AN EFFICIENT GRADIENT FLOW METHOD FOR UNCONSTRAINED OPTIMIZATION

A DISSERTATION
SUBMITTED TO THE PROGRAM IN
SCIENTIFIC COMPUTING AND COMPUTATIONAL MATHEMATICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
William Behrman
June 1998

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

Walter Murray
(Principal Adviser)

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

Joseph Oliger

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

Michael Saunders

Approved for the University Committee on Graduate Studies:

To Edward and Mary Behrman, my parents.

Abstract

This dissertation presents a method for unconstrained optimization based upon approximating the gradient flow of the objective function. Under mild assumptions the method is shown to converge to a critical point from any initial point and to converge quadratically in the neighborhood of a solution.

Two implementations of the method are presented, one using explicit Hessians and $O(n^2)$ storage, the other using Hessian-vector products and $O(n)$ storage. These implementations were written in ANSI-standard Fortran 77 for others to use. They have been extensively tested and have proven to be very reliable and efficient in comparison to leading alternative routines.

Acknowledgments

A person could not ask for a better set of advisors, colleagues, and friends than I have had the privilege of knowing and working with. Words cannot express how deeply grateful I am for the many wonderful people who have made this work a joy. I would particularly like to thank the following:

Walter Murray, my principal advisor, for having faith in this project and giving me the freedom to pursue it, and for being most generous with his time, offering sage advice and infusing the undertaking with the necessary sense of humor.

Michael Saunders for also being most generous with his time, for being a patient and kind teacher whose excellent editing greatly improved the quality of the writing and whose ideas played a key role in the software.

Joe Oliger, who was with this project from the start, for giving me the freedom to explore and to formulate my ideas.

Juan Simo for providing encouragement early on and lending his considerable enthusiasm.

Steve Smale for providing the inspiration for the project, both through his ideas and through his approach to science.

Velvel Kahan for giving incisive help on several numerical matters and for having developed ideas used to improve the software.

Horst Simon for providing encouragement early on and for likewise having developed ideas used to improve the software.

Gene Golub for creating and nurturing in the Scientific Computing and Computational Mathematics Program an environment in which this sort of work could flourish.

My fellow students and friends here at Stanford for making this a special place that will always be in my heart.

And finally my parents, to whom this dissertation is dedicated, for the love and encouragement and support and good humor that have sustained me throughout this project and throughout my life. I have truly been blessed.

Contents

Chapter 1

Introduction

Newton's method is the basis of current unconstrained optimization algorithms with the desirable properties of finding a solution from any starting point and converging rapidly in the neighborhood of a solution. This dissertation presents algorithms that are not based on Newton's method but still possess the same desirable properties.

Algorithms based upon Newton's method have at their core the solution of linear systems of equations. For large problems, such algorithms are therefore constrained by their ability to approximate and solve the resulting large linear systems of equations. The algorithms of this dissertation have a different computational kernel. In practice their performance compares favorably with that of the leading alternatives.

1.1 History

On May 3, 1941, Richard Courant gave an address to the American Mathematical Society in which he proposed three methods for numerically solving variational partial differential equations [6]. Two of the methods, the finite element method and the finite difference method, went into widespread use. The third method, which he called the method of gradients, has not been pursued nearly as much.

The idea behind the method of gradients is very old; Courant himself cites work of Hadamard published in 1908 [11]. The idea arose in the study of variational partial differential equations. Each of these equations has a function $f : X \to \mathbb{R}$ (also called a functional) such that a solution of the equation is a minimizer of f. The method of gradients starts with an initial point $x_0 \in X$ and seeks to find a minimizer of f by following a curve ϕ defined by the ordinary differential equation

$$\begin{aligned} \phi'(t) &= -\nabla f(\phi(t)) \\ \phi(0) &= x_0, \end{aligned}$$

where ∇f is the gradient of f. The solution is called an integral curve and is simply the curve that at each instant proceeds in the direction of the steepest descent of f.

In the early days of computing, the finite element and finite difference methods had several advantages over the method of gradients. The finite element and finite difference methods have as their computational kernel the solution of a linear system of equations, and this kernel was particularly well suited to the small linear partial differential equations that were the focus of early numerical work. The method of gradients, on the other hand, has as its computational kernel the approximate solution of an ordinary differential equation, and it was not clear how to best approach this subproblem. According to Peter Lax, who worked with Courant in the early days, another issue was the small memories of early computers, which severely limited program size. The programs required for the finite element and finite difference methods were simpler and shorter than those required for the method of gradients. Finally, the mathematical foundations upon which to build analysis and convergence theory were much stronger for the finite element and finite difference methods than for the method of gradients. For the method of gradients, a convergence theory did not exist even in the case of exact integral curves.

Since these early days, a number of significant changes have occurred. The size of computer memory has greatly increased and no longer places severe constraints on the size of programs for numerical algorithms. In the 1960s the theory of nonlinear functional analysis was greatly expanded and included a convergence theory for the method of gradients in the case of exact integral curves [20]. This dissertation extends the theory to the case of approximate integral curves.

Along with the changes in computers, the nature of the problems to be solved has changed. While in the early days the problems were typically small and linear, now they are much larger and are frequently nonlinear. As problems grow in size, the success of a method depends largely upon how its computational kernel scales. A kernel that works well on small problems may not be as successful on large problems. And while the finite element and finite difference methods are well-suited to linear problems, their application to nonlinear problems requires drastic modifications. The method of gradients, on the other hand, requires no modifications to be applied to nonlinear problems.

While historically much of the motivation for and mathematical analysis behind the method of gradients came from the desire to solve partial differential equations, the function it seeks to minimize need not come from a partial differential equation. The method of gradients is actually a general-purpose method for unconstrained optimization. This dissertation will focus upon it as such and compare it with other methods of optimization. However, some of the test problems on which it will be tested come from partial differential equations, including some that were of great interest to Courant.

1.2 The problem

Unconstrained optimization is one of the fundamental problems of numerical analysis, with numerous applications. The problem is the following:

$$\begin{aligned}&\text{For a function } f : X \to \mathbb{R} \text{ and an initial point } x_0,\\&\text{find a point } x_* \text{ that minimizes } f.\end{aligned} \tag{P}$$

Before we consider this problem, and algorithms for solving it, let us discuss the assumptions we make and our expectations for the solution.

The only assumption we make on the function f is that it has a certain amount of differentiability and that its derivatives are available or can be approximated. When solving an optimization problem, it is usually best to use as much information about the function as possible. If, for example, the function is convex or of a special form, the best approach is usually an algorithm that takes advantage of these properties. We make no assumptions regarding the form or properties of the function beyond a certain differentiability. Software is now available that greatly simplifies the often error-prone process of generating computer code for a function's higher derivatives. Packages such as ADIFOR [2], [3] are applicable to a wide range of functions and reliably produce code for the gradient given code for the function, or code for the Hessian or Hessian-vector product given code for the gradient.

Our problem is unconstrained, that is, the minimizer x_* is not restricted to be in any particular subset $S \subset X$. Constrained problems, where the minimizer x_* is restricted to a proper subset $S \subsetneq X$, are very important, and various approaches to solving them depending upon the nature of the set S exist. One approach is to transform the constrained problem into an unconstrained problem or a sequence of unconstrained problems. Such problems may then be solved by the algorithms described in this dissertation.

Finally, we are seeking a local minimizer x_*, a point where the function f has its lowest value over a local neighborhood. We are not seeking a global minimizer, a point where the function f has its lowest value over the entire space; this is a very difficult task. Algorithms for global minimization often contain algorithms for local minimization but add other features such as a random sampling of initial points. While we are only seeking a local minimizer, we are seeking the local minimizer associated with the initial point in a sense that will be made precise below.

1.3 Unconstrained optimization methods

The most efficient methods for unconstrained optimization currently in use are based upon Newton's method. They rely on the quadratic approximation of the objective function f at the point x,

$$f(x) + \nabla f(x) \cdot (y - x) + \frac{1}{2}(y - x) \cdot \nabla^2 f(x)(y - x),$$

where ∇f is the gradient and $\nabla^2 f$ is the Hessian of f. When $\nabla^2 f(x)$ is positive definite, this approximation has a unique minimizer

$$y = x - (\nabla^2 f(x))^{-1} \nabla f(x), \tag{1.1}$$

and the idea of Newton's method is to use the minimizer of the quadratic approximation as the next point in an iterative process to find a minimizer of f.

Close to a minimizer of f with a positive-definite Hessian, there is a neighborhood where Newton's method works very well, but outside this neighborhood several important issues arise. First, $\nabla^2 f(x)$ need not be positive definite. If $\nabla^2 f(x)$ has a negative eigenvalue, the quadratic approximation will be unbounded below; (1.1) may still be defined, but the value of the quadratic approximation at y could be greater than its value at x. If $\nabla^2 f(x)$ is singular, (1.1) is not defined. Second, even if $\nabla^2 f(x)$ is positive definite, the value of the objective function at y could be greater than its value at x. Consequently, Newton's method need not converge or even be defined.

The shortcomings of Newton's method have led to extensions that fall into two broad families: line-search methods and trust-region methods. Line-search methods insure that the quadratic approximation has a unique minimizer by always using a positive-definite matrix for the second-order term—replacing the Hessian if necessary. If the value of the objective function at the computed point is greater than that at the current point, these methods search along the ray connecting the current and computed points; with a sufficiently small step, a point with a lower value is guaranteed to be found. Trust-region methods take a different approach that uses the Hessian in all cases. At each iteration these methods solve a constrained minimization of the quadratic approximation within a ball centered at the current point and in this way obtain a point even when the quadratic approximation is unbounded below. Determining the solution of this constrained subproblem itself requires an iterative process. If the value of the objective function at the computed point is greater than that at the current point, these methods shrink the size of the ball and perform the constrained minimization again; with a sufficiently small ball, a point with a lower value is guaranteed to be found. Both line-search and trust-region methods have as their computational kernel the solution of a linear system of equations; line-search methods solve one system at each iteration, and trust-region methods solve at least one system at each iteration.

For more information on line-search methods, trust-region methods, and their extensions to the large-scale case where the full Hessian is not available, and for information on software implementations of these methods, see [15, Chapter 2].

The method presented in this dissertation is based upon the solution of an ordinary differential equation. This equation is defined, makes sense, and is solvable regardless of the nature of the Hessian. The solution is a curve guaranteed to have a point that has a lower value of the objective function than the current point.

1.4 Prior work on methods based on ordinary differential equations

The idea of using ordinary differential equations for unconstrained optimization is an old one. A number of methods based on this idea have been proposed over the years, none of which has gone into widespread use. In this section we review this prior work.

Several different ordinary differential equations have been proposed. In what follows, γ will be the displacement from the current point, so the initial condition for all the equations is $\gamma(0) = 0$. One equation, that of Courant's method of gradients, is

$$\gamma'(t) \quad = \quad -\nabla f(x_0 + \gamma(t)). \tag{1.2}$$

A sequence of equations comes from approximating (1.2) locally at a sequence of points,

$$\gamma_i'(t) \quad = \quad -\nabla f(x_i) - \nabla f^2(x_i)\gamma_i(t). \tag{1.3}$$

Another equation, sometimes called the continuous Newton equation, is

$$\gamma'(t) = -(\nabla f^2(x_0 + \gamma(t)))^{-1} \nabla f(x_0 + \gamma(t)). \quad (1.4)$$

This equation has the same disadvantage as (1.1) in that it is undefined when the Hessian is singular. Finally, the equation

$$a\gamma''(t) + b(t)\gamma'(t) + \nabla f(x_0 + \gamma(t)) = 0 \quad (1.5)$$

comes from considering the trajectory of a point mass in the force field $-\nabla f$ with dissipation. The limit of γ depends upon the dissipation, and it need not be the minimizer in the basin of the initial point.

There are no formulas, in general, for the solutions to (1.2), (1.4), or (1.5), so all methods based on these equations use ordinary differential equation solvers. There is a formula for the solution of (1.3), which some methods based on (1.3) use, while others use ordinary differential equations solvers. A disadvantage of ordinary differential equation solvers is that they expend work accurately finding points along the curve γ when all we are interested in is the limit point. Also, to handle stiffness, all of the methods use implicit solvers, which require solving linear systems of equations of the size of the problem at each step. All of the methods reviewed require $O(n^2)$ storage. All were tested on relatively small test problems; half the papers had a largest test problem of size $n = 4$, and the largest test problem in all the papers was one of size $n = 50$.

The following review is in chronological order, and the references for this section appear at the end of the section.

Botsaris and Jacobson [1] use (1.3) and the formula for its solution. They use the Hessian; if it is nonsingular, they replace it with the operator with the same eigenvectors and the absolute value of the eigenvalues so that the solution is bounded.

Vial and Zang [2] use (1.3) and the formula for its solution. They use a quasi-Newton approximation of the Hessian.

Boggs [3] uses (1.2). He uses a predictor-corrector ordinary differential equation solver and quasi-Newton approximations of both the Hessian and its inverse.

Botsaris [4] uses (1.3) and the formula for its solution. He uses an approximation of the Hessian that he updates at each step using the gradient and the Sherman-Morrison formula. In [5], he again uses (1.3) and the formula for its solution. He uses an approximation of the eigensystem of the Hessian that he updates at each step by solving n linear systems of equations each of dimension $n+1$ and then orthogonalizing the resulting n vectors. In [6], he uses (1.4). He uses an implicit ordinary differential equation solver and an approximation of the inverse of the Hessian that he updates at each step using the Sherman-Morrison formula.

Zang [7] uses (1.3) and the formula for its solution. He uses a quasi-Newton approximation of the Hessian.

Aluffi-Pentini, Parisi, and Zirilli [8], [9] use (1.5). Their method is for solving nonlinear equations $g(x) = 0$, but they do this by minimizing $f = g \cdot g$. They use an implicit ordinary differential equation solver and an approximation of the Hessian.

Brown and Bartholomew-Biggs [10] experiment with a number of methods using of (1.2)–(1.5). All of their methods used ordinary differential equation solvers. In their computational experiments, the two most successful methods were based on (1.3); one used the Hessian, the other used a quasi-Newton approximation of the Hessian.

References

[1] C. A. Botsaris and D. H. Jacobson. A Newton-type curvilinear search method for optimization. *J. Math. Anal. Appl.*, 54(1):217–229, 1976.

[2] Jean-Philippe Vial and Israel Zang. Unconstrained optimization by approximation of the gradient path. *Math. Oper. Res.*, 2(3):253–265, 1977.

[3] Paul T. Boggs. An algorithm, based on singular perturbation theory, for ill-conditioned minimization problems. *SIAM J. Numer. Anal.*, 14(5):830–843, 1977.

[4] C. A. Botsaris. Differential gradient methods. *J. Math. Anal. Appl.*, 63(1):177–198, 1978.

[5] C. A. Botsaris. A curvilinear optimization method based upon iterative estimation of the eigensystem of the Hessian matrix. *J. Math. Anal. Appl.*, 63(2):396–411, 1978.

[6] C. A. Botsaris. A class of methods for unconstrained minimization based on stable numerical integration techniques. *J. Math. Anal. Appl.*, 63(3):729–749, 1978.

[7] Israel Zang. A new arc algorithm for unconstrained optimization. *Math. Programming*, 15(1):36–52, 1978.

[8] Filippo Aluffi-Pentini, Valerio Parisi, and Francesco Zirilli. A differential-equations algorithm for nonlinear equations. *ACM Trans. Math. Software*, 10(3):299–316, 1984.

[9] Filippo Aluffi-Pentini, Valerio Parisi, and Francesco Zirilli. Algorithm 617, DAFNE: a differential-equations algorithm for nonlinear equations. *ACM Trans. Math. Software*, 10(3):317–324, 1984.

[10] A. A. Brown and M. C. Bartholomew-Biggs. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *J. Optim. Theory Appl.*, 62(2):211–224, 1989.

1.5 Overview of dissertation

First, a brief illustration of the algorithm that is the subject of this dissertation. For a continuously differentiable objective function f, there is a vector field that at each point is the negative gradient vector $-\nabla f$. From a given point, the algorithm calculates a curve that is an approximation to the integral curve of this vector field from this point. It then searches along this curve for a point that reduces the value of the objective function and then repeats the process until it finds a point that satisfies the specified convergence criteria.

If the objective function is quadratic, then the curve that the algorithm calculates is the exact integral curve. For this reason, the algorithm finds the minimizer of a

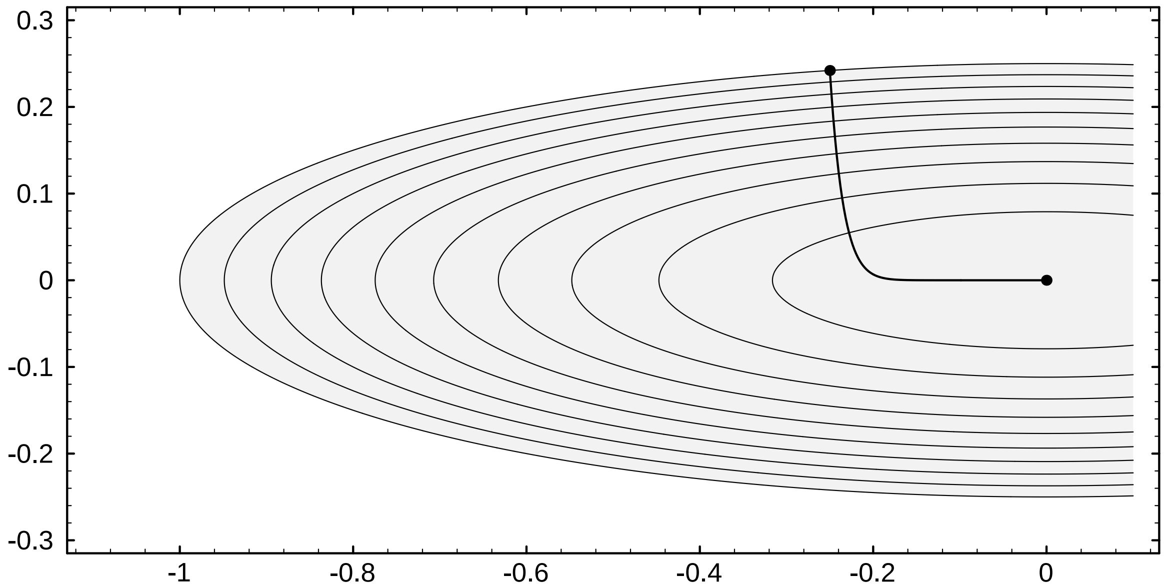

Figure 1.1: Search curve for positive-definite quadratic objective function.

positive-definite quadratic in one step. Figure 1.1 is a contour plot of a positive-definite quadratic, with the algorithm's search curve connecting the initial point on the boundary with the minimizer in the interior. For all objective functions, the algorithm's search curve is initially tangent to the negative gradient. Likewise for all objective functions, if the Hessian at the initial point of the search curve is positive definite, then the search curve will be bounded and the step to the end of the curve is the Newton step.

The objective function value of an indefinite quadratic is unbounded below. Figure 1.2 is a contour plot of the indefinite quadratic $y^2 - x^2$ with part of the algorithm's search curve. This function has its largest values at the top and bottom of the figure and its smallest values at the left and right of the figure. The point $(0, 0)$ at the center is a saddle point. The initial point of the search curve is at the top of the figure. For all objective functions, if the Hessian at the initial point is indefinite and the gradient has a component in the eigenspace of non-negative eigenvalues, then the search curve will be unbounded. In the case of the indefinite quadratic, the objective function value along the unbounded search curve is unbounded below.

