AIAA Journal, 24 (1986), pp. 1173–1178.

[16] M. Heinkenschloss, *On the solution of a two ball trust region subproblem*, Mathematical Programming, 64 (1994), pp. 249–276.

[17] J. J. Moré, *Recent developments in algorithms and software for trust region methods*, in Mathematical Programming. The State of the Art, Bonn 1982, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, Berlin, 1983, pp. 258–287.

[18] A. J. Morris, *Approximation and complementary geometric programming*, SIAM Journal on Applied Mathematics, 23 (1972), pp. 527–531.

[19] ———, *Structural optimization by geometric programming*, International Journal of Solids and Structures, 8 (1972), pp. 847–864.

[20] ———, *The optimisation of statically indeterminate structures by means of approximate geometric programming*, in Advisory Group for Aerospace Research & Development, Conference Proceedings No. 123, Second Symposium on Structural Optimization, 1974, pp. 6–1–6–15.

[21] M. J. D. Powell, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, NY, 1975, pp. 1–27.

[22] L. A. Schmit, Jr. and B. Farshi, *Some approximation concepts for structural synthesis*, AIAA Journal, 12 (1974), pp. 692–699.

[23] L. A. Schmit, Jr. and C. Fleury, *Structural synthesis by combining approximation concepts and dual methods*, AIAA Journal, 18 (1980), pp. 1252–1260.

[24] L. A. Schmit, Jr. and H. Miura, *Approximation concepts for efficient structural synthesis*, Tech. Rep. CR–2552, NASA, 1976.

[25] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust region based algorithms for unconstrained minimization with strong global convergence properties*, SIAM Journal on Numerical Analysis, 22 (1985), pp. 47–67.

[26] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 409–426.

[27] J. H. Starnes, Jr. and R. T. Haftka, *Preliminary design of composite wings for buckling, strength, and displacement constraints*, Journal of Aircraft, 16 (1979), pp. 564–570.

[28] O. O. Storaasli and J. Sobieszczanski-Sobieski, *On the accuracy of Taylor approximation for structure resizing*, American Institute of Aeronautics and Astronautics (AIAA) Journal, 12 (1974), pp. 231–233.

[29] A. B. Templeman and S. K. Winterbottom, *Structural design applications of geometric programming*, in Advisory Group for Aerospace Research & Development Conference Proceedings No. 123, Second Symposium on Structural Optimization, 1974, pp. 5–1–5–15.

[30] G. N. Vanderplaats, *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill Series in Mechanical Engineering, McGraw-Hill, 1984.