A full explanation of the algorithm is given in Chapter 2. In Section 2.1 we tie the

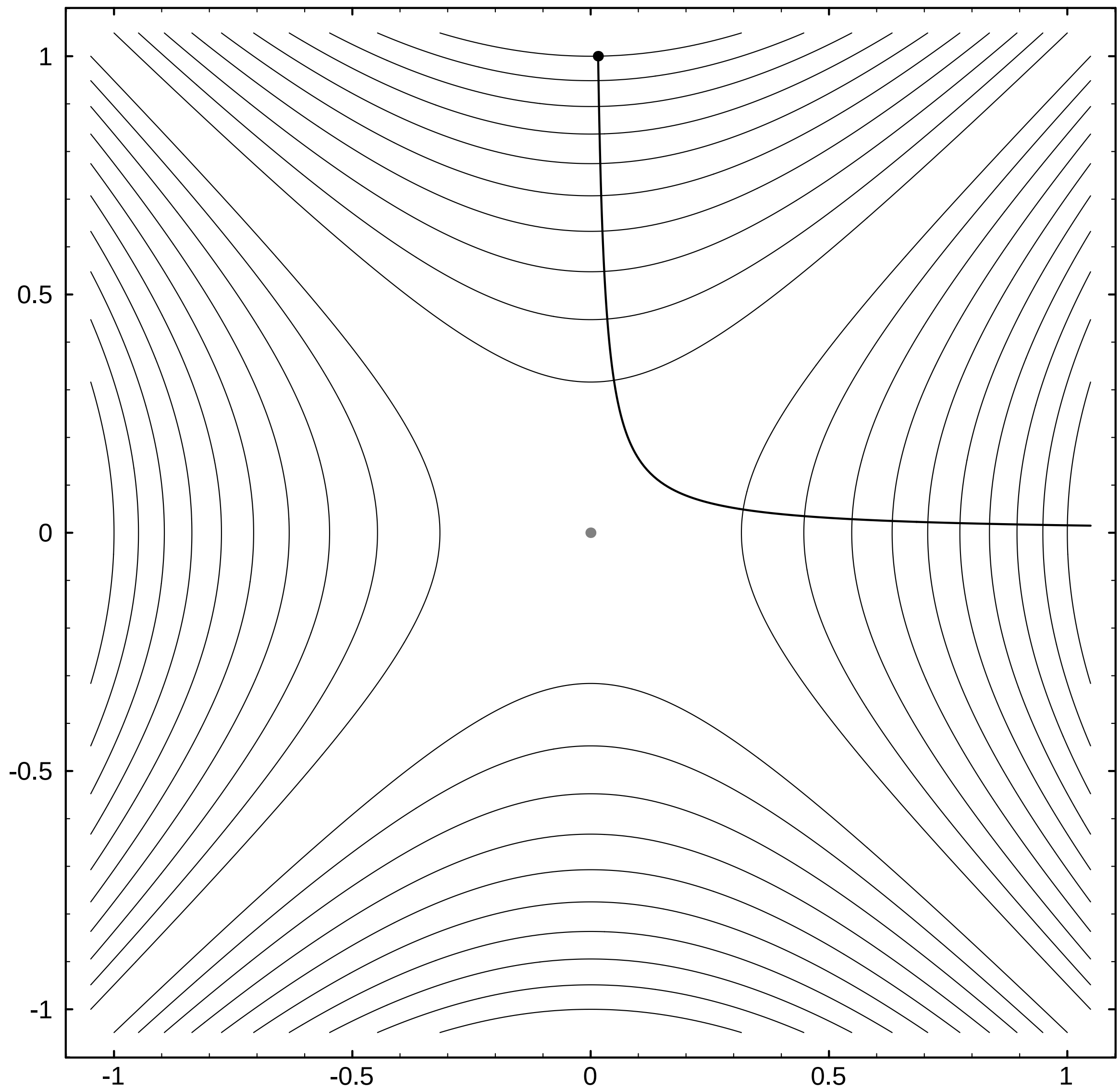

Figure 1.2: Search curve for indefinite quadratic objective function.

algorithm to the mathematical tradition discussed in Section 1.1. We present a key theorem from nonlinear functional analysis, published in 1964, that is central to our analysis. Since the computer algorithm is iterative in nature, the theory of Section 2.1, which is continuous in nature, has to be adapted; this is done in Section 2.2. The mathematical basis and derivation of the algorithm is first explained. We then present search criteria that constrain the union of the search curves to be a sufficiently good approximation of the exact integral curve to apply the continuous theory. We then combine these ideas and give a precise statement of the algorithm. Finally, we show that under mild assumptions the algorithm converges to a critical point from any initial point and converges quadratically in the neighborhood of a solution.

Turning from theory to practice, Chapters 3 and 4 present two implementations of the algorithm. These implementations were written in ANSI-standard Fortran 77 for others to use and have been extensively tested. Chapter 3 describes the implementation UMINH, which uses explicit Hessians and therefore $O(n^2)$ storage. Less storage is required when the search curve is restricted to lie in a subspace. It turns out that by using the right low-dimensional subspaces we can achieve performance close to that using the entire space, with significant savings of storage and time. Chapter 4 describes the implementation UMINHV, which uses the Hessian-vector products and $O(n)$ storage. Both Chapters 3 and 4 discuss the key issues of the implementations and present their performance in numerical tests in comparison to that of leading alternative algorithms. Chapter 5 has some concluding remarks and discusses areas for future work.

Appendix A contains the details of how to use the 142 test problems that were developed to test the software. Appendix B contains the details of how to use UMINH and UMINHV. Finally, Appendix C contains the detailed data of the performance of UMINH, UMINHV, and the alternative routines on the test problems.

Chapter 2

Theory

This chapter presents an algorithm for solving (P), along with its convergence theory. We begin by examining an ordinary differential equation associated with the problem. This ordinary differential equation defines a curve that typically connects the initial point to the solution. We next consider a readily computable approximation to this curve. The algorithm then uses a sequence of these approximations.

2.1 Continuous theory

As our point of departure, we consider an approach to solving problem (P) that uses information of the function f at a continuum of points. While such a method of solution is not practicable on a computer, to consider it is nevertheless instructive because it provides us with an ideal on which to base our algorithms.

If f is continuously differentiable, we can consider the vector field defined by the negative gradient $-\nabla f : X \to X$. Given an initial point x_0 and this vector field, we can define the following ordinary differential equation for ϕ,

$$\begin{aligned} \phi'(t) &= -\nabla f(\phi(t)) \\ \phi(0) &= x_0. \end{aligned} \tag{2.1}$$

A solution ϕ to this equation is called an *integral curve* of $-\nabla f$.

An integral curve for the 2-dimensional test problem BEAL58KO is shown in Figure 2.1. In this contour plot, the integral curve connects the initial point $x_0 = (0, 0)$ with a minimizer and is normal to all contours; the shaded area is the region where the Hessian is positive definite.

If $f : X \to Y$ is a differentiable function between Banach spaces, then a point x is called a *critical point* of f when $Df(x) = 0$. If X is a Hilbert space and $Y = \mathbb{R}$, then we can equivalently say that x is a critical point when $\nabla f(x) = 0$. A solution ϕ of (2.1) is a parameterized curve through X. If this curve contains no critical points of f, then ϕ has the desirable property that f is always decreasing along it,

$$\begin{aligned} \frac{d}{dt} f(\phi(t)) &= \nabla f(\phi(t)) \cdot \phi'(t) \\ &= -\left\| \nabla f(\phi(t)) \right\|^2 . \end{aligned}$$

In addition, ϕ has the desirable property that for a large class of functions it connects the initial point x_0 to a critical point of f.

Theorem 2.1.1 *Let X be a Hilbert space. Let $f : X \to \mathbb{R}$ be Lipschitz continuously differentiable, be bounded below, have isolated critical points, and satisfy the Palais-Smale condition. Then there exists a unique solution $\phi : [0, \infty[\to X$ of (2.1); moreover, $\lim_{t\to\infty} \phi(t)$ exists and is a critical point of f.*

Proof. See [1], Section 3.2C.

We need to define one of the hypotheses of this theorem.

Definition 2.1.2 *Let X be a Banach space, and let $f : X \to \mathbb{R}$ be differentiable. Then f is said to satisfy the* Palais-Smale condition *when any subset $S \subset X$ on which f is bounded and on which $\|Df\|$ is not bounded away from zero contains in its closure a critical point of f.*

An example of a function that does not satisfy this condition is $f : \mathbb{R} \to \mathbb{R}$, $f(x) = e^{-x}$. On the set $[0, \infty[$, f is bounded and $\|Df\|$ is not bounded away from zero, but f has no critical point on $\mathbb{R}$.

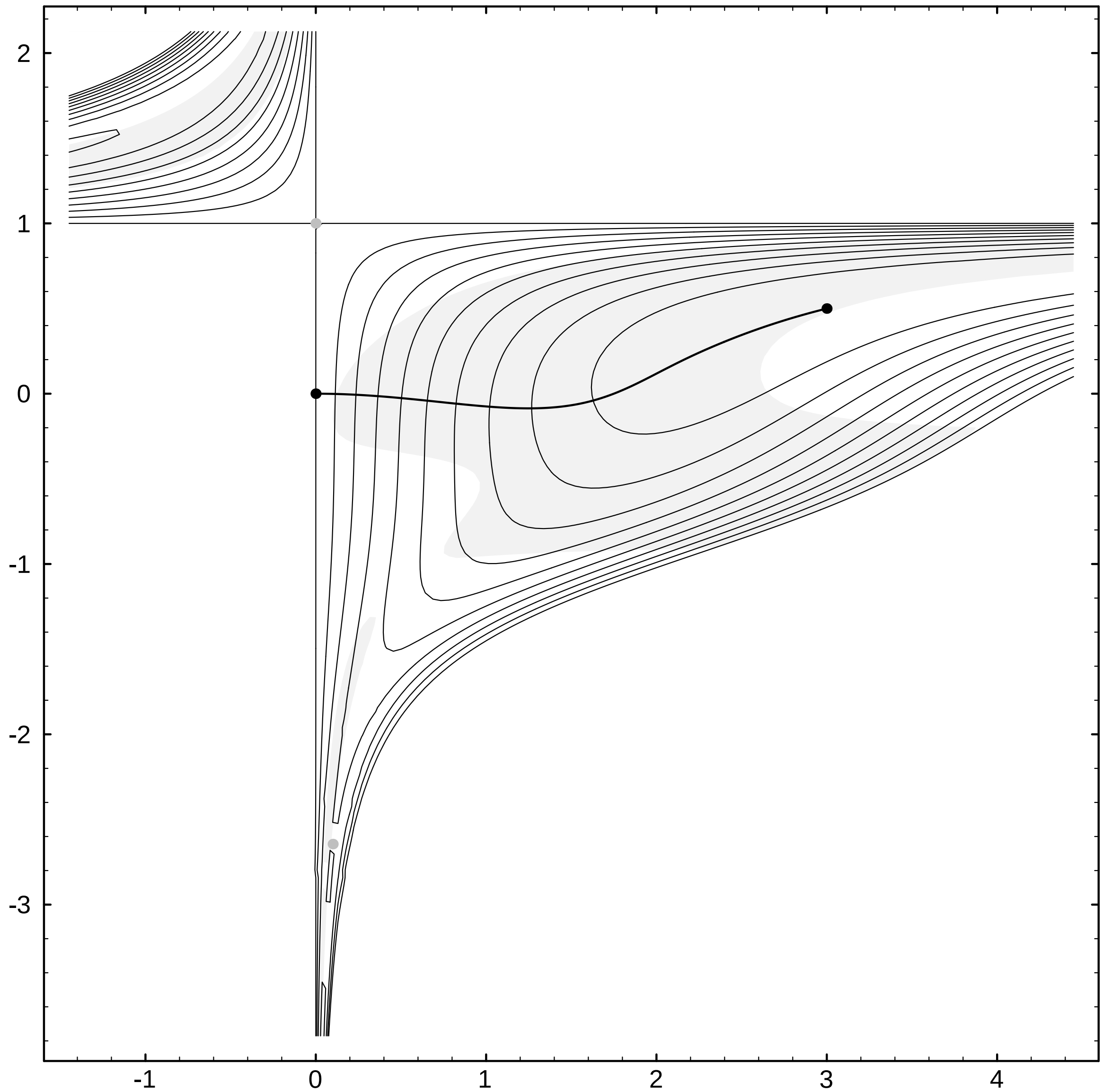

Figure 2.1: An integral curve of $-\nabla f$.

A class of functions that does satisfy this condition are the differentiable functions $f : \mathbb{R}^n \to \mathbb{R}$ such that $f(x) \to +\infty$ as $\|x\| \to \infty$. For if such an f is bounded on a set S, S must be bounded and thus have compact closure.

Through Theorem 2.1.1 we can associate with every initial point x_0 a unique critical point. Now critical points exist that are not local minimizers, and these may be limit points of integral curves ϕ; for example, x_0 could be a saddle point, in which case $\phi(t) = x_0$ for all $t \geq 0$. But in our situation, with isolated critical points, it is extremely rare for an initial point to be associated with a critical point that is not a local minimizer, and we will restrict our attention to initial points that are associated with local minimizers. From such an initial point, the goal for the algorithms that follow will be to find not just *a* local minimizer but *the* local minimizer associated with the initial point.

2.2 Discrete theory

The integral curve in the previous section required information of the function f at a continuum of points. We now consider curves that require information of f only at discrete points.

2.2.1 Derivation of algorithm

Many algorithms for unconstrained nonlinear optimization are based upon approximations of f, usually through its Taylor series expansion. We instead approximate the vector field $-\nabla f$ and then use the integral curves of the approximating vector fields.

Let ϕ_i be the solution to

$$\begin{aligned} \phi_i'(s) &= -\nabla f(x_i + \phi_i(s)) \\ \phi_i(0) &= 0. \end{aligned} \tag{2.2}$$

Then $x_i + \phi_i(s)$ is the integral curve of $-\nabla f$ through x_i. We approximate ϕ_i with a curve γ_i by approximating the vector field $-\nabla f$ near the point x_i with the vector field g_i and letting γ_i be the solution to

$$\begin{aligned} \gamma_i'(s) &= g_i(x_i + \gamma_i(s)) \\ \gamma_i(0) &= 0. \end{aligned} \tag{2.3}$$

Diagrammatically,

$$\begin{array}{ccc} -\nabla f & \xrightarrow{\text{approximation}} & g_i \\ {\scriptstyle \text{exact solution}}\big\downarrow & & \big\downarrow{\scriptstyle \text{exact solution}} \\ \phi_i & & \gamma_i \end{array}$$

Our approximate vector field g_i will always satisfy $g_i(x_i) = -\nabla f(x_i)$. Thus the curve γ_i is initially tangent to ϕ_i, and $f(x_i + \gamma_i(s))$ is initially decreasing. We search along $x_i + \gamma_i(s)$ for a new point x_{i+1} such that $f(x_{i+1}) < f(x_i)$ and certain search criteria are satisfied. From this new point we then repeat the process.

If the only information of f at x_i is its gradient $\nabla f(x_i)$, we can use for g_i the 0th-order approximation,

$$g_i(x) = -\nabla f(x_i).$$

The solution to (2.3) with this constant vector field is $\gamma_i(s) = -\nabla f(x_i)s$. Using the ray $x_i + \gamma_i(s)$ to search for a new point that satisfies certain search criteria and then repeating the process is the *steepest-descent* algorithm of Cauchy [5], [9]. While this algorithm has the desirable property of convergence to a critical point, it can be very slow: its asymptotic rate of convergence is linear with a constant that may be arbitrarily close to 1.

If at x_i the gradient $\nabla f(x_i)$ and the Hessian $\nabla^2 f(x_i)$ are known, we can use for g_i the 1st-order approximation,

$$g_i(x) = -\nabla f(x_i) - \nabla^2 f(x_i)(x - x_i). \tag{2.4}$$

The solution γ_i to (2.3) with this linear vector field is readily computable. We will use these curves below, pasting parts of them together to form a piecewise-smooth curve γ that connects the initial point x_0 with a critical point x_* of f.

2.2.2 Search criteria

The convergence theory for the discrete algorithm is based upon the convergence theory for the continuous algorithm, and for this reason the search must be constrained from going “too far”, into a region in which the approximation used to define the search curve is no longer valid.

Since the integral curve of the negative gradient field is not available to the algorithm, it will measure the validity of the search curve at a point by comparing the tangent to the search curve and the negative gradient field at that point. For the integral curve, the tangent and negative gradient are equal. For the search curve, the algorithm will require that they be sufficiently close in a way made precise in (2.6) below.

Even if the search curve at each iteration were the true integral curve, the iterates would still not converge to a critical point if a series of too small steps were taken. For this reason, the search must be constrained from being “too close” to the initial point. Again, a comparison of the tangent to the search curve and the negative gradient will be used. At the initial point of the search curve, these two vectors are equal, but at later search points they may differ. To keep the search point from being “too close” to the initial point, we will seek to have the two vectors differ a certain minimum amount in a way made precise in (2.7) below.

To express our search criteria, we first define a set of vectors “close to” a given vector $x \in X$. For $p = (p_1, p_2, p_3)$ with $0 < p_1 \leq 1 \leq p_2 < \infty$ and $0 < p_3 \leq 1$, let

$$V(x,p) \;=\; \{y \in X : p_1 \|x\| \leq \|y\| \leq p_2 \|x\| \text{ and } p_3 \|x\| \|y\| \leq x \cdot y\}. \tag{2.5}$$

This set is the intersection of a spherical shell and a cone. The set

$$\{y \in X : p_1 \|x\| \leq \|y\| \leq p_2 \|x\|\}$$

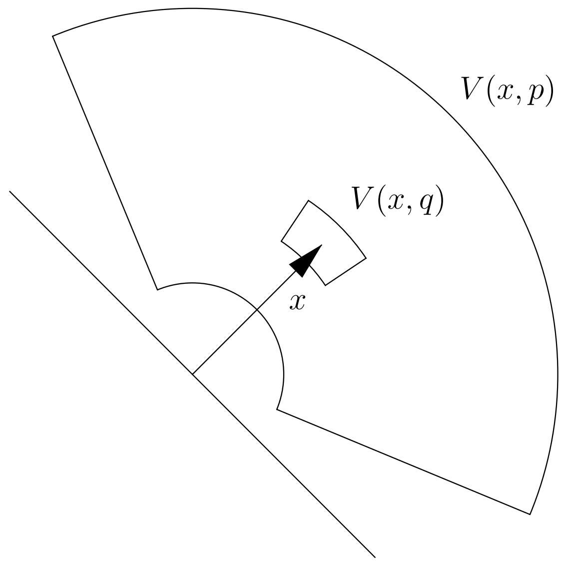

Figure 2.2: The sets $V(x,p)$ and $V(x,q)$.

is a spherical shell of vectors; the closer p_1 and p_2 are to 1, the closer the magnitudes of these vectors are to x. The set

$$\{y \in X : p_3 \|x\| \, \|y\| \le x \cdot y\}$$

is a cone of vectors; the closer p_3 is to 1, the smaller the angle between these vectors and x. The algorithm actually uses two sets, $V(x,q) \subsetneq V(x,p)$, where $q = (q_1, q_2, q_3)$ and $p = (p_1, p_2, p_3)$ satisfy

$$0 < p_1 < q_1 < 1 < q_2 < p_2 < \infty$$
$$0 < p_3 < q_3 < 1,$$

see Figure 2.2.

We may now express our criteria for $x_{i+1} = x_i + \gamma_i(s_i)$. The restriction keeping $\gamma_i'(s)$ from being "too far" from $-\nabla f(x_i + \gamma_i(s))$ is that

$$\gamma_i'(s) \;\in\; V(-\nabla f(x_i + \gamma_i(s)), p), \quad \text{for } s \in [0, s_i]. \tag{2.6}$$

As a simple consequence of (2.6),

$$\begin{aligned} \frac{d}{ds} f(x_i + \gamma_i(s)) &= \nabla f(x_i + \gamma_i(s)) \cdot \gamma_i'(s) \\ &< 0, \quad \text{for } s \in [0, s_i], \end{aligned}$$

and therefore $f(x_{i+1}) < f(x_i)$. The restriction of $\gamma_i'(s)$ from being "too close" to $-\nabla f(x_i + \gamma_i(s))$ is that

$$\gamma_i'(s_i) \notin V(-\nabla f(x_i + \gamma_i(s_i)), q). \tag{2.7}$$

The integral curve $\gamma_i(s)$ of the vector g_i defined by (2.4) is defined for all $s \geq 0$. Now it may happen that $\gamma_i'(s) \in V(-\nabla f(x_i + \gamma_i(s)), q)$ for all $s \geq 0$. But if there is an $S > 0$ such that $\gamma_i'(S) \notin V(-\nabla f(x_i + \gamma_i(S)), q)$, then from the continuity of γ_i' and the intermediate value theorem there must be an $s_i \leq S$ such that (2.6) and (2.7) simultaneously hold.

Figure 2.3 illustrates the tangent and the negative gradient at various points along the search curve $x_i + \gamma_i(s)$. At point y_1, (2.6) is satisfied but not (2.7), hence y_1 is "too close". At point y_3, (2.7) is satisfied but not (2.6), hence y_3 is "too far". At point y_2, both (2.6) and (2.7) are satisfied, and hence y_2 could be used as x_{i+1}.

The derivation of the algorithm and its convergence theory are quite different from that of line-search methods. Still, it is instructive to compare the search criteria for the former with those commonly used for the latter. For this purpose, let us adopt the notation $f(s)$ for $f(x_i + \gamma_i(s))$ and $f'(s)$ for $\frac{d}{ds} f(x_i + \gamma_i(s))$, where γ_i is either a search curve from the algorithm or a ray from a line-search method.

Line-search methods commonly constrain the search parameter s_i from being "too far" by requiring that it satisfy

$$f(s_i) < f(0) - \mu s_i,$$

for a constant $\mu > 0$. Whereas line-search methods are in a sense parameterized by arclength, the curves of the algorithm are not and may have bounded length with unbounded parameter. On a quadratic function with positive-definite Hessian, for

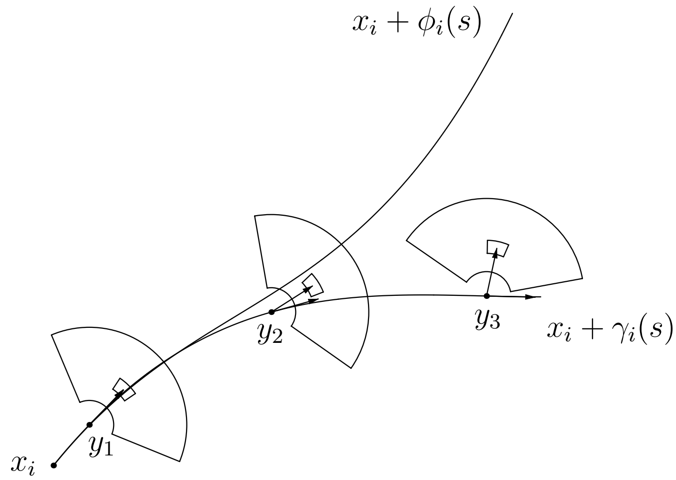

Figure 2.3: Tangent and negative gradient along search curve.

example, the algorithm would find the minimizer in one step as the limit of the bounded curve $\gamma_0(s)$ as $s \to \infty$. The line-search requirement would be too strict in this case; nevertheless, (2.6) implies that the weaker requirement of $f(s_i) < f(0)$ is always satisfied.

Line-search methods commonly constrain the search parameter s_i from being "too close" by requiring that it satisfy

$$|f'(s_i)| \leq \beta\, |f'(0)|,$$

for a constant $0 < \beta < 1$. Both this requirement and (2.7) have the same aim of ensuring a sufficient change in the function f, and the algorithm need not contain points that satisfy either. We use (2.7) because of its close relationship to (2.6). At all points where (2.7) is not satisfied, (2.6) is; and if (2.7) is not satisfiable, this means that the current search curve is sufficiently accurate for the algorithm to stop. On the other hand, both (2.7) and the line-search requirement are satisfiable if the search contains a point where $f'(s) = 0$.

2.2.3 Statement of algorithm

Having explained the approximations g_i to the vector field $-\nabla f$, the integral curves γ_i of g_i to be searched along, and the search criteria, we now combine these ideas into an algorithm that will paste together parts of the γ_i into one curve that connects the initial point x_0 with a critical point of f.

Algorithm 2.2.1

for finding a critical point of f from the initial point x_0.

Let X be a Hilbert space. Let $f : X \to \mathbb{R}$ be twice continuously differentiable and bounded below. Let $x_0 \in X$. Let parameters $p, q \in \mathbb{R}^3$ be such that

$$0 < p_1 < q_1 < 1 < q_2 < p_2 < \infty$$
$$0 < p_3 < q_3 < 1.$$

$t_0 = 0$;

domain$(\gamma) = [0, 0]$;

$\gamma(0) = x_0$;

$i = 0$;

while (domain$(\gamma) \neq [0, \infty[$)

Let $\gamma_i : \mathbb{R} \to X$ be the unique solution to

$$\begin{aligned} \gamma_i'(s) &= -\nabla f(x_i) - \nabla^2 f(x_i)\gamma_i(s) \\ \gamma_i(0) &= 0; \end{aligned}$$

if There is an $s_i > 0$ such that

$$\begin{aligned} \gamma_i'(s) &\in V(-\nabla f(x_i + \gamma_i(s)), p), \qquad \text{for } 0 \leq s \leq s_i \\ \gamma_i'(s_i) &\notin V(-\nabla f(x_i + \gamma_i(s_i)), q). \end{aligned}$$

then

Extend γ to include the interval $]t_i, t_i + s_i]$

$$\gamma(t_i + s) = x_i + \gamma_i(s), \qquad \text{for } 0 < s \le s_i;$$

$t_{i+1} = t_i + s_i$;

$x_{i+1} = \gamma(t_{i+1})$;

$i = i + 1$;

else

Extend γ to include the interval $]t_i, \infty[$

$$\gamma(t_i + s) = x_i + \gamma_i(s), \qquad \text{for } s > 0.$$

end if

end while

2.2.4 Convergence theory of algorithm

Convergence to critical point from any initial point

We first prove that the algorithm solves the problem for which it was designed.

Theorem 2.2.1 *Let X be a Hilbert space, and let $f : X \to \mathbb{R}$ be twice continuously differentiable, be bounded below, have isolated critical points, and satisfy the Palais-Smale condition. Then for any initial point x_0 and any curve γ generated by Algorithm 2.2.1, $\lim_{t\to\infty} \gamma(t)$ exists and is a critical point of f. In particular, the critical point x_* is either found in a finite number of iterations or $\lim_{i\to\infty} x_i = x_*$.*

Proof. The theorem follows immediately from the following two propositions. □

Proposition 2.2.2 *Let X be a Hilbert space, and let $f : X \to \mathbb{R}$ be twice continuously differentiable and bounded below. Then Algorithm 2.2.1 extends the domain of γ to the entire interval $[0, \infty[$.*

Proof. Suppose that the algorithm extends the domain of γ to $[0, t_*[$ where $t_* < \infty$. We will show that this assumption leads to a contradiction.

From (2.24) and the fact that f is bounded below, we have an $M > 0$ such that for all $t_1, t_2 \in [0, t_*[$,

$$\left| \int_{t_1}^{t_2} \|\nabla f(\gamma(t))\|^2 \, dt \right| \leq M.$$

Thus from (2.23), we have that γ is Cauchy. Hence there is an $x_* \in X$ such that $\gamma(t) \to x_*$ as $t \to t_*$. From the continuity of $D^2 f$, there is an $L > 0$ and an $R > 0$ such that $\|D^2 f(x)\| \leq L$ for $x \in X$ with $\|x - x_*\| \leq R$. From the continuity of γ, there is an $i_R \geq 0$ such that $\|\gamma(t) - x_*\| \leq R$ for $t \in [t_{i_R}, t_*[$. Let $A_i = -\nabla^2 f(x_i)$ and $b_i = -\nabla f(x_i)$ for $i \geq 0$.

Recall that for $i \geq 0$,

$$\gamma_i'(0) = -\nabla f(x_i + \gamma_i(0)) = b_i.$$

The search criterion for the algorithm,

$$\gamma_i'(s_i) \notin V(-\nabla f(x_i + \gamma_i(s_i)), q),$$

and Lemma 2.2.8 below imply that for $i \geq 0$ at least one of the following inequalities holds:

$$\begin{aligned} Q \|b_i\| &< \|\gamma_i'(s_i) - b_i\| \\ Q \|b_i\| &< \|-\nabla f(x_i + \gamma_i(s_i)) - b_i\|, \end{aligned} \tag{2.8}$$

where $Q = \min\{(1 - q_1)/(1 + q_1), (q_2 - 1)/(q_2 + 1), ((1 - q_3)/2)^{1/2}\} > 0$. We may bound the right-hand sides of (2.8) for $i \geq i_R$ using the bound on $D^2 f$ and the mean value inequality,

$$\begin{aligned} \|\gamma_i'(s_i) - b_i\| = \|A_i \gamma_i(s_i) + b_i - b_i\| &\leq L \|\gamma_i(s_i)\| \\ \|-\nabla f(x_i + \gamma_i(s_i)) - b_i\| &\leq L \|\gamma_i(s_i)\|. \end{aligned} \tag{2.9}$$

Combining (2.8) and (2.9), we have for $i \geq i_R$,

$$Q \|b_i\| \quad < \quad L \|\gamma_i(s_i)\| . \tag{2.10}$$

Now for $i \geq 0$,

$$\gamma_i(s_i) \quad = \quad \int_0^{s_i} e^{(s_i - r)A_i} b_i \, dr.$$

Hence for $i \geq i_R$,

$$\begin{aligned} \|\gamma_i(s_i)\| \quad &\leq \quad \int_0^{s_i} e^{(s_i - r)L} \|b_i\| \, dr \\ &= \quad \frac{1}{L}(e^{s_i L} - 1) \|b_i\| . \end{aligned} \tag{2.11}$$

By the assumption that $t_* < \infty$, we know that for each $i \geq 0$ there must be an $s_i < \infty$ that satisfies (2.7). But if $b_i = 0$ for any $i \geq 0$, then we would have $\gamma_i(s) = 0$ and $\gamma_i'(s) = 0$ for all $s \geq 0$, and no $s_i \geq 0$ would satisfy (2.7). Hence $b_i \neq 0$ for $i \geq 0$. It follows from this fact, (2.10), and (2.11) that for $i \geq i_R$,

$$Q \quad < \quad e^{s_i L} - 1.$$

This and the fact that $Q > 0$ yields

$$0 \quad < \quad \frac{1}{L} \ln(1 + Q) \quad < \quad s_i.$$

Finally,

$$t_* \quad = \quad \sum_{i=0}^{\infty} s_i \quad \geq \quad \sum_{i=i_R}^{\infty} s_i \quad > \quad \sum_{i=i_R}^{\infty} \frac{1}{L} \ln(1 + Q) \quad = \quad \infty,$$

contradicting the assumption that $t_* < \infty$. □

The algorithm generates a curve with the following property.

Definition 2.2.3 *A p-*approximate integral curve *of the vector field* $-\nabla f$ *is a continuous, piecewise continuously differentiable map* γ *from an interval* $I \subset \mathbb{R}$ *to* X *such that at all points* $t \in I$ *where* γ *is differentiable,*

$$\gamma'(t) \in V(-\nabla f(\gamma(t)), p). \tag{2.12}$$

We may now state the second proposition, which completes the proof of Theorem 2.2.1.

Proposition 2.2.4 *Let* X *be a Hilbert space, and let* $f : X \to \mathbb{R}$ *be continuously differentiable, be bounded below, have isolated critical points, and satisfy the Palais-Smale condition* (Definition 2.1.2)*. Let* $\gamma : [0, \infty[\to X$ *be a p-approximate integral curve of* $-\nabla f$*. Then* $\lim_{t\to\infty} \gamma(t)$ *exists and is a critical point of* f*.*

Proof. From (2.24) and the fact that f is bounded below, it follows that

$$\int_0^\infty \|\nabla f(\gamma(t))\|^2 \, dt < \infty.$$

We may therefore construct a sequence $\{t_i\}_{i\in\mathbb{I}} \subset [0, \infty[$ such that $t_i \to \infty$ and $\|\nabla f(\gamma(t_i))\| \to 0$ as $i \to \infty$. Again from (2.24) and the fact that f is bounded below, it follows that $\{f(\gamma(t_i))\}_{i\in\mathbb{I}}$ is bounded. Thus from the Palais-Smale condition, we have a subsequence $\{t_j\}_{j\in\mathbb{J}} \subset \{t_i\}_{i\in\mathbb{I}}$ and a point $x_* \in X$ such that $\lim_{j\to\infty} \gamma(t_j) = x_*$ and $\nabla f(x_*) = 0$.

Our goal is to show that $\lim_{t\to\infty} \gamma(t) = x_*$.

Suppose otherwise. If γ does not converge to x_*, then there is an $R_1 > 0$ such that for any point $t \geq 0$ there is a later point $s > t$ such that $\|\gamma(s) - x_*\| > R_1$. Since the critical points of f are isolated, there is an $R_2 > 0$ such that for any critical point $x_c \neq x_*$ we have $\|x_c - x_*\| > R_2$. Let $R = \min\{R_1, R_2\}$. Since $t_j \to \infty$ and $\gamma(t_j) \to x_*$ as $j \to \infty$, for every point $t \geq 0$ such that $\|\gamma(t) - x_*\| > R$, there is a later point $s > t$ such that $\|\gamma(s) - x_*\| < R/2$. And by construction, for every point $t \geq 0$ such that $\|\gamma(t) - x_*\| < R/2$, there is a later point $s > t$ such that $\|\gamma(s) - x_*\| > R$. From the continuity of γ, we may therefore find an infinite number of disjoint intervals

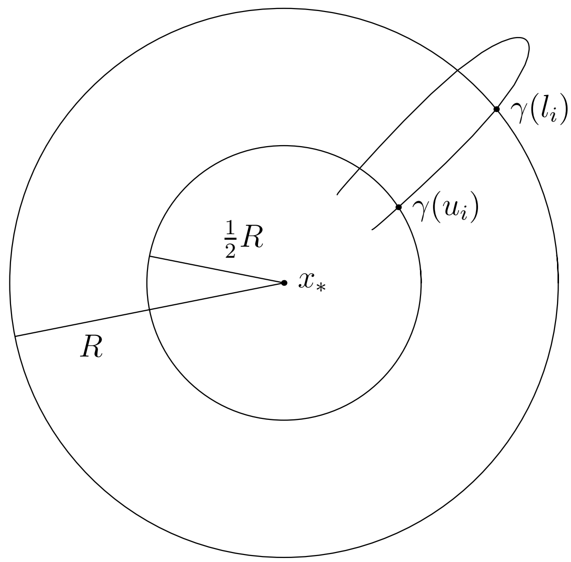

Figure 2.4: $\gamma([l_i, u_i])$.

$[l_i, u_i]$ with $\|\gamma(l_i) - x_*\| = R$, $\|\gamma(u_i) - x_*\| = R/2$, and $R/2 \leq \|\gamma(t) - x_*\| \leq R$ for $t \in [l_i, u_i]$ for $i \geq 0$. See Figure 2.4.

Since f is bounded on $\gamma([0, \infty[) \bigcap \{x \in X : R/2 \leq \|x - x_*\| \leq R\}$ and since the closure of this set has no critical points by construction, we have by the Palais-Smale condition an $M > 0$ such that $\|\nabla f(\gamma(t))\| \geq M$ for all $t \geq 0$ with $R/2 \leq \|\gamma(t) - x_*\| \leq R$. Using this fact, the fact that $\|\gamma(u_i) - \gamma(l_i)\| \geq R/2$ for $i \geq 0$, and

(2.22) and (2.24), we get

$$\begin{aligned}
\lim_{t\to\infty} f(\gamma(t)) &\leq f(\gamma(0)) - p_1 p_3 \int_0^\infty \|\nabla f(\gamma(t))\|^2 \, dt \\
&\leq f(\gamma(0)) - p_1 p_3 \sum_{i=0}^{\infty} \int_{l_i}^{u_i} \|\nabla f(\gamma(t))\|^2 \, dt \\
&\leq f(\gamma(0)) - p_1 p_3 M \sum_{i=0}^{\infty} \int_{l_i}^{u_i} \|\nabla f(\gamma(t))\| \, dt \\
&\leq f(\gamma(0)) - \frac{p_1 p_3 M}{p_2} \sum_{i=0}^{\infty} \|\gamma(u_i) - \gamma(l_i)\| \\
&\leq f(\gamma(0)) - \sum_{i=0}^{\infty} \frac{p_1 p_3 M}{p_2} \frac{R}{2} \\
&= -\infty,
\end{aligned}$$

contradicting the assumption that f is bounded below. Thus $\lim_{t\to\infty} \gamma(t) = x_*$. □

Local quadratic rate of convergence

We next prove that the algorithm *quickly* solves the problem for which it was designed.

The algorithm typically generates an infinite sequence of points converging to a critical point x_* of the function f. The number of iterations required to reach a given radius of x_* will typically depend upon the distance of the initial point x_0 to x_*. For this reason, our measure of the quickness of the algorithm will be its rate of convergence in the neighborhood of a solution. For the following theorem, we combine the finite termination case with the infinite case by extending the finite sequence generated by the algorithm to an infinite sequence in the following way: if the solution x_* is found at iteration k, define $x_i = x_*$ for $i > k$.

Theorem 2.2.5 *Let X be a Hilbert space, and let $f : X \to \mathbb{R}$ be twice differentiable. Let $x_* \in X$ be such that $\nabla f(x_*) = 0$, the spectrum $\sigma(\nabla^2 f(x_*)) \subset \mathbb{R}_{>0}$, and $\nabla^2 f$ is Lipschitz continuous in a neighborhood of x_*. Then there is an $R > 0$ such that from any initial point x_0 with $\|x_0 - x_*\| \leq R$ the sequence $\{x_i\}_{i\in\mathbb{I}}$ generated by Algorithm 2.2.1 converges to x_* quadratically; that is, there is an $M > 0$ such that for all*

$i \in \mathbb{I}$,

$$\|x_{i+1} - x_*\| \leq M \|x_i - x_*\|^2 .$$

Proof. By assumption, there is a neighborhood of x_* on which $\nabla^2 f$ is Lipschitz continuous with constant L. By the assumption on the spectrum of $\nabla^2 f(x_*)$, there is a neighborhood of x_* on which the spectrum of $\nabla^2 f$ belongs to an interval $[c_l, c_u]$ with $0 < c_l \leq c_u < \infty$; see, for example, [7, Section 7.6]. Let $B(x_*, R_1)$ be an open ball belonging to both neighborhoods.

Let $A_i = -\nabla f^2(x_i)$ and $b_i = -\nabla f(x_i)$. Recall that at iteration i of the algorithm, the search curve γ_i satisfies

$$\gamma_i'(s) = A_i \gamma_i(s) + b_i \tag{2.13}$$

and

$$\gamma_i(s) = \int_0^s e^{(s-r)A_i} b_i \, dr. \tag{2.14}$$

From (2.14), we get that

$$\gamma_i'(s) = e^{sA_i} b_i. \tag{2.15}$$

For $x_i \in B(x_*, R_1)$, A_i is invertible. In this case, (2.14) yields

$$\gamma_i(s) = A_i^{-1}(e^{sA_i} - I) b_i. \tag{2.16}$$

Below we will need to bound $\|g(A)\|$, where $A \in L(X)$ is self-adjoint and $g : \sigma(A) \to \mathbb{R}$ is continuous. To do this, we will make use of the fact that $g(A)$ is self-adjoint, $\sigma(g(A)) = g(\sigma(A))$, and that for any self-adjoint $B \in L(X)$, we have $\|B\| = \sup_{\lambda \in \sigma(B)} |\lambda|$; see, for example, [22, Section 7.1].

We first find a smaller open ball $B(x_*, R_2) \subset B(x_*, R_1)$ so that from any initial point $x_i \in B(x_*, R_2)$ the curve $x_i + \gamma_i(s)$ remains confined to $B(x_*, R_1)$ for all $s \geq 0$.

Let $x_i \in B(x_*, R_1)$ with $\|x_i - x_*\| = \epsilon$. For $s \geq 0$,

$$\begin{aligned} \|\gamma_i(s)\| &\leq \left\| {A_i}^{-1}(e^{sA_i} - I) \right\| \|b_i\| \\ &= \frac{1}{c_l}(1 - e^{-c_l s}) \|b_i\| \\ &\leq \frac{1}{c_l} \|b_i\| . \end{aligned} \tag{2.17}$$

Since $\|\nabla^2 f(x)\| \leq c_u$ for $x \in B(x_*, R_1)$, we have by the mean value inequality that

$$\begin{aligned} \|b_i\| &= \|\nabla f(x_i) - \nabla f(x_*)\| \\ &\leq c_u \|x_i - x_*\| . \end{aligned} \tag{2.18}$$

It follows that for $s \geq 0$,

$$\begin{aligned} \|x_i + \gamma_i(s) - x_*\| &\leq \|x_i - x_*\| + \|\gamma_i(s)\| \\ &\leq (1 + \frac{c_u}{c_l})\epsilon. \end{aligned}$$

Therefore the curve $x_i + \gamma_i(s)$ will be confined to $B(x_*, R_1)$ whenever $x_i \in B(x_*, R_2)$ with $R_2 = R_1/(1 + c_u/c_l)$.

Using the notation of the algorithm,

$$x_{i+1} = x_i + \gamma_i(s_i).$$

For $x_i \in B(x_*, R_2)$, we then have from (2.16) that

$$\begin{aligned} x_{i+1} - x_* &= x_i + {A_i}^{-1} e^{s_i A_i} b_i - {A_i}^{-1} b_i - x_* \\ &= -{A_i}^{-1}(b_i + A_i(x_* - x_i) - e^{s_i A_i} b_i). \end{aligned} \tag{2.19}$$

We first seek to bound $\|b_i + A_i(x_* - x_i)\|$. By Taylor's theorem and the Lipschitz continuity of $\nabla^2 f$,

$$\begin{aligned} \|b_i + A_i(x_* - x_i)\| &= \left\|\nabla f(x_*) - \nabla f(x_i) - \nabla^2 f(x_i)(x_* - x_i)\right\| \\ &= \left\|\int_0^1 (\nabla^2 f(x_i + r(x_* - x_i)) - \nabla^2 f(x_i))\, dr\, (x_* - x_i)\right\| \\ &\leq \frac{1}{2} L \|x_i - x_*\|^2 . \end{aligned} \tag{2.20}$$

We next seek to bound $\left\|e^{s_i A_i} b_i\right\|$. From the search criteria, we have

$$\gamma_i'(s_i) \quad \notin \quad V(-\nabla f(x_i + \gamma_i(s_i)), (q_1, q_2, q_3)).$$

By Lemma 2.2.6, this implies

$$-\nabla f(x_i + \gamma_i(s_i)) \quad \notin \quad V(\gamma_i'(s_i), (1/q_2, 1/q_1, q_3)).$$

And by Lemma 2.2.7, we have

$$Q \|\gamma_i'(s_i)\| \quad < \quad \|-\nabla f(x_i + \gamma_i(s_i)) - \gamma_i'(s_i)\| ,$$

where $Q = \min\{(1 - 1/q_2), (1/q_1 - 1), (1 - {q_3}^2)^{1/2}\} > 0$. Hence from (2.13), (2.15), Taylor's theorem, and the Lipschitz continuity of $\nabla^2 f$,

$$\begin{aligned} Q\left\|e^{s_i A_i} b_i\right\| &< \left\|-\nabla f(x_i + \gamma_i(s_i)) + \nabla^2 f(x_i)\gamma_i(s_i) + \nabla f(x_i)\right\| \\ &= \left\|-\int_0^1 (\nabla^2 f(x_i + r\gamma_i(s_i)) - \nabla^2 f(x_i))\, dr\, \gamma_i(s_i)\right\| \\ &\leq \frac{1}{2} L \|\gamma_i(s_i)\|^2 . \end{aligned}$$

And using (2.17) and (2.18), we get

$$\left\|e^{s_i A_i} b_i\right\| \leq \frac{L{c_u}^2}{2Q{c_l}^2} \|x_i - x_*\|^2 . \tag{2.21}$$

Combining (2.19)–(2.21),

$$\|x_{i+1} - x_*\| \;\; \leq \;\; M \|x_i - x_*\|^2 ,$$

where $M = (L/(2c_l))(1 + {c_u}^2/(Q{c_l}^2))$. Finally, let $R = \min\{R_2, M/2\}$. Then $x_i \in B(x_*, R)$ implies

$$\|x_{i+1} - x_*\| \;\; \leq \;\; \frac{1}{2} \|x_i - x_*\| ,$$

which in turn implies that $x_{i+1} \in B(x_*, R)$ and $x_i \to x_*$ as $i \to \infty$. □

Ancillary bounds and lemmas

We now present some bounds and lemmas for the p-approximate integral curves (Definition 2.2.3) constructed by the algorithm.

Bounds for approximate integral curves. Let $f : X \to \mathbb{R}$ be continuously differentiable, and let $\gamma : I \subset \mathbb{R} \to X$ be a p-approximate integral curve of $-\nabla f$. The following bounds follow immediately from (2.12) and (2.5). For $t_1, t_2 \in I$ with $t_1 \leq t_2$,

$$\begin{aligned} \|\gamma(t_2) - \gamma(t_1)\| \;\; &\leq \;\; \int_{t_1}^{t_2} \|\gamma'(t)\| \, dt \\ &\leq \;\; p_2 \int_{t_1}^{t_2} \|\nabla f(\gamma(t))\| \, dt, \end{aligned} \tag{2.22}$$

and

$$\|\gamma(t_2) - \gamma(t_1)\| \;\; \leq \;\; p_2 \left(\int_{t_1}^{t_2} \|\nabla f(\gamma(t))\|^2 \, dt \right)^{1/2} (t_2 - t_1)^{1/2} . \tag{2.23}$$

For $t_1, t_2 \in I$,

$$
\begin{aligned}
f(\gamma(t_2)) - f(\gamma(t_1)) &= \int_{t_1}^{t_2} \nabla f(\gamma(t)) \cdot \gamma'(t)\, dt \\
&\leq -p_1 p_3 \int_{t_1}^{t_2} \|\nabla f(\gamma(t))\|^2 \, dt.
\end{aligned}
\tag{2.24}
$$

Properties of $V(x,p)$.

Lemma 2.2.6 *For $x, y \in X$, we have $y \in V(x, (p_1, p_2, p_3))$ if and only if $x \in V(y, (1/p_2, 1/p_1, p_3))$.*

Proof. Since $p_1, p_2 > 0$, the inequality $p_1 \|x\| \leq \|y\| \leq p_2 \|x\|$ is equivalent to $\|y\| / p_2 \leq \|x\| \leq \|y\| / p_1$. And the inequality $p_3 \|x\| \|y\| \leq x \cdot y$ is symmetric in x and y. □

Lemma 2.2.7 *For $x \in X$, we have $x + \delta \in V(x, (p_1, p_2, p_3))$ for all $\delta \in X$ with $\|\delta\| \leq Q \|x\|$, where $Q = \min\{(1-p_1), (p_2-1), (1-{p_3}^2)^{1/2}\}$.*

Proof. The result is immediate for $x = 0$, so we will assume otherwise. Let $\delta \in X$ with $\|\delta\| \leq Q \|x\|$. Then

$$
p_1 \|x\| = \|x\| - (1-p_1)\|x\| \leq \|x\| - \|\delta\| \leq \|x + \delta\|
$$

and

$$
\|x + \delta\| \leq \|x\| + \|\delta\| \leq \|x\| + (p_2 - 1)\|x\| = p_2 \|x\| .
$$

Since $0 \leq Q < 1$, we have $\|x+\delta\| \geq (1-Q)\|x\| > 0$. Let $y \in X$ be the projection of x onto the line spanned by $x+\delta$, and let θ be the angle between x and $x+\delta$. Then

$$\begin{aligned}
\frac{x \cdot (x+\delta)}{\|x\|\,\|x+\delta\|} &= \cos\theta = \frac{\|y\|}{\|x\|} = \frac{(\|x\|^2 - \|y-x\|^2)^{1/2}}{\|x\|} \\
&\geq \frac{(\|x\|^2 - \|\delta\|^2)^{1/2}}{\|x\|} \\
&\geq \frac{(\|x\|^2 - (1-p_3{}^2)\,\|x\|^2)^{1/2}}{\|x\|} \\
&= p_3. \qquad \square
\end{aligned}$$

Lemma 2.2.8 *For $x \in X$, we have $x+\delta_1 \in V(x+\delta_2,(p_1,p_2,p_3))$ for all $\delta_1, \delta_2 \in X$ with $\|\delta_1\|, \|\delta_2\| \leq Q\|x\|$, where $Q = \min\{(1-p_1)/(1+p_1), (p_2-1)/(p_2+1), ((1-p_3)/2)^{1/2}\}$.*

Proof. The result is immediate for $x = 0$, so we will assume otherwise. Let $\delta_1, \delta_2 \in X$ with $\|\delta_1\|, \|\delta_2\| \leq Q\|x\|$. Then

$$\begin{aligned}
p_1\|x+\delta_2\| &\leq p_1(\|x\| + \|\delta_2\|) \leq p_1(1 + \frac{1-p_1}{1+p_1})\,\|x\| \\
&= (1 - \frac{1-p_1}{1+p_1})\,\|x\| \leq \|x\| - \|\delta_1\| \leq \|x+\delta_1\|
\end{aligned}$$

and

$$\begin{aligned}
\|x+\delta_1\| &\leq \|x\| + \|\delta_1\| \leq (1 + \frac{p_2-1}{p_2+1})\,\|x\| \\
&= p_2(1 - \frac{p_2-1}{p_2+1})\,\|x\| \leq p_2(\|x\| - \|\delta_2\|) \leq p_2\,\|x+\delta_2\|.
\end{aligned}$$

In the following, $i \in \{1,2\}$. Since $0 \leq Q < 1$, we have $\|x+\delta_i\| \geq (1-Q)\|x\| > 0$. Let $y_i \in X$ be the projection of x onto the line spanned by $x+\delta_i$, and let θ_i be the angle between x and $x + \delta_i$. Then

$$\begin{aligned}
\cos\theta_i &= \frac{\|y_i\|}{\|x\|} = \frac{(\|x\|^2 - \|y_i - x\|^2)^{1/2}}{\|x\|} \\
&\geq \frac{(\|x\|^2 - \|\delta_i\|^2)^{1/2}}{\|x\|} \\
&\geq \frac{(\|x\|^2 - \frac{1}{2}(1-p_3)\,\|x\|^2)^{1/2}}{\|x\|} \\
&= \left(\frac{p_3+1}{2}\right)^{1/2} > \left(\frac{1}{2}\right)^{1/2}.
\end{aligned}$$

It follows that $0 \leq \theta_i < \pi/4$. Now let θ be the angle between $x + \delta_1$ and $x + \delta_2$. Then

$$\begin{aligned}
\frac{(x+\delta_1)\cdot(x+\delta_2)}{\|x+\delta_1\|\,\|x+\delta_2\|} &= \cos\theta \\
&\geq \cos(\theta_1 + \theta_2) \\
&\geq \cos(2\max\{\theta_1, \theta_2\}) \\
&= 2\cos^2(\max\{\theta_1, \theta_2\}) - 1 \\
&\geq 2\left(\frac{p_3+1}{2}\right) - 1 \\
&= p_3. \qquad \square
\end{aligned}$$

Chapter 3

An algorithm using $O(n^2)$ storage

This chapter describes an implementation of Algorithm 2.2.1 in $\mathbb{R}^n$ that uses $O(n^2)$ storage. The implementation is called UMINH, for **u**nconstrained **min**imization with **H**essian. Its performance on a large set of test problems is presented and discussed.

3.1 Implementation

3.1.1 Search curve

At the point x_i, Algorithm 2.2.1 uses a curve γ_i to search for a point that reduces the value of the objective function. We now discuss how this curve is calculated.

In what follows, the space X will be $\mathbb{R}^n$. Let $A_i = -\nabla^2 f(x_i)$ and $b_i = -\nabla f(x_i)$. Let the spectral decomposition of A_i be denoted by $Q_i \Lambda_i {Q_i}^T$, where ${Q_i}^T Q_i = I$, $Q_i = [q_i^1, \ldots, q_i^n]$, and $\Lambda_i = \text{diag}(\lambda_i^1, \ldots, \lambda_i^n)$ with $\lambda_i^1 \leq \ldots \leq \lambda_i^n$. Then the solution to the ordinary differential equation defining γ_i is

$$\begin{aligned} \gamma_i(t) &= Q_i g(\Lambda_i, t) {Q_i}^T b_i && (3.1) \\ &= \sum_{j=1}^{n} (q_i^j \cdot b_i) g(\lambda_i^j, t) q_i^j, && (3.2) \end{aligned}$$

where

$$g(\lambda, t) = \begin{cases} \frac{1}{\lambda}(e^{\lambda t} - 1) & \text{for } \lambda \neq 0 \\ t & \text{for } \lambda = 0 \end{cases}.$$

The initial condition of the ordinary differential equation defining γ_i is that $\gamma_i'(0) = -\nabla f(x_i)$; that is, the search curve is initially tangent to the steepest-descent direction. Therefore when $\nabla f(x_i) \neq 0$, there is a $t > 0$ for which $f(x_i + \gamma_i(t)) < f(x_i)$.

If $q_i^j \cdot b_i = 0$ for all j, then $\gamma_i(t) = 0$ for $t \geq 0$. Otherwise, let p be the largest integer such that $q_i^j \cdot b_i \neq 0$. Then

$$\gamma_i(t) = \sum_{j=1}^{p} (q_i^j \cdot b_i) g(\lambda_i^j, t) q_i^j.$$

The eigenvalue λ_i^p determines the nature of the curve. We see from the formula for γ_i that when $\lambda_i^p \geq 0$ the curve is unbounded, and when $\lambda_i^p < 0$ the curve is bounded with

$$\lim_{t\to\infty} \gamma_i(t) = \sum_{j=1}^{p} (q_i^j \cdot b_i)(-1/\lambda_i^j) q_i^j.$$

If $\lambda_i^p < 0$ and A_i is invertible, this last equation may be equivalently expressed

$$\lim_{t\to\infty} \gamma_i(t) = -A_i^{-1} b_i,$$

where the right-hand side is just the Newton direction. Therefore when $\nabla^2 f(x_i)$ is positive definite, it follows that the search curve γ_i is bounded and its limit is the Newton direction. See Figure 3.1.

Ideally we would like the parameterization of the curve γ_i to be proportional to its arclength. For such a parameterization, $\|\gamma_i'\|$ would be constant. The parameterization above, which we term the t-parameterization, is far from this ideal since $\|\gamma_i'\| \to \infty$ when $\lambda_i^p \geq 0$ and $\|\gamma_i'\| \to 0$ when $\lambda_i^p < 0$. Unfortunately, parameterization

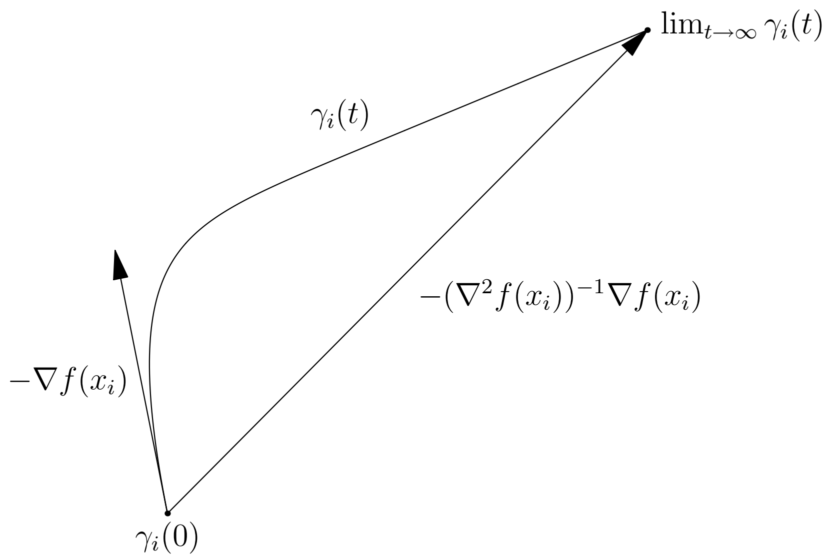

Figure 3.1: Search curve γ_i when $\nabla^2 f(x_i)$ is positive definite.

by arclength is not practically computable. Instead, we have chosen the parameterization

$$s = \begin{cases} \frac{1}{\lambda_i^p}(e^{\lambda_i^p t} - 1) & \text{for } \lambda_i^p \neq 0 \\ t & \text{for } \lambda_i^p = 0 \end{cases},$$

which we term the s-parameterization.

With the s-parameterization, $\|\gamma_i'\|$ is always bounded and bounded away from 0. This follows from the fact that the component of γ_i' associated with the eigenvector q_i^p is now constant and its other components are either constant or decrease in magnitude as s increases. When the curve is straight ($\lambda_i^1 = \cdots = \lambda_i^p$), the s-parameterization reduces to parameterization by arclength. With the t-parameterization, all curves have the parameter range $t \in [0, \infty[$. With the s-parameterization, unbounded curves ($\lambda_i^p \geq 0$) have the parameter range $s \in [0, \infty[$, but bounded curves ($\lambda_i^p < 0$) have the parameter range $s \in [0, s_{\max}]$, where $s_{\max} = -1/\lambda_i^p$ is the parameter for the limit point at the end of the curve.

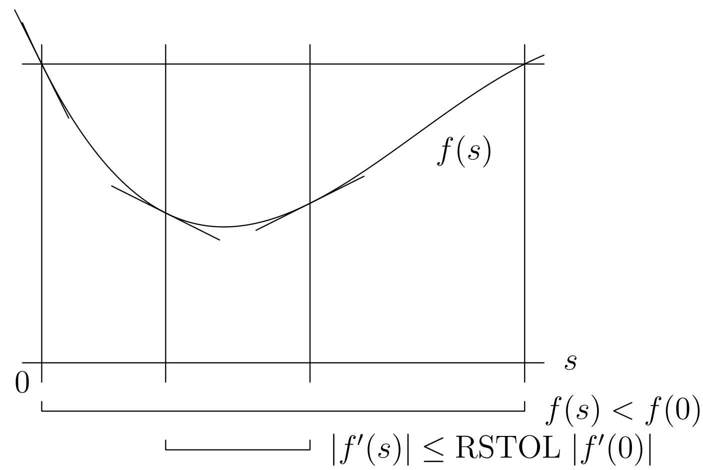

Figure 3.2: Search criteria.

3.1.2 Curve search

Once we have the curve $x_i + \gamma_i$, we seek a point along it that satisfies certain search criteria. This section describes how we perform the search along this curve for the new point.

Since we will be restricting our attention to the behavior of f along the curve, let us adopt the notation $f(s)$ for $f(x_i + \gamma_i(s))$ and $f'(s)$ for $\frac{d}{ds}f(x_i + \gamma_i(s))$. With this notation, our criteria for accepting a new point $x_i + \gamma_i(s)$ are

$$f(s) < f(0) \tag{C1}$$

$$|f'(s)| \leq \text{RSTOL}\, |f'(0)|, \tag{C2}$$

where $0 < \text{RSTOL} < 1$. The search criteria are illustrated in Figure 3.2.

Our first task in finding a point along $x_i + \gamma_i$ that satisfies the search criteria is to select an initial trial point. The scheme for selecting this point is an important factor in the performance of the algorithm, both its speed and robustness, and for this reason we experimented with a variety of different schemes. A good scheme will produce initial points that themselves satisfy the search criteria a large percentage of the time and that when they do not are not far from points that do.

Recall that the search curve is bounded when the parameterization eigenvalue $\lambda_i^p < 0$ and is unbounded when $\lambda_i^p \geq 0$. For a bounded curve, the point at the end of the curve has parameter $-1/\lambda_i^p$, and a step to this point is the Newton step. This point has long been recognized as a good initial trial point in other nonlinear optimization algorithms, especially close to the solution. We use this step as the initial trial point for bounded curves subject to some safeguards discussed below. For unbounded curves, we use the point with parameter $1/\lambda_i^p$ as the initial trial point, again with some safeguards (against for example $\lambda_i^p = 0$) discussed below.

The just-mentioned points are usually, but not always, good initial trial points. We need to constrain them somewhat as a safeguard for those cases in which they are not. For this purpose, at each iteration we calculate the infinity-norm length of the step taken. Then for each iteration after the first, we constrain the initial trial point to be no farther in infinity-norm distance from x_i than a constant times the infinity-norm length of the preceding step. All the infinity-norms are calculated in the current orthonormal eigenvector basis. At the first iteration, we do not have a preceding step and instead we use λ_i^p to constrain the initial trial point. When $|\lambda_i^p|$ is sufficiently small, we use the point with parameter 1 as the initial trial point.

Once we have an initial trial point, we evaluate the function f at the point. If criterion (C1) is met, we evaluate the derivative f'. If both criteria (C1) and (C2) are met, we return the point as the next point in the iteration; otherwise, we make additional trials. By construction, the search curve will always contain points that satisfy (C1), but it need not contain points that satisfy (C2). The search will either yield a point that satisfies at least (C1), or, if no such point is found after a maximum number of trials, will stop the algorithm and return an error condition.

We search for a point with a parameter in the search interval $[0, -1/\lambda_i^p]$ for a bounded curve or in the search interval $[0, \infty[$ for an unbounded curve. A trial point divides the search interval into two subintervals (if the curve is bounded and the trial point is at parameter $-1/\lambda_i^p$, then one of the subintervals is empty). Based upon information of the function at the trial point, either it is accepted as the next point in the iteration or one of the two subintervals is discarded and the trial point becomes the new upper bound or the new lower bound of the search interval.

The search proceeds in two phases. The first phase ends when: 1) a trial point is accepted, or 2) a trial point becomes the upper bound for the search interval, in which case the second phase begins, or 3) a maximum number of trials occurs with neither of the preceding outcomes, in which case either a point that satisfies (C1) is returned or, if no such point was found, an error condition is returned and the algorithm stops. The second phase begins with a bounded search interval. The derivative f' is always known at the search interval lower bound, but it is not always evaluated at the search interval upper bound. If it is known at the upper bound, then a safeguarded cubic interpolant is used to choose a trial point; otherwise the search interval midpoint is used. The function is then evaluated at the trial point, and the point is either accepted or used to discard part of the search interval. The process continues until either 1) a trial point is accepted or 2) a maximum number of trials occurs, in which case, as before, either a point that satisfies (C1) is returned or, if no such point was found, an error condition is returned and the algorithm stops.

3.2 Performance of UMINH

In this section, we examine the performance of UMINH on a large set of test problems and compare its performance with that of two leading alternatives.

3.2.1 Test problems

We use a set of test problems taken from the Buckley test set [4]. This is a compilation from many sources of 417 test problems from 83 test functions; some have been developed by researchers for the testing of algorithms, others are real problems from practitioners. Here we use the term *test function* to mean a functional form that may have a variable size or have parameters that may be varied, and we use the term *test problem* to mean a test function with specified size and parameters and a specified initial point.

From the Buckley test set we selected for implementation 142 test problems from all 83 test functions. The problems ranged in size from $n = 2$ to $n = 1000$. For a gallery of some of the two-dimensional problems, see Figures 3.3–3.7. Each figure has a contour plot of the objective function with a shaded area for the region where the Hessian is positive definite. The black dot on the highest contour is the initial point. Black dots in the interior are local minimizers, and grey dots are saddle points.

Each of the 142 test problems was implemented in Fortran with the aid of Mathematica [25] and ADIFOR [2], [3]. Mathematica was used to generate the objective functions and gradients for all the problems and the Hessians for problems of size $n \leq 20$. ADIFOR was used to generate the Hessian-vector products for all problems and the Hessians for problems of size $n > 20$.

We found the limit value of the objective function for all problems and the limit point associated with the initial point for 99 of the 142 problems. To do this, we first solved the problems in double precision with three routines: 1) UMINH, 2) the NAG library routine E04LBF [18], a modified-Newton line-search method, and 3) the AT&T PORT library routine DMNH [8], a trust-region method. If the local minimizers found by each of these methods were close together, they were then used as the starting points for the solution of the problem in quadruple precision. If different local minimizers were found, a highly accurate ordinary differential equation solver was used to determine which local minimizers were close to the limit point associated with the initial point, and those local minimizers were then used as starting points for the solution of the problem in quadruple precision. The minimizers found by UMINH were improved by a quadruple-precision Newton method, and those found by E04LBF and DMNH were improved by quadruple-precision versions of these respective routines. The quadruple-precision results from the different methods always agreed to the double-precision accuracy used in the numerical tests.

Further information on the test problems, including information on how to use them, is included in Appendix A.

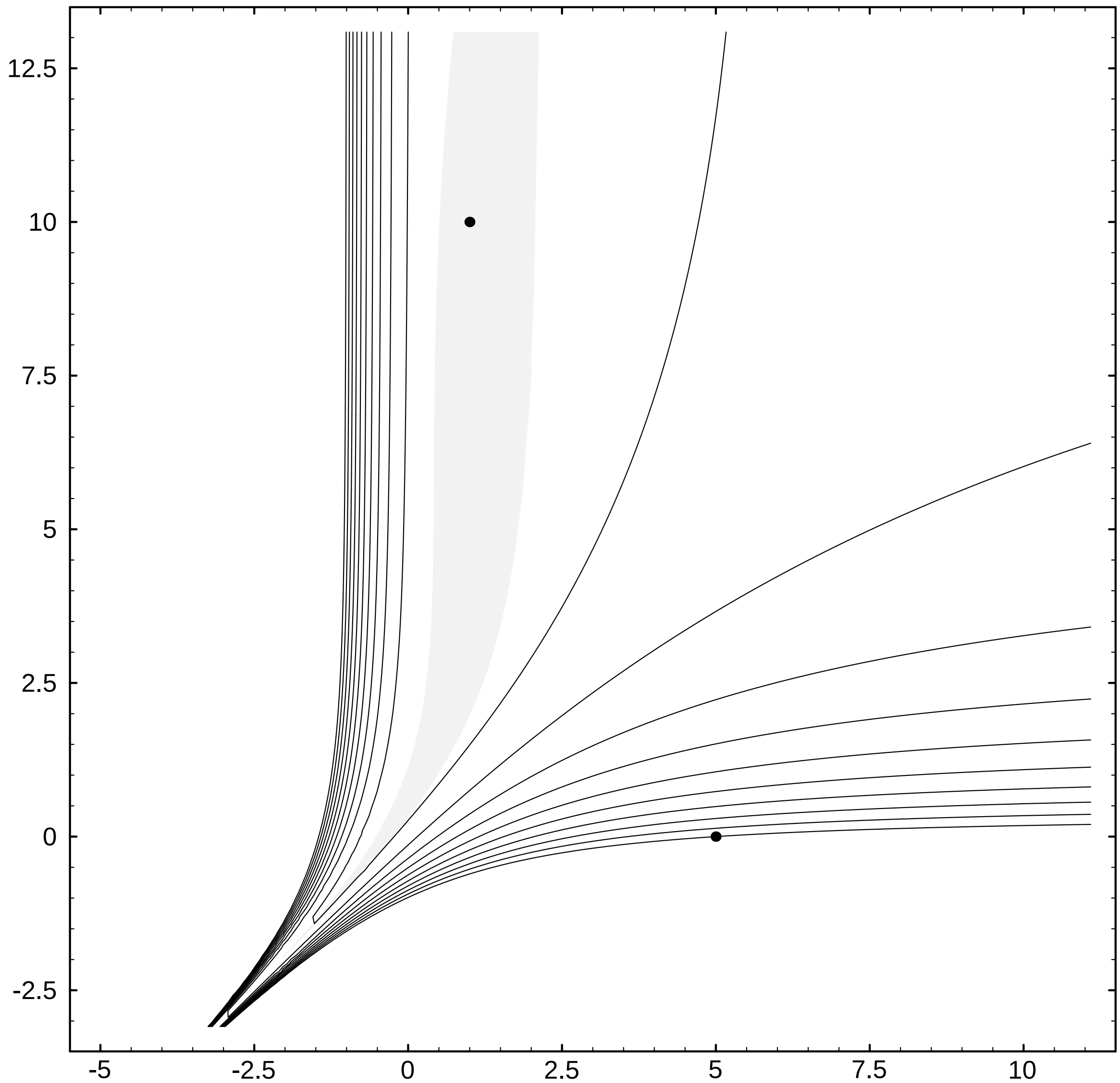

Figure 3.3: The test problem BOX662HL.

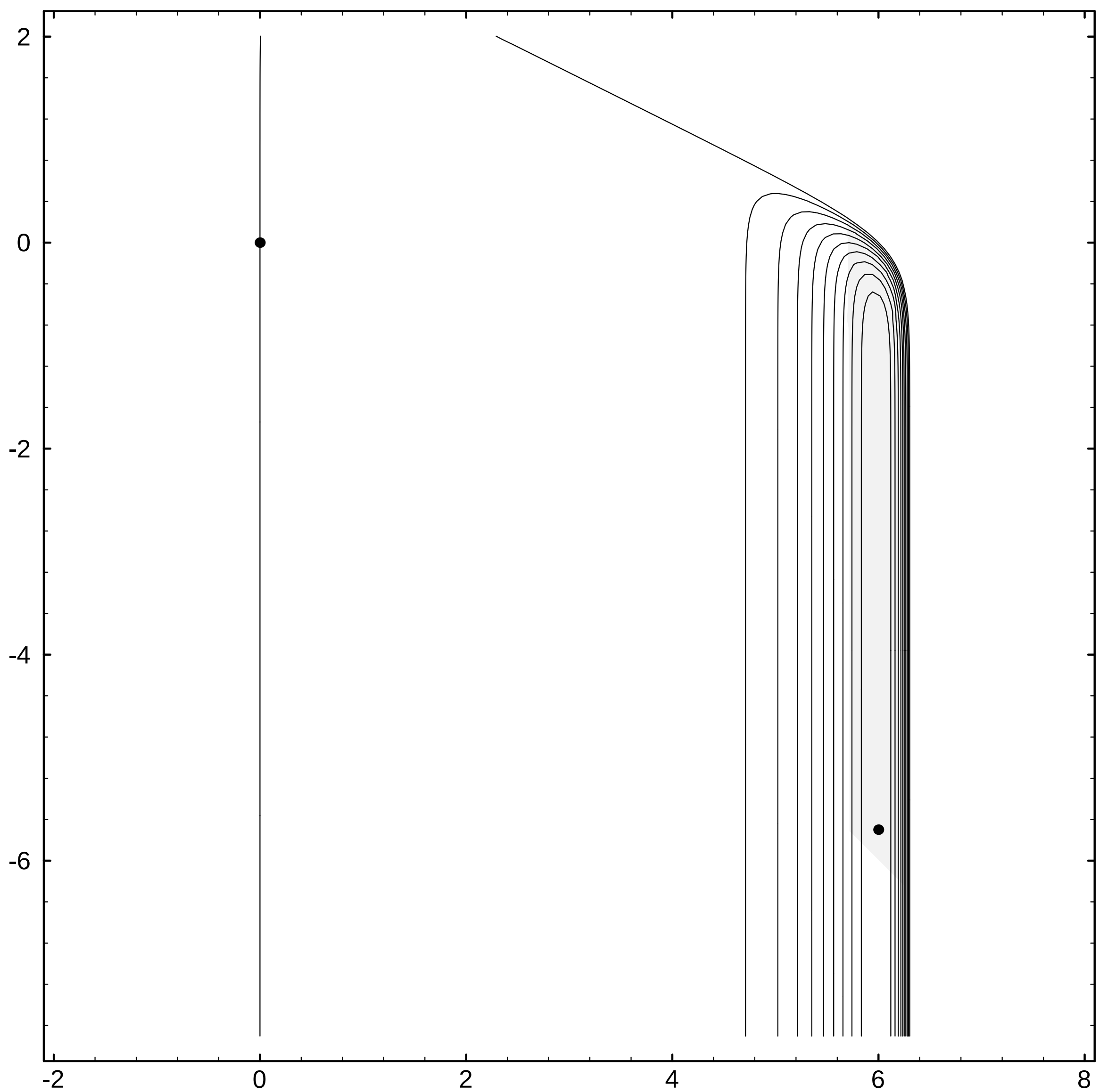

Figure 3.4: The test problem BROWNB, log-log plot.

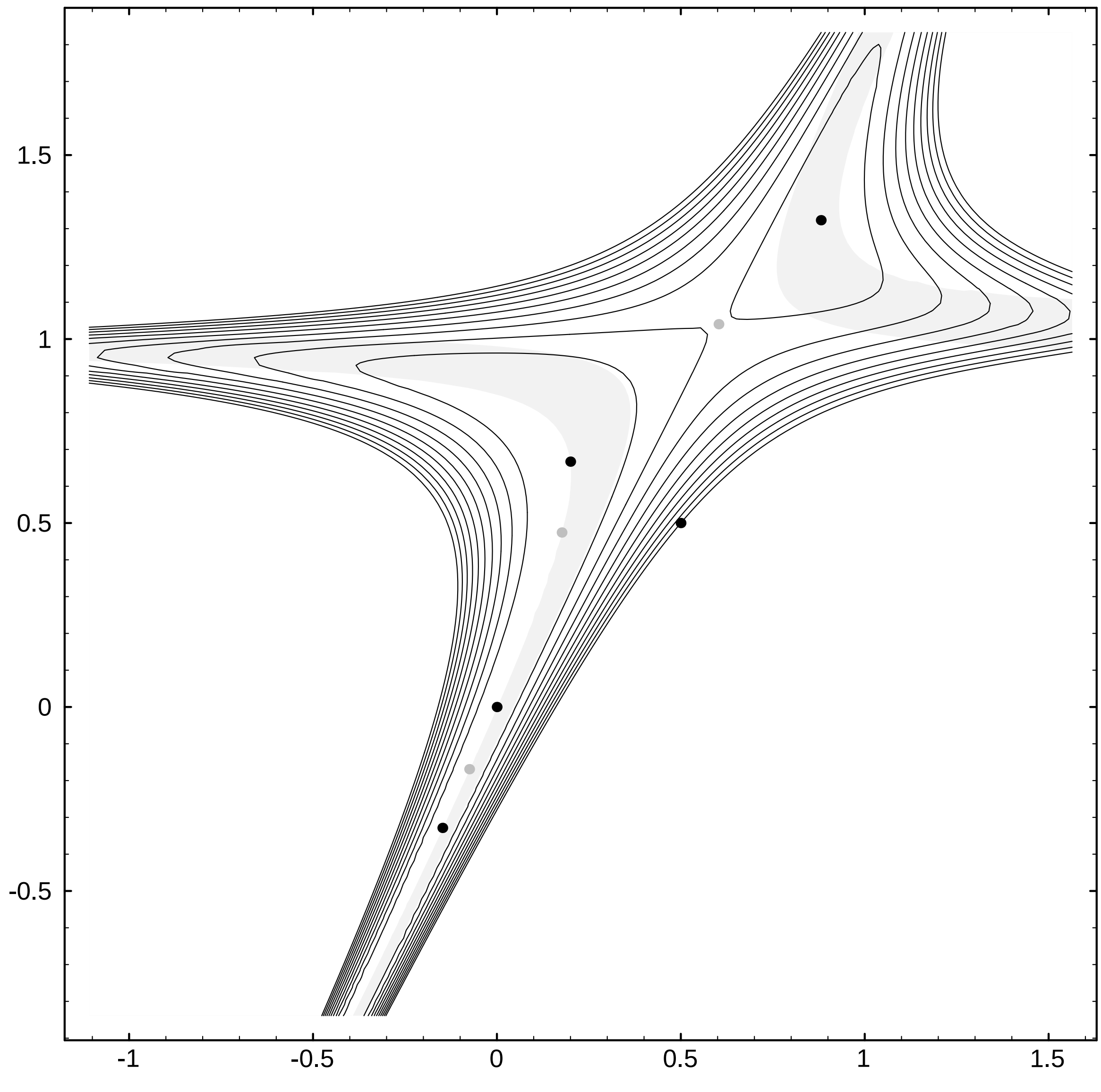

Figure 3.5: The test problem GOTTFR.

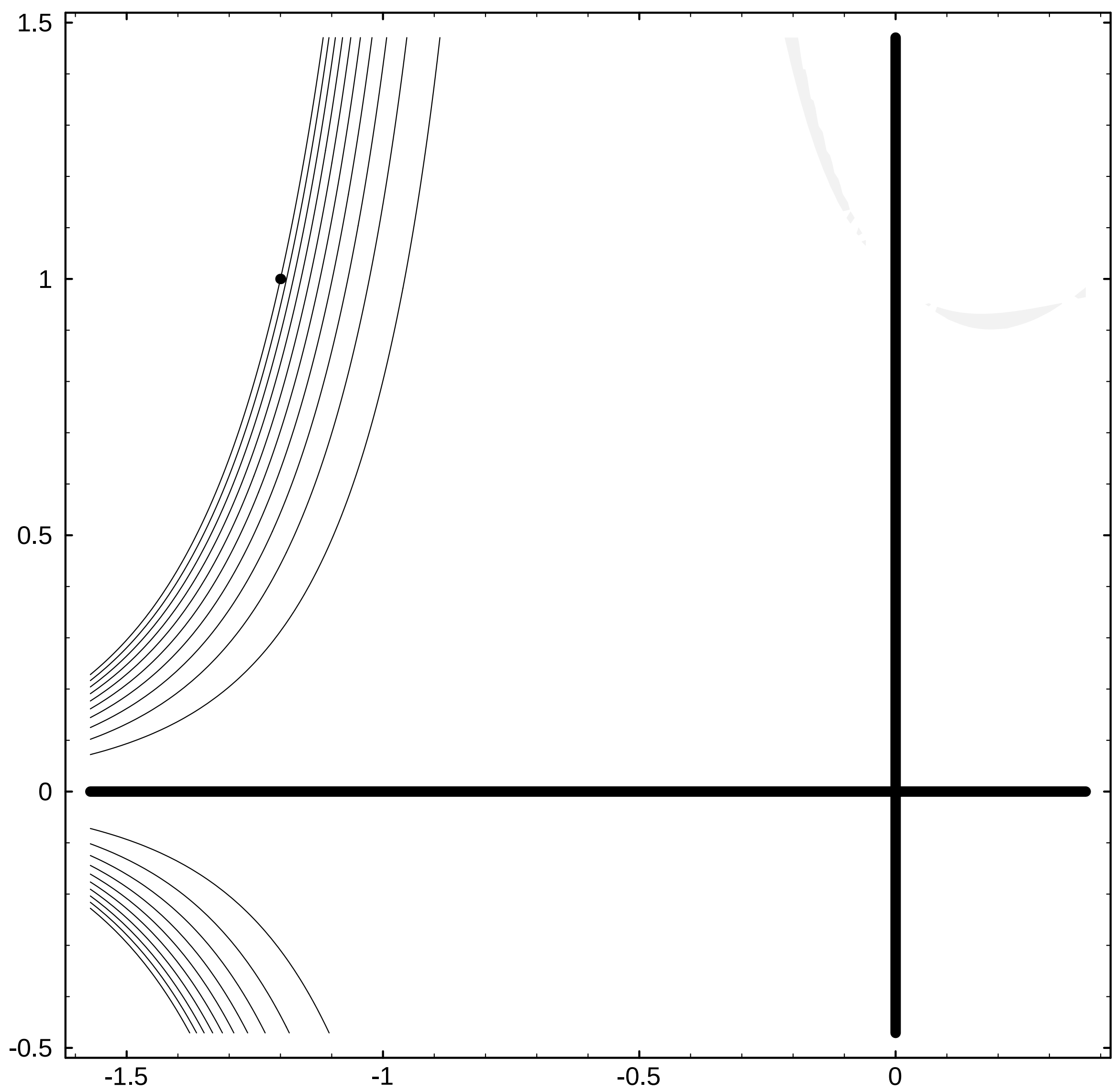

Figure 3.6: The test problem HIMM27.

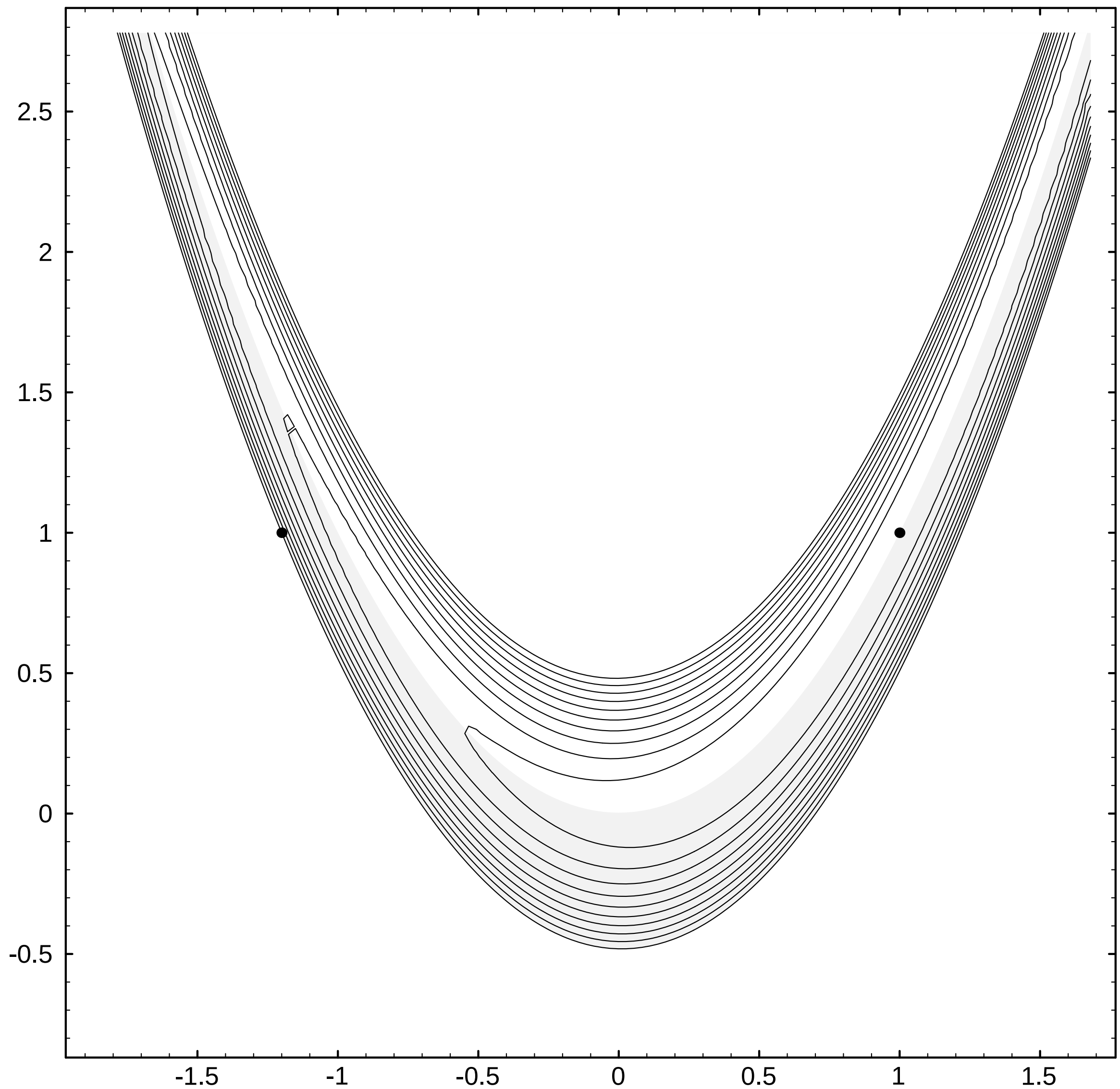

Figure 3.7: The test problem ROSENB.

3.2.2 Numerical results

The numerical testing was performed on a Sun SPARCstation 20 running SunOS release 4.1.3_U1. The routines and test problems were all compiled with the Sun Fortran compiler version SC3.0.1.

In Figure 3.8 we show how UMINH performed on the two-dimensional test problem BEAL58KO introduced in Section 2.1. In this contour plot, the shaded area is the region where the Hessian is positive definite, the smooth curve connecting the initial point $x_0 = (0, 0)$ with a minimizer is the integral curve of $-\nabla f$, and the small dots on the piecewise-linear curve are the iterates of UMINH.

Routines used

We compared the performance of our routine UMINH against that of two leading commercial routines that also use first and second derivatives. For a line-search method, we used the routine E04LBF from the NAG library [18]. For a trust-region method, we used the routine DMNH from the AT&T PORT library [8]. The 142 test problems were run on each of these three routines, using the routines' default or suggested parameter values.

Performance measures

With the limit value f_* of the objective function known for each test problem, we were able to use the reduction in error

$$\epsilon_i = \left| \frac{f_i - f_*}{f_0 - f_*} \right|$$

as a uniform measure of the routines' convergence. We recorded the number of iterations, function evaluations, and gradient evaluations required by each routine to achieve tolerance levels $\epsilon_i = 10^{-3}$, 10^{-6}, 10^{-9}, and 10^{-12}.

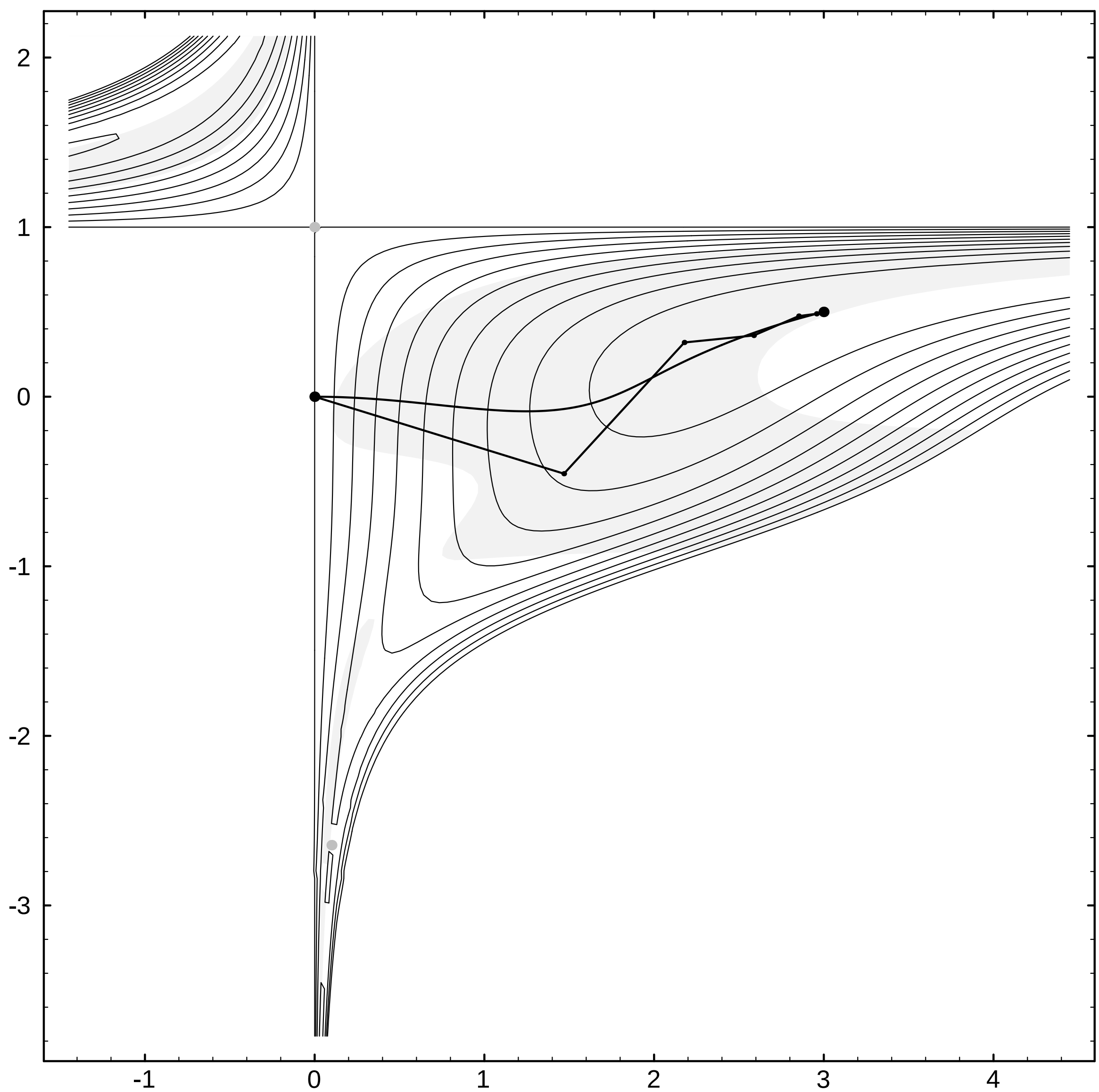

Figure 3.8: The iterates of UMINH for the test problem BEAL58KO.

Problem	Routine		
	UMINH	E04LBF	DMNH
ARTRIG10		10^{-8}	
BIGGS6			cycled
BROY7D	10^{-14}		
BRWNL100		10^{-17}	
CHEBYQ10		10^{-10}	10^{-10}
MORBV998		10^{-6}	
MSQRTB49		10^{-5}	
NMSUR484		10^{-3}	
NMSURF64		10^{-4}	
OSBRN1			cycled
OSBRN2		10^{-10}	
PEN2GM1		10^{-10}	
PEN2GM2	10^{-6}	10^{-10}	
POWB1000		10^{-1}	
POWBS500		cycled	10^{-20}
POWBSC50		cycled	10^{-20}
POWSSQ	10^{-10}		
QUARTC		10^{-9}	
SNMSR484		diverged	
SNMSUR64		diverged	
TRLN100		1	
TRLN1000		1	

Table 3.1: Order of magnitude of $\|\nabla f_{\text{last}}\| \,/\, \|\nabla f_0\|$ at solution failures.

Robustness

Of the 142 test problems, 22 caused at least one of the three routines to fail. Table 3.1 shows those instances where the routines failed to reduce ϵ_i to 10^{-12}. In the case of the Powell badly-scaled problem POWB1000, the solution tolerance was set to 10^{-9} since none of the three routines was able to solve this problem to 10^{-12}. The ratios of the gradient norms given in the table show that in some instances a routine would find a critical point, but not the one associated with the initial point.

Run with its default parameters, UMINH correctly solved 139 of the 142 test problems. With its default parameters, it found the wrong local minimizers for three problems, but it found the correct local minimizers when it was set to perform a more careful search. DMNH had the second lowest number of failures with 5. E04LBF had the highest number of failures with 18.

Although it is common practice to compare the performance of optimization algorithms on a standard set of test problems as we have done, their performance on real problems may differ. While the test set included some real problems, many others were designed to be particularly difficult to solve. Some had functions designed to be hard to solve, such as the Powell badly-scaled problem. Others had bad initial points. Since real problems rarely exhibit the extreme pathologies present in the test problems and frequently have good initial points, the routines are likely to be more robust in practice than they were in our tests.

Speed

The 142 test problems were run on each of the three routines, using the routines' default or suggested parameter values. For each problem and routine, we recorded the number of iterations, function evaluations, and gradient evaluations required for the convergence measure ϵ_i to reach the tolerance levels 10^{-3}, 10^{-6}, 10^{-9}, and 10^{-12}. The performance of each routine on each problem is given in Appendix C. The results are summarized in Tables 3.2–3.4. The tables contain for each performance measure, each tolerance level, and each routine, the number of test problems on which the routine finished first, second, and third among the three routines and the number of problems that it failed to solve. Ties occurred frequently. Since none of the three routines was able to solve the Powell badly-scaled problem POWB1000 to a tolerance of 10^{-12}, it was omitted from testing at this tolerance.

UMINH had the highest number of first-place finishes in each of the three performance measures. E04LBF had the second highest number of first-place finishes for each of the three measures. DMNH had the lowest number of first-place finishes. The performance of the routines was consistent across the four tolerance levels.

Tolerance	Routine	Test Problem Place Finishes			
		First	Second	Third	Failed
10^{-3}					
	UMINH	116	22	2	2
	E04LBF	109	15	9	9
	DMNH	74	58	7	3
10^{-6}					
	UMINH	113	24	2	3
	E04LBF	104	18	7	13
	DMNH	60	65	14	3
10^{-9}					
	UMINH	110	28	1	3
	E04LBF	89	30	6	17
	DMNH	55	66	18	3
10^{-12}					
	UMINH	113	25	0	3
	E04LBF	90	24	10	17
	DMNH	57	62	17	5

Table 3.2: Performance summary for number of iterations.

Tolerance	Routine	Test Problem Place Finishes			
		First	Second	Third	Failed
10^{-3}					
	UMINH	114	22	4	2
	E04LBF	93	22	18	9
	DMNH	52	75	12	3
10^{-6}					
	UMINH	111	25	3	3
	E04LBF	90	18	21	13
	DMNH	45	76	18	3
10^{-9}					
	UMINH	107	29	3	3
	E04LBF	83	19	23	17
	DMNH	45	78	16	3
10^{-12}					
	UMINH	107	28	3	3
	E04LBF	83	19	22	17
	DMNH	47	71	18	5

Table 3.3: Performance summary for number of function evaluations.

Tolerance	Routine	Test Problem Place Finishes			
		First	Second	Third	Failed
10^{-3}					
	UMINH	110	28	2	2
	E04LBF	81	25	27	9
	DMNH	88	46	5	3
10^{-6}					
	UMINH	111	27	1	3
	E04LBF	80	21	28	13
	DMNH	73	61	5	3
10^{-9}					
	UMINH	110	28	1	3
	E04LBF	75	18	32	17
	DMNH	65	71	3	3
10^{-12}					
	UMINH	112	26	0	3
	E04LBF	74	16	34	17
	DMNH	66	67	3	5

Table 3.4: Performance summary for number of gradient evaluations.

Chapter 4

An algorithm using $O(n)$ storage

The algorithms discussed in Chapter 3 all require $O(n^2)$ storage. For a given amount of storage, this requirement places a limit on the size of the problems that may be solved. To solve larger problems, we need algorithms that require less storage. This chapter describes an implementation of Algorithm 2.2.1 in $\mathbb{R}^n$ that uses $O(n)$ storage. The implementation is called UMINHV, for **u**nconstrained **min**imization with **H**essian-**v**ector products. Its performance on a large set of test problems is presented and discussed.

4.1 Implementation

4.1.1 Search subspace

Recall that our aim is to approximate the integral curve of $-\nabla f$ (Section 2.2.1). If the integral curve lies in a low-dimensional submanifold, our approximation will only require information of the Hessian on a low-dimensional subspace. But regardless of the nature of the integral curve, for large problems we have storage only for information of the Hessian on a subspace. The central issue is what subspace to use.

Before we examine the issue of the choice of subspace, let us examine how the search curve will be calculated for a given subspace. In what follows, let $A_i = -\nabla^2 f(x_i) \in \mathbb{R}^{n\times n}$ and $b_i = -\nabla f(x_i) \in \mathbb{R}^n$. Let $Q_i \in \mathbb{R}^{n\times m}$, where $m \leq n$ and

${Q_i}^T Q_i = I$. The columns of Q_i then form an orthonormal basis for the m-dimensional subspace range(Q_i). There is a canonical correspondence between range(Q_i) and $\mathbb{R}^m$, namely ${Q_i}^T$: range(Q_i) $\to \mathbb{R}^m$, $Q_i : \mathbb{R}^m \to$ range(Q_i). Using this correspondence, we define $B_i = {Q_i}^T A_i Q_i \in \mathbb{R}^{m\times m}$, which is readily seen to be a self-adjoint linear operator. Let the spectral decomposition of B_i be denoted by $S_i \Theta_i {S_i}^T$, where ${S_i}^T S_i = I$ and $\Theta_i = \mathrm{diag}(\theta_i^1, \ldots, \theta_i^m)$ with $\theta_i^1 \leq \ldots \leq \theta_i^m$. The vectors in range($Q_i$) that correspond to the eigenvectors of B_i are called *Ritz vectors* and will be denoted by $[y_i^1, \ldots, y_i^m] = Y_i = Q_i S_i$. The diagonal elements of Θ_i are called *Ritz values*, and the (θ_i^j, y_i^j) are called *Ritz pairs*. If range(Q_i) $= \mathbb{R}^n$, then the Ritz pairs are simply the eigenpairs of A_i, and if range(Q_i) $\subsetneq \mathbb{R}^n$, then the Ritz pairs are approximations of eigenpairs.

Recall that the search curve for UMINH was defined using the eigenpairs of A_i (3.1), (3.2). The search curve for UMINHV is defined by simply substituting Ritz pairs for eigenpairs,

$$\gamma_i(t) = Y_i g(\Theta_i, t) {Y_i}^T b_i \tag{4.1}$$

$$= \sum_{j=1}^{m} (y_i^j \cdot b_i) g(\theta_i^j, t) y_i^j, \tag{4.2}$$

where as before

$$g(\theta, t) = \begin{cases} \frac{1}{\theta}(e^{\theta t} - 1) & \text{for } \theta \neq 0 \\ t & \text{for } \theta = 0 \end{cases}.$$

The entire discussion of the search curve for UMINH—when it is bounded and unbounded, its parameterization, and so on—carries over to the search curve for UMINHV. In fact, exactly the same code is used to calculate and search along the curves in both routines.

Initially, $\gamma_i'(0) = Y_i {Y_i}^T(-\nabla f(x_i))$, that is, the search curve is initially tangent to the orthogonal projection of the negative gradient into range(Q_i). Therefore when $\nabla f(x_i) \neq 0$ and $\nabla f(x_i) \not\perp$ range(Q_i), there is a $t > 0$ for which $f(x_i + \gamma_i(t)) < f(x_i)$. Hence a minimal requirement for range(Q_i) is that it not be perpendicular to $\nabla f(x_i)$.

In order for $\gamma_i'(0) = -\nabla f(x_i)$, we would need the stronger requirement that $\nabla f(x_i) \in \text{range}(Q_i)$.

When A_i is negative definite, the search curve of UMINH was shown to be bounded, with its limit being the Newton direction, $-A_i^{-1}b_i$. When the subspace range(Q_i) contains the Newton direction, it turns out that, despite the Ritz approximation, the search curve of UMINHV maintains these properties.

Lemma 4.1.1 *Let A_i, b_i, Q_i, and γ_i be as above. If A_i is negative definite and $-A_i^{-1}b_i \in$ range(Q_i), then γ_i is bounded and* $\lim_{t\to\infty} \gamma_i(t) = -A_i^{-1}b_i$.

Proof. If A_i is negative definite, then $B_i = Q_i^T A_i Q_i$ is likewise and hence is invertible. From the formula for γ_i (4.1), we get that

$$\lim_{t\to\infty} \gamma_i(t) = -Q_i(Q_i^T A_i Q_i)^{-1} Q_i^T b_i.$$

We have assumed

$$Q_i Q_i^T A_i^{-1} b_i = A_i^{-1} b_i$$

and hence

$$\begin{aligned} Q_i^T b_i &= Q_i^T A_i Q_i Q_i^T A_i^{-1} b_i \\ -Q_i(Q_i^T A_i Q_i)^{-1} Q_i^T b_i &= -Q_i Q_i^T A_i^{-1} b_i \\ &= -A_i^{-1} b_i. \qquad \square \end{aligned}$$

Because of the importance of the Newton direction in other optimization algorithms, we decided to see how well UMINHV would perform when at each point it used the subspace span$\{b_i, A_i^+ b_i\}$, where A_i^+ denotes the pseudo-inverse of A_i. When A_i is negative definite, $-A_i^+ b_i = -A_i^{-1} b_i$, the Newton direction. UMINHV and UMINH were run on 75 of the 78 test problems described in Section 4.2.1, those that could be run quickly. For each routine, we recorded the number of iterations, function evaluations, and gradient evaluations required to solve each problem to several tolerance levels. UMINH failed to solve 2 of the problems. UMINHV failed to solve these 2 problems and 3 others, and it solved 2 others to the 10^{-6} tolerance but not to 10^{-9} or 10^{-12}. Thus there were 68 problems that both routines solved to all tolerances. In number of iterations, UMINHV did better than UMINH on 10 problems, the same on 49 problems, and worse on 9 problems. In number of function evaluations, UMINHV did better on 13 problems, the same on 47 problems, and worse on 8 problems. In number of gradient evaluations, UMINHV did better on 12 problems, the same on 49 problems, and worse on 7 problems.

The conclusion to be drawn from this experiment is that UMINHV can perform quite well with a very small search subspace. With the 2-dimensional subspace in the experiment, its performance was comparable to that of UMINH, which uses the entire space. Unfortunately, ${A_i}^+ b_i$ is very expensive to compute, requiring an amount of computation and storage comparable to that used by UMINH. Nevertheless, it provides a goal to aim for in more practical search subspaces.

4.1.2 Krylov search subspace

UMINHV was shown to perform quite well with the search subspace $\mathrm{span}\{b_i, {A_i}^+ b_i\}$. With the gradient available, it is easy to include b_i in the search subspace. With the Hessian available only through matrix-vector products and with only $O(n)$ storage, we now seek to generate efficiently a subspace nearly parallel to ${A_i}^+ b_i$. The *Krylov subspaces*

$$K(A_i, b_i, k) \equiv \mathrm{span}\{b_i, A_i, \dots, {A_i}^{k-1} b_i\}$$

have long been effectively applied to approximating ${A_i}^+ b_i$. These subspaces also have the desirable property that their Ritz pairs approximate the eigenpairs associated with the extreme eigenvalues of A_i. The significance of this property can be seen from the expression of the search curve of UMINH in terms of the eigenpairs of A_i (3.2). The function $g(\lambda, t)$ is a monotonically increasing function of λ for all $t > 0$; therefore, the terms in (3.2) with the largest values of $g(\lambda, t)$ are those associated with the largest eigenvalues of A_i.

There is an algorithm by Lanczos for efficiently generating orthonormal bases for the Krylov subspaces, [13] and [10, Chapter 9]. The Lanczos bases have the desirable property of producing tridiagonal reduced matrices. If the columns of Q_i are the Lanczos vectors that form an orthonormal basis for $K(A_i, b_i, k)$, then ${Q_i}^T A_i Q_i \approx T_i$, a tridiagonal matrix; furthermore, the elements of T_i are generated in the calculation of the Lanczos vectors.

While the Lanczos algorithm has many advantages in exact arithmetic, it requires careful implementation in finite-precision arithmetic because of the Lanczos vectors' loss of orthogonality. Simon [23], [24] has shown that the algorithm remains effective when the Lanczos vectors maintain at least semiorthogonality; that is, when $|q_i^{j_1} \cdot q_i^{j_2}| \leq \mathrm{EPS}^{\frac{1}{2}}$, where $j_1 \neq j_2$ and EPS is the relative machine precision. In the same papers he has also shown how to monitor efficiently the vectors' orthogonality and how to maintain semiorthogonality by reorthogonalizing the newest vector against some of the earlier vectors, as necessary.

With this approach, semiorthogonality can be maintained with little additional work. In our implementation, we use the Kahan algorithm that accurately orthogonalizes a vector against a subspace (see [21, Section 6.9]) to orthogonalize each new vector against the two preceding vectors; we monitor the new vector's orthogonality with respect to all previous vectors and reorthogonalize, as necessary, to maintain semiorthogonality.

Finally, we need a way to measure how nearly parallel ${A_i}^+ b_i$ is to $K(A_i, b_i, k)$ to know when to stop expanding the search subspace. We use a simple formula by Paige and Saunders [19] that calculates

$$\min_{y \in K(A_i, b_i, k)} \|A_i y - b_i\|$$

at each k and requires only a few flops.

Combining the above ideas we get the following.

Algorithm for calculating the Ritz pairs of the Krylov search subspace.

Apply the Lanczos algorithm with partial reorthogonalization until

$$\min_{y \in K(A_i, b_i, k)} \|A_i y - b_i\| \quad \leq \quad \text{RTOL}\, \|b_i\|$$

or until

$$k \quad = \quad \text{LMAX}$$

to generate a matrix of Lanczos vectors Q_i and its corresponding tridiagonal matrix T_i, where $T_i \approx {Q_i}^T A_i Q_i$.

Calculate the spectral decomposition $T_i = S_i \Theta_i {S_i}^T$.

Calculate the Ritz vectors $Y_i = Q_i S_i$.

The generation of the Ritz pairs requires only $O(n)$ flops and $O(n)$ storage. Aside from the LMAX n storage for the Ritz vectors and the LMAX storage for the Ritz values, it needs only $n + \text{LMAX}^2 + 2$ LMAX additional storage, where for large problems LMAX $\ll n$. Our default values are RTOL $= \frac{1}{8}$ and LMAX $= 16$.

The above algorithm, with the default values for RTOL and LMAX, was run on the 78 test problems described in Section 4.2.1, and for each problem at each step the dimension of the search subspace used was recorded. Table 4.1 shows for each problem the minimum, the average, and the maximum dimensions of the subspace used. The table is sorted in increasing average dimension, with the three values ranging from 1 to 16. Of the average dimensions, the median is 3.61 and the average is 5.69. On average, fairly small search subspaces were used.

Table 4.1: Search subspace dimensions

Function	Problem	Size	Min.	Ave.	Max
BRWNAL	BRWNL100	100	1	1.00	1
MANCIN	MANCIN50	50	1	1.00	1
PENAL1	PEN1LN1	50	1	1.00	1
PENAL1	PEN1LN2	100	1	1.00	1
PENAL1	PEN1LN3	1000	1	1.00	1
VARDIM	VARDIM	10	1	1.00	1
VARDIM	VARDM100	100	1	1.00	1
XTX	XTX16	16	1	1.00	1
POWBSC	POWBS500	500	1	1.22	2
DIX7DG	DIX7DGA	15	1	1.30	3
POWBSC	POWB1000	1000	1	1.33	2
NONDIA	NONDIA20	20	1	1.49	2
MORCIN	MORCIN10	10	1	1.50	2
NONDIA	NONDI500	500	1	1.51	2
NONDIA	NOND1000	1000	1	1.51	2
BRWNAL	BRWNAL10	10	1	1.57	2
EXTRSN	EXTRAR50	50	1	1.60	2
EXTRSN	EXTRA100	100	1	1.60	2
EXTRSN	EXTR1000	1000	1	1.60	2
BRYBND	BRYBND18	100	1	1.67	3
BRYBND	BRYBND	10	1	1.75	3
TDQUAD	TDQ500	500	1	1.75	4
TDQUAD	TDQ1000	1000	1	1.75	4
QUARTC	QUARTC	25	2	2.00	2
ENGVL1	ENGVL1B6	1000	1	2.11	4
ENGVL1	ENGVL1B4	100	1	2.30	4
ENGVL1	ENGVL1B2	10	1	2.33	4
TDQUAD	TDQ10	10	2	2.67	4
POWER	POWER75	75	1	2.68	4
FRDRTH	FRDRTHB7	1000	1	2.82	5
FRDRTH	FRDRTHB3	50	1	2.85	5
WOODS	WOODS80	80	1	2.95	4
FRDRTH	FRDRTHB4	100	1	3.00	5
PENAL3	PENL3GM3	50	1	3.27	8
PENAL3	PENL3GM5	1000	1	3.44	11
HILBRT	HILBRT12	12	1	3.50	6
PENAL3	PENL3GM4	100	1	3.50	9
PWSING	PWSING60	60	1	3.61	4
PWSING	PWSIN100	100	1	3.61	4
PWSING	PWSI1000	1000	1	3.61	4
CHEBYQ	CHEBYQ10	10	2	4.54	7
TRIDIA	TRIDIA10	10	5	5.00	5
ARTRIG	ARTRIG10	10	2	5.33	9
DIXON	DIXON	10	2	5.50	9
SCHMVT	SCHV1000	1000	2	5.57	8
SCHMVT	SCHMV500	500	2	5.71	8
CHNRSN	CHNRSH10	10	2	5.81	8
CRGLVY	CRGLVY10	10	1	5.84	9
BRYTRI	BRYTRI6	20	4	5.86	8
BRYTRI	BRYTRI10	600	5	6.14	8
TOINT	PSPTOINT	50	2	6.33	11
WATSON	WATSON12	12	1	6.33	12
SCHMVT	SCHMVT50	50	4	6.38	8
ROSENB	SHNRSN10	10	1	6.64	10
PENAL2	PEN2GM1	50	4	7.10	12
OSBRN2	OSBRN2	11	2	7.62	11
CRGLVY	CRGLY500	500	1	7.76	16
CRGLVY	CRGL1000	1000	1	7.86	16
PENAL2	PEN2GM2	100	2	8.00	16
GENRSN	GENT50A	50	1	9.28	16
POWBSC	POWBSC50	50	1	9.64	16
TRIDIA	TRLN100	100	6	10.20	13
TRIGTO	TRIGT50	50	6	10.46	16
GENRSN	GENT500A	500	1	10.65	16
GENRSN	GENT1000	1000	1	10.75	16
MNSRF1	NMSURF64	36	4	11.35	16
FRANK	FRANK12	12	12	12.00	12
TRIGTO	TRIGT100	100	6	12.16	16
BROY7D	BROY7D	60	4	12.61	16
VAROSB	VAROSBG1	50	5	13.05	16
SQRTMX	MSQRTB49	49	5	13.47	16
MNSRF2	SNMSUR64	36	4	13.80	16
TRIDIA	TRLN1000	1000	5	15.10	16
MNSRF2	SNMSR484	400	7	15.56	16
VAROSB	VAROSBG2	100	5	15.67	16
MNSRF1	NMSUR484	400	7	15.75	16
MOREBV	MOREBV18	18	12	15.84	16
MOREBV	MORBV998	998	16	16.00	16

4.1.3 Information reuse

The Krylov search subspace in the preceding subsection uses at each step only fresh information of the current Hessian through matrix-vector products. An alternative approach is to approximate the Hessian at the current step based upon a model that reuses information gathered from the Hessian at preceding steps. For the information from the prior steps to be useful, the good search subspaces and the Hessian would have to change little from step to step. There is also a trade-off between the computational savings achieved at each iteration by reusing information and the increase in the total number of iterations due to the degradation in the quality of the information used. Nevertheless, modeling the Hessian and reusing information have been effective techniques in other methods and may be effective here.

A model-based approach can be applied in many ways. One way is simply to extend the method of the preceding subsection by using a model to precondition the Lanczos iteration. Iterative methods for calculating ${A_i}^+b_i$ can be substantially accelerated through effective preconditioning. In our application, this would mean less computation required at each step to achieve the same level of parallelness between the search subspace and ${A_i}^+b_i$. A model could also be used in a completely different manner. At each step the algorithm could produce a search subspace Y_i and an approximation to A_iY_i based upon old and new information. From there it is straightforward to calculate the Ritz pairs and the search curve.

The many ways to apply modeling and the reuse of information in the algorithm is an area for future work.

4.2 Performance of UMINHV

This section examines the performance of UMINHV on a large set of test problems and compares this performance with that of two leading alternatives.

4.2.1 Test problems

In Section 3.2.1 we discussed our implementation of 142 test problems from all 83 test functions of the Buckley test set [4]. Here we are using the term *test function* to mean a functional form that may have a variable size or have parameters that may be varied and the term *test problem* to mean a test function with specified size and parameters and a specified initial point. For the testing of UMINHV, we restricted our attention to the 42 test functions that had test problems of size $n \geq 10$. For each of the 42 test functions that had test problems of several sizes, one test problem was chosen from each of the size ranges 10–99, 100–999, and 1000 that contained a test problem. This resulted in 78 test problems, 42 with size $10 \leq n \leq 99$, 23 with size $100 \leq n \leq 999$, and 13 with size $n = 1000$.

Further information on the test problems, including information on how to use them, is included in Appendix A.

4.2.2 Numerical results

The numerical testing was performed on a Sun SPARCstation 20 running SunOS release 4.1.3_U1. The routines and test problems were all compiled with the Sun Fortran compiler version SC3.0.1.

Routines used

We compared the performance of our routine UMINHV against that of two leading routines that also use $O(n)$ storage: L-BFGS, a limited-memory quasi-Newton method [14], and TN, a truncated-Newton method [16].

The routine L-BFGS is a limited-memory version of the BFGS method (the BFGS method is discussed in [9, Section 4.5.2]. It uses information from preceding steps to build a model of the inverse Hessian. The matrix approximating the inverse Hessian is not actually formed, however, since this would require $O(n^2)$ storage; instead, the m preceding displacements and changes in gradient are saved and used to calculate the matrix-vector product with an approximation of the inverse Hessian. The code has no default value for m, though it recommends $3 \leq m \leq 7$; we used the value $m = 5$ used by Nocedal in [17].

The routine TN approximates $-{A_i}^{-1} b_i$ at each iteration using a preconditioned linear conjugate-gradient method (discussed in [10, Sections 10.2 and 10.3]). It uses information from preceding steps for the preconditioner, a scaled two-step limited-memory BFGS method. It then calculates matrix-vector products of the current Hessian through finite differences of the gradient. The code requires an estimate of the value of the objective function at the solution; for this estimate we used the exact value.

The information used by L-BFGS at each step is mostly from the preceding steps. TN uses a mixture of information from preceding steps and the current step. UMINHV uses only information at the current step. As tested, L-BFGS required about $11n$ workspace, TN required $14n$ workspace, and UMINHV required about $23n$ workspace.

Performance measures

With the limit value f_* of the objective function known for each test problem, we were able to use the reduction in error

$$\epsilon_i = \left| \frac{f_i - f_*}{f_0 - f_*} \right|$$

as a uniform measure of the routines' convergence. L-BFGS and TN evaluate the function and the gradient. UMINHV in addition relies on the Hessian-vector product. Since the computation required for the ADIFOR-generated Hessian-vector product is roughly comparable to that for the gradient, and since in general this product could be approximated with a gradient evaluation, we combined gradient with Hessian-vector product evaluations. Thus we recorded the number of iterations, function evaluations, and gradient and Hessian-vector product evaluations required by each routine to achieve tolerance levels $\epsilon_i = 10^{-3}$, 10^{-6}, 10^{-9}, and 10^{-12}.

Problem	Routine		
	UMINHV	L-BFGS	TN
BROY7D	10^{-10}	10^{-9}	10^{-10}
BRYBND			10^{-11}
BRYBND18			10^{-11}
MORBV998	10^{-2}	10^{-2}	10^{-3}
PEN2GM2	10^{-6}	10^{-6}	10^{-6}
PENL3GM4			10^{-10}
PENL3GM5			10^{-1}
POWB1000		10^{-7}	1
POWBS500		10^{-10}	1
POWBSC50		10^{-12}	1
WATSON12		10^{-11}	10^{-10}

Table 4.2: Order of magnitude of $\|\nabla f_{\text{last}}\| \,/\, \|\nabla f_0\|$ at solution failures.

Robustness

Of the 78 test problems, 11 caused at least one of the three routines to fail. Table 4.2 shows those instances where the routines failed to reduce ϵ_i to 10^{-12}. The ratios of the gradient norms given in the table show that in some instances a routine would find a critical point, but not the one associated with the initial point.

Run with its default parameters, UMINHV correctly solved 75 of the 78 test problems. It found the wrong local minimizers for BROY7D and PEN2GM2. UMINH also found the wrong local minimizers for these two problems. Both UMINH and UMINHV found the correct local minimizers when set to perform a more careful search. UMINHV reached its iteration limit before solving MORBV998. The Hessian of MORBV998 is highly ill-conditioned at the points generated, with a condition number of about 10^{11}. UMINHV solved this problem within its iteration limit when set to use a smaller RTOL and a larger LMAX than the default values.

L-BFGS failed to solve the same three problems that UMINHV failed to solve. In addition, it failed to solve four others: POWB1000, POWBS500, POWBSC50, and WATSON12. However, it came very close to solving these problems, reducing ϵ_i to 10^{-9} but not to 10^{-12}.

TN also failed to solve the three problems that UMINHV failed to solve. In addition, it failed to solve eight others. For one problem, WATSON12, it reduced ϵ_i to 10^{-9} but not to 10^{-12}. For three problems, BRYBND, BRYBND18, and PENL3GM4, it appears to have found critical points, but not the ones associated with the initial point. For the remaining four problems, PENL3GM5, POWB1000, POWBS500, and POWBSC50, it did not find critical points.

Although it is common practice to compare the performance of optimization algorithms on a standard set of test problems as we have done, their performance on real problems may differ. While the test set included some real problems, many others were designed to be particularly difficult to solve. Some had functions designed to be hard to solve, such as the Powell badly-scaled problem. Others had bad initial points. Since real problems rarely exhibit the extreme pathologies present in the test problems and frequently have good initial points, the routines are likely to be more robust in practice than they were in our tests.

Speed

The 78 test problems were run on each of the three routines with the parameter values given above. For each problem and routine, we recorded the number of iterations, function evaluations, and gradient and Hessian-vector product evaluations required for the convergence measure ϵ_i to reach the tolerance levels 10^{-3}, 10^{-6}, 10^{-9}, and 10^{-12}. The performance of each routine on each problem is given in Appendix C. The results are summarized in Tables 4.3–4.5. The tables contain for each performance measure, each tolerance level, and each routine, the number of test problems on which the routine finished first, second, and third among the three routines and the number of problems that it failed to solve. Ties occurred frequently.

UMINHV and TN had fewer, more-expensive iterations than L-BFGS. In terms of number of iterations and number of function evaluations, UMINHV had the highest number of first-place finishes, TN had the second highest, and L-BFGS had the lowest. In terms of number of gradient and Hessian-vector product evaluations, L-BFGS had the highest number of first-place finishes, followed by UMINHV and TN. The performance of the routines was consistent across the four tolerance levels. Note

Tolerance	Routine	Test Problem Place Finishes			
		First	Second	Third	Failed
10^{-3}					
	UMINHV	52	22	2	2
	L-BFGS	2	27	46	3
	TN	33	31	6	8
10^{-6}					
	UMINHV	46	27	2	3
	L-BFGS	1	26	48	3
	TN	37	25	6	10
10^{-9}					
	UMINHV	43	28	4	3
	L-BFGS	1	24	50	3
	TN	39	25	4	10
10^{-12}					
	UMINHV	45	26	4	3
	L-BFGS	1	20	50	7
	TN	36	26	5	11

Table 4.3: Performance summary for number of iterations.

that since L-BFGS is a first derivative method, it does not compute any Hessian-vector products. No attempt was made to account for the work required by its inverse Hessian approximation.

Tolerance	Routine	Test Problem Place Finishes			
		First	Second	Third	Failed
10^{-3}					
	UMINHV	70	4	2	2
	L-BFGS	8	61	6	3
	TN	0	13	57	8
10^{-6}					
	UMINHV	66	7	2	3
	L-BFGS	9	58	8	3
	TN	1	13	54	10
10^{-9}					
	UMINHV	66	6	3	3
	L-BFGS	8	55	12	3
	TN	1	16	51	10
10^{-12}					
	UMINHV	65	6	4	3
	L-BFGS	10	52	9	7
	TN	2	16	49	11

Table 4.4: Performance summary for number of function evaluations.

Tolerance	Routine	Test Problem Place Finishes			
		First	Second	Third	Failed
10^{-3}					
	UMINHV	17	32	27	2
	L-BFGS	63	11	1	3
	TN	10	31	29	8
10^{-6}					
	UMINHV	9	31	35	3
	L-BFGS	62	11	2	3
	TN	11	32	25	10
10^{-9}					
	UMINHV	8	26	41	3
	L-BFGS	57	14	4	3
	TN	14	34	20	10
10^{-12}					
	UMINHV	14	21	40	3
	L-BFGS	56	12	3	7
	TN	13	36	18	11

Table 4.5: Performance summary for number of gradient and Hessian-vector product evaluations.

Chapter 5

Summary and future work

5.1 Summary

This dissertation presented a method for unconstrained optimization based upon approximating the gradient flow of the objective function (Algorithm 2.2.1). Under mild assumptions the method was shown to converge to a critical point from any initial point (Theorem 2.2.1) and to converge quadratically in the neighborhood of a solution (Theorem 2.2.5).

Two implementations of the method were presented: UMINH, using explicit Hessians and $O(n^2)$ storage, and UMINHV, using Hessian-vector products and $O(n)$ storage. These implementations were written in ANSI-standard Fortran 77 for others to use. They have been extensively tested and have proven to be very reliable and efficient in comparison to leading alternative routines.

5.2 Future work

5.2.1 Software

UMINH and UMINHV sometimes found local minimizers not in the basin of the initial point. The option for a more careful search can be provided in the following way. The value of the objective function along the search curve can be modeled with the available information, and the deviation of this value from the actual value can be used as a measure of trust of the search curve. By specifying bounds on this deviation, we can specify the level of care for the search. Experiments with this approach showed that when sufficient care was used, the solution in the basin of the initial point was found in all cases.

UMINHV uses at each step only fresh information of the current Hessian through matrix-vector products. An alternative approach is to approximate the Hessian at the current step based upon a model that reuses information gathered from the Hessian at preceding steps. Modeling the Hessian and reusing information have been effective techniques in other methods and may be effective here. There are many different ways to pursue modeling and information reuse, several of which are discussed in Section 4.1.3.

5.2.2 Applications

The method of this dissertation is an outgrowth of a method proposed for solving variational partial differential equations (Section 1.1). Some of the test problems come from such equations (including some, such as the minimum surface area problem, that were of interest Courant). It would be instructive to compare the performance of the method on nonlinear variational partial differential equations with that of methods specifically designed for this purpose.

Finally, in the past several years a wide range of problems has been shown to be solvable through the technique of following a gradient flow on a manifold [12]. These problems include problems in constrained optimization, discrete optimization, control theory, and signal processing. The extension of the ideas of this dissertation to the realm of manifolds would therefore extend their applicability to a number of important practical problems.

Appendix A

Test problem documentation

The software in Chapters 3 and 4 was tested with 142 test problems from the Buckley test set [4]. Section 3.2.1 describes the problems, how they were implemented in Fortran with the aid of Mathematica [25] and ADIFOR [2], [3], and how they were solved in quadruple precision. In this section we describe how to use the problems.

The test problems were implemented in an object-oriented style that proved easy to use with all of the software tested. Each test problem has two versions of the form prob.f and prob.ad.f, which we call the ANSI and ADIFOR versions respectively. The ANSI version is in ANSI-standard Fortran 77. The ADIFOR version contains ADIFOR-generated code using the following three extensions to the ANSI standard: use of the underscore in identifiers, the DO ... END DO statement, and symbolic names longer than 6 characters. If the user's Fortran compiler can compile the ADIFOR version, we recommend using it instead of the ANSI version because the ADIFOR version has additional functionality. The ADIFOR version does not need to be linked with the ADIFOR library.

Each version of the test problems contains 15 functions and subroutines. First are two functions to get the problem name and number of variables.

```
      CHARACTER*32 FUNCTION GPNAME()
*
*  Get problem name.
*
      INTEGER FUNCTION GN()
*
*  Get number of variables.
*
```

Next are six logical functions that return whether the initial and limit points are provided, whether the limit value of the objective function is provided, and whether the gradient, Hessian-vector product, and Hessian are provided.

```
      LOGICAL FUNCTION XOQ()
*
*  Query whether initial point provided.
*
      LOGICAL FUNCTION XLQ()
*
*  Query whether limit point provided.
*
      LOGICAL FUNCTION FLQ()
*
*  Query whether limit value of objective function provided.
*
      LOGICAL FUNCTION GRADQ()
*
*  Query whether analytic gradient provided.
*
      LOGICAL FUNCTION HESSVQ()
*
*  Query whether analytic Hessian-vector product provided.
*
      LOGICAL FUNCTION HESSQ()
*
*  Query whether analytic Hessian provided.
*
```

Next are three routines for getting the initial point, limit point, and limit value of the objective function.

```
      SUBROUTINE GX0( N, X0, INFO )
*
*  Get problem initial point.
*
      SUBROUTINE GXL( N, XL, INFO )
*
*  Get problem limit point.
*
      SUBROUTINE GFL( FL, INFO )
*
*  Get limit value of objective function.
*
```

The initial point and limit value of the objective function are provided for all test problems. The limit point is provided for 99 of the 142 problems. The limit point is provided for all problems of size $n \leq 20$ except for 6 problems that had minimizers sharing the same limit value, such as a line of minimizers. In addition, the limit point is provided for 18 problems for which the exact minimizer is known and for PENL3GM3, $n = 50$, which is particularly difficult to solve. The logical function XLQ returns whether or not the limit point is provided for a particular problem. The INFO argument for each routine returns 0 for a successful exit and 1 when the information of the routine is not provided for the problem. The limit points and limit values of the objective functions were calculated in several different ways in quadruple precision, and the solutions always agreed to the double precision provided.

The final four routines calculate the objective function, the gradient, the Hessian-vector product, and the Hessian.

```
      SUBROUTINE CF( N, X, F, INFO )
*
*  Calculate value of objective function at point X.
*
      SUBROUTINE CGRAD( N, X, GRAD, INFO )
*
*  Calculate gradient of objective function at point X.
*
      SUBROUTINE CHESSV( N, X, V, HESSV, INFO )
*
*  Calculate Hessian of objective function at point X multiplied by
*  vector V.
*
      SUBROUTINE CHESS( N, X, HESS, LDHESS, INFO )
*
*  Calculate Hessian of objective function at point X.
*
```

The objective function and the gradient are provided for all test problems. The Hessian-vector product is not provided in the ANSI version but is provided in the ADIFOR version. The Hessian-vector product is calculated directly without computing the Hessian. For most problems, the time to compute the product is comparable to that of computing the gradient. The Hessian is provided in the ANSI version for all problems of size $n \leq 20$ and is provided in the ADIFOR version for all problems. The INFO argument for each routine returns 0 for a successful exit, 1 when the information of the routine is not provided for the problem, and another number, described in the comments of each routine, when an error is encountered.

Appendix B

Software documentation

We provide two routines for solving unconstrained optimization problems: UMINH, which requires $O(n^2)$ storage, and UMINHV, which requires $O(n)$ storage. Which of the two routines to use will depend upon the available hardware, the problem to be solved, and the user's aims. The size of the problem and the available storage could preclude UMINH and require UMINHV. If there is sufficient storage to solve the problem with both routines, the user may be guided by the following considerations. Both routines use the objective function and the gradient. In addition UMINH uses the Hessian, and UMINHV uses the Hessian-vector product; however, both of these are equally easily to generate using ADIFOR. As a rule UMINH will require fewer iterations to solve a problem than UMINHV, but each iteration will take longer to compute. The total computation time to solve the problem will depend upon the cost of evaluating the user's routines. As a rule, UMINH will be more robust, that is, more likely to find a solution.

We compiled data on the time it took to solve the same test problems with both routines. On a Sun SPARCstation 20 (in 1997), we solved 78 test problems of size $10 \leq n \leq 1000$ and timed how long it took each routine to satisfy the convergence criterion $|f_* - f_i| \leq 10^{-12}|f_* - f_0|$, where f_* is the value of the objective function at the solution. On the 55 test problems of size $10 \leq n \leq 100$, both routines took less than 1 minute. On the 23 test problems of size $400 \leq n \leq 1000$, UMINHV took less time than UMINH; for 20 of these problems, UMINHV still took less than 1 minute. UMINHV, however, failed to solve one of the problems that UMINH solved.

B.1 Files and compilation

UMINH and UMINHV each has its own directory containing everything necessary to run each routine. These two directories and their subdirectories contain README files that explain their contents.

Each of the top-level directories for UMINH and UMINHV has a Makefile for compiling the routines. Each Makefile contains eight variables that can be set by the user. Each variable is explained in comments in the Makefile and is set to default values. The variables specify the Fortran compiler (default: f77), the compiler options (default: ⟨none⟩), the name of the file containing the problem to solve (default: prob.f), whether to use LAPACK and BLAS libraries available on the system or to use the provided source (default: ⟨use provided source⟩), and the remove command (default: /bin/rm -f).

The directories for UMINH and UMINHV also contain several ancillary files. The file run.f contains a driver routine. The driver contains a parameter NMAX that must be at least as large as the problem's number of variables (default: 100 for UMINH, 1000 for UMINHV). The driver gets the problem's number of variables from the function GN, and it gets the initial point from the subroutine GX0 in prob.f. The file prob.f provided is a sample file of problem routines. The user need not provide all of the routines provided in the sample file, only the ones explicitly mentioned above or below. The file spar.f contains the routine for setting algorithm parameters. We recommend that the user change only the parameter ITMAX to specify the maximum number of iterations (default: 2000).

The directory ccnvg contains several files to specify the convergence criteria for the algorithm.

ccnvg_false.f	Return not converged.
ccnvg_f_absolute.f	$\lvert f_i - f_{i-1}\rvert \ \leq\ \text{TOL}$
ccnvg_f_relative.f	$\lvert f_i - f_{i-1}\rvert \ \leq\ \text{TOL}\lvert f_i - f_0\rvert$
ccnvg_grad_absolute.f	$\lVert\nabla f_i\rVert_2 \ \leq\ \text{TOL}$
ccnvg_grad_relative.f	$\lVert\nabla f_i\rVert_2 \ \leq\ \text{TOL}\,\lVert\nabla f_0\rVert_2$

Each routine contains the parameter TOL to specify the convergence tolerance (default: 10^{-6}). The file ccnvg.f in the same directory as the Makefile is the file used by the compile. This file is initially a copy of ccnvg_f_relative.f. The user may specify another of the provided convergence routines or one of his or her own by copying the desired routine into ccnvg.f for the compile.

The directory out contains two files for writing out iteration information. The file out_null.f contains a routine that writes out no information. The file out_f_grad.f contains a routine that writes out the iteration, the value of the objective function, and the 2-norm of the gradient; for UMINH, it in addition writes out whether the Hessian is positive definite. It writes out these values at the initial point, the final point, and at iterations that are integral multiples of the parameter INC (default: 1). It gets the problem's name from the function GPNAME in prob.f. The name of the output file is specified by the parameter FILE (default: OUT_F_GRAD). A file of this name will be created unless a file with the same name already exists, in which case an error message will result. The file out.f in the same directory as the Makefile is the file used by the compile. This file is initially a copy of out_f_grad.f. The user may specify the other output routine provided or one of his or her own by copying the desired routine into out.f for the compile.

The directory explog contains files for accurately calculating $\exp(x) - 1$ and $\log(1 + x)$. Fortran 77, the implementation language, does not have intrinsics for these two functions. Many C math libraries, however, contain the functions expm1 and log1p for this purpose. It is preferable to use these C intrinsics, if they are available, and the file explog.c contains routines to access these intrinsics from Fortran. If expm1 and log1p are not available on a system, or if for some other reason explog.c does not work, we have provided the file explog.f, which contains ANSI-standard Fortran 77 routines to calculate accurately $\exp(x) - 1$ and $\log(1 + x)$ from exp and log using algorithms developed by Prof. William Kahan at the University of California, Berkeley. In our testing, explog.c worked on the following platforms: Sun Solaris, SGI IRIX, IBM AIX, and DEC Ultrix. Special files were needed for HP UX and Cray Unicos; these files are in explog/special. One of explog.c or explog.f must be in the same directory as the Makefile for the compile; the default is explog.f.

B.2 UMINH

The routine UMINH is for solving unconstrained optimization problems using the objective function, its gradient, and its Hessian; it requires $O(N^2)$ storage. The argument list for UMINH is as follows:

```
      SUBROUTINE UMINH( N, X, F, GRAD, CF, CGRAD, CHESS,
     &  SPAR, CCNVG, OUT, WORK, LWORK, INFO )
*
*  Find minimizer of objective function from initial point X.
*
*  .. Arguments ..
*  I   N        Number of variables.
*  IO  X        On entry, initial point.
*               On exit, final point.
*   O  F        Value of objective function at final point.
*   O  GRAD     Gradient of objective function at final point.
*      CF       Calculate value of objective function.
*      CGRAD    Calculate gradient of objective function.
*      CHESS    Calculate Hessian of objective function.
*      SPAR     Set parameters.
*      CCNVG    Determine whether point satisfies convergence criteria.
*      OUT      Write iteration information.
*      WORK     Workspace.
*      LWORK    Length of WORK.
*   O  INFO     Information on subroutine execution.
*                   INFO =  0   Successful exit.
*                   INFO =  1   Error, invalid N.
*                   INFO =  2   Error, insufficient workspace.
*                   INFO =  3   Error, objective function value could
*                               not be improved from final point.
*                   INFO =  4   Error, ITMAX iterations reached without
*                               convergence.
*                   INFO = x5   Error exit x from CF.
*                   INFO = x6   Error exit x from CGRAD.
*                   INFO = x7   Error exit x from SPAR.
*                   INFO = x8   Error exit x from CCNVG.
*                   INFO = x9   Error exit x from OUT.
*                   INFO = x0   Error exit x from NX.
*
```

On entry, the user provides the number of variables in N and the initial point in X. On exit, the routine provides the final point in X and the value and gradient of the objective function at the final point in F and GRAD, respectively. The user provides the routines to calculate the value, gradient, and Hessian of the objective function, respectively CF, CGRAD, and CHESS in the file prob.f. As discussed above, the routines SPAR, CCNVG, and OUT are provided for the user, or others may be substituted. Finally, the user must provide a vector of workspace of length LWORK $\geq N^2 + 9\,N$.

On exit, the variable INFO contains information on whether the routine was successful or not. If INFO returns a value of 0, then the convergence criteria were satisfied at the returned X. If INFO returns a value of 3, then the objective function value could not be improved from the returned X. This will occur when the returned X is a local minimizer but the convergence criteria were too strict to be satisfied. If INFO returns a value of 4, then ITMAX iterations were reached without convergence, where ITMAX is a parameter set in spar.f. If INFO returns a value ending in 9, then the routine encountered a problem writing the output. This will occur when a file already exists, perhaps from a previous run, that has the same name as the output file that the routine is trying to write; this may be remedied by renaming the existing file. If INFO returns the value of 2, then there was insufficient workspace; this may be remedied by increasing LWORK. If INFO returns any other value, the user should consult the comments of uminh.f to learn the nature of the error.

B.3 UMINHV

The routine UMINHV is for solving unconstrained optimization problems using the objective function, its gradient, and its Hessian-vector product (directional derivative); it requires $O(N)$ storage. The argument list for UMINHV is as follows:

```
      SUBROUTINE UMINHV( N, X, F, GRAD, CF, CGRAD, CHESSV,
     &  SPAR, CCNVG, OUT, WORK, LWORK, INFO )
*
*  Find minimizer of objective function from initial point X.
*
*  .. Arguments ..
*  I   N        Number of variables.
*  IO  X        On entry, initial point.
*               On exit, final point.
*   O  F        Value of objective function at final point.
*   O  GRAD     Gradient of objective function at final point.
*      CF       Calculate value of objective function.
*      CGRAD    Calculate gradient of objective function.
*      CHESSV   Calculate directional derivative of objective function.
*      SPAR     Set parameters.
*      CCNVG    Determine whether point satisfies convergence criteria.
*      OUT      Write iteration information.
*      WORK     Workspace.
*      LWORK    Length of WORK.
*   O  INFO     Information on subroutine execution.
*                   INFO =  0   Successful exit.
*                   INFO =  1   Error, invalid N.
*                   INFO =  2   Error, insufficient workspace.
*                   INFO =  3   Error, objective function value could
*                               not be improved from final point.
*                   INFO =  4   Error, ITMAX iterations reached without
*                               convergence.
*                   INFO = x5   Error exit x from CF.
*                   INFO = x6   Error exit x from CGRAD.
*                   INFO = x7   Error exit x from SPAR.
*                   INFO = x8   Error exit x from CCNVG.
*                   INFO = x9   Error exit x from OUT.
*                   INFO = x0   Error exit x from NX.
*
```

On entry, the user provides the number of variables in N and the initial point in X. On exit, the routine provides the final point in X and the value and gradient of the objective function at the final point in F and GRAD, respectively. The user provides the routines to calculate the value, gradient, and directional derivative of the objective function, respectively CF, CGRAD, and CHESSV in the file prob.f. As discussed above, the routines SPAR, CCNVG, and OUT are provided for the user, or others may be substituted. Finally, the user must provide a vector of workspace of length $\mathrm{LWORK} \geq \mathrm{LMAX\ N} + 2\ \mathrm{LMAX} + \max(7\ \mathrm{N}, \mathrm{N} + \mathrm{LMAX}^2 + 2\ \mathrm{LMAX})$, where LMAX is the maximum number of Lanczos vectors, a parameter set in spar.f.

On exit, the variable INFO contains information on whether the routine was successful or not. If INFO returns a value of 0, then the convergence criteria were satisfied at the returned X. If INFO returns a value of 3, then the objective function value could not be improved from the returned X. This will occur when the returned X is a local minimizer but the convergence criteria were too strict to be satisfied. If INFO returns a value of 4, then ITMAX iterations were reached without convergence, where ITMAX is a parameter set in spar.f. If INFO returns a value ending in 9, then the routine encountered a problem writing the output. This will occur when a file already exists, perhaps from a previous run, that has the same name as the output file that the routine is trying to write; this may be remedied by renaming the existing file. If INFO returns the value of 2, then there was insufficient workspace; this may be remedied by increasing LWORK. If INFO returns any other value, the user should consult the comments of uminhv.f to learn the nature of the error.

Appendix C

Numerical results

The detailed results of the numerical testing of UMINH and UMINHV and their comparison routines are given in the tables of this section. The tables contain the relative performance of each routine on each problem for each performance measure and each tolerance level of the convergence measure

$$\epsilon_i = \left| \frac{f_i - f_*}{f_0 - f_*} \right| .$$

We compared the performance of UMINH against E04LBF, a line-search method from the NAG library [18], and DMNH, a trust-region method from the AT&T PORT library [8], on 142 test problems. The routines' performance in terms of number of iterations, function evaluations, and gradient evaluations for the tolerance levels 10^{-3}, 10^{-6}, 10^{-9}, and 10^{-12} is contained in Tables C.1–C.12.

We compared the performance of UMINHV against L-BFGS, a limited-memory quasi-Newton method [14], and TN, a truncated-Newton method [16], on 78 test problems. The routines' performance in terms of number of iterations, function evaluations, and gradient and Hessian-vector product evaluations for the tolerance levels 10^{-3}, 10^{-6}, 10^{-9}, and 10^{-12} is contained in Tables C.13–C.24.

Each table has columns for the function name, problem name, and size. Recall that one function can have several problems that specify different function sizes, parameters, or initial points. Next, for each of the three routines there is a set of four columns marked 1, 2, 3, and f. If the performance of a routine on a given problem ranked first, second, or third among the three routines, then there will be a number respectively in its 1, 2, or 3 column. The number in these columns is the ratio of the routine's performance on the problem to that of the best routine's performance. In Table C.1, for example, for the solution of the problem BOX662HL, the DMNH required the fewest number of iterations, UMINH came in second, requiring about 40% more iterations than DMNH, and E04LBF came in third, requiring about 60% more iterations than DMNH. Ties occurred frequently. If a routine failed to solve a given problem, there is an entry in its f column.

The bottom row of each table summarizes the number of first-place, second-place, and third-place finishes for each routine along with the number of problems that it failed to solve. Tables 3.2–3.4 collect these summary statistics from Tables C.1–C.12. Tables 4.3–4.5 collect these summary statistics from Tables C.13–C.24.

Table C.1: Place finishes for number of iterations, tolerance $= 10^{-3}$.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					2.00		
ARGQDO	ARGQO10	5	1				1					2.00		
ARGQDZ	ARGQDZ10	3	1				1					2.00		
ARTRIG	ARTRIG10	10	1				1				1			
AVRIEL	AVRIEL3	2	1				1					2.00		
BARD70	BARD70	3	1				1				1			
BEAL58	BEAL58KO	2	1					1.25				1.25		
BIGGS	BIGGS6	6		3.50			1							f
BOOTH	BOOTH	2	1				1				1			
BOX66	BOX662HL	2		1.40					1.60		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2		1.20					2.00		1			
BROWND	BROWND	4	1				1					1.20		
BROY7D	BROY7D	60				f		1.50			1			
BRWNAL	BRWNAL10	10	1				1					2.00		
BRWNAL	BRWNL100	100	1				1					2.00		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1				1			
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1					1.50				1.50		
BRYTRI	BRYTRI10	600	1				1					1.50		
CHEBYQ	CHEBYQ8	8	1					1.20			1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10	1					1.12				1.12		
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2	1					1.67					2.00	
CRGLVY	CRGLVY	4	1					1.17			1			
CRGLVY	CRGLVY10	10	1				1				1			
CRGLVY	CRGLY500	500	1				1					1.29		
CRGLVY	CRGL1000	1000	1				1					1.14		
DIX7DG	DIX7DGA	15	1					1.25				1.25		
DIXON	DIXON	10	1				1					2.00		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1				1			
ENGVL1	ENGVL1B4	100	1				1				1			
ENGVL1	ENGVL1B6	1000	1				1				1			
ENGVL2	ENGVL2	3		1.60				1.60			1			
EXTRSN	EXTRAR10	10	1				1				1			
EXTRSN	EXTRAR50	50	1				1					1.25		
EXTRSN	EXTRA100	100	1				1					1.06		
EXTRSN	EXTR1000	1000	1				1					1.12		
FRANK	FRANK8	8	1				1					2.00		
FRANK	FRANK12	12	1				1					2.00		
FRDRTH	FRDRTH	2		1.33				1.33			1			
FRDRTH	FRDRTHB3	50	1				1					1.33		
FRDRTH	FRDRTHB4	100	1				1				1			
FRDRTH	FRDRTHB7	1000	1				1				1			
GENRSN	GENT2B	2	1					1.50			1			
GENRSN	GENT50A	50		1.02			1						1.35	
GENRSN	GENT500A	500		1.03			1						1.25	
GENRSN	GENT1000	1000		1.03			1						1.26	
GOTTFR	GOTTFR	2		4.00			1					4.00		
GULF	GULFSH2	3	1					1.75				1.75		
HELIX	HELIX	3		1.20			1				1			
HILBRT	HILBR10A	10		2.00			1					2.00		
HILBRT	HILBRT12	12		2.00			1					2.00		
HIMLN3	HIMLN3	2	1				1				1			
HIMM1	HIMM1	2	1				1					2.00		
HIMM25	HIMM25	2	1				1					2.00		
HIMM27	HIMM27	2		1.17			1				1			
HIMM28	HIMM28	2	1				1				1			
HIMM29	HIMM29	2	1				1				1			
HIMM30	HIMM30	3	1				1				1			
HIMM32	HIMM32	4	1				1				1			
HIMM33	HIMM33A	2	1						2.00			1.50		
HYPCIR	HYPCIR	2	1				1				1			
JENSMP	JENSMP	2	1				1				1			
KOWOSB	KOWOSB1	4		1.50			1					1.50		
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1				1			
MEYER	MEYER	3			3.00		1					1.67		
MNSRF1	NMSURF64	36			1.64		1					1.27		
MNSRF1	NMSUR484	400		4.27			1						8.67	
MNSRF2	SNMSUR64	36	1							f		1.46		
MNSRF2	SNMSR484	400		1.04						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1				1					2.00		
NONDIA	NONDIA10	10	1				1					1.20		
NONDIA	NONDIA20	20	1				1				1			
NONDIA	NONDI500	500	1				1				1			
NONDIA	NOND1000	1000	1				1					1.10		
OSBRN1	OSBRN1	5		2.67			1							f

OSBRN2	OSBRN2	11	1							f		1.38		
PENAL1	PEN1GM6	10	1				1				1			
PENAL1	PEN1LN1	50	1				1				1			
PENAL1	PEN1LN2	100	1				1				1			
PENAL1	PEN1LN3	1000	1				1				1			
PENAL2	PEN2GM6	4	1				1				1			
PENAL2	PEN2GM1	50	1							f		1.75		
PENAL2	PEN2GM2	100	1							f		2.89		
PENAL3	PENL3GM3	50	1				1					1.25		
PENAL3	PENL3GM4	100	1						1.38			1.12		
PENAL3	PENL3GM5	1000	1					1.44					1.67	
POWBSC	POWBSC	2	1				1					1.29		
POWBSC	POWBSC50	50	1						7.60			1.20		
POWBSC	POWBS500	500		1.25					69.00		1			
POWBSC	POWB1000	1000		1.50						f	1			
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1				1			
POWQUD	POWQUD8A	4	1				1				1			
POWSSQ	POWSSQ	2				f	1					9.00		
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1				1			
PWSING	PWSIN100	100	1				1				1			
PWSING	PWSI1000	1000	1				1				1			
QUARTC	QUARTC	25	1				1					1.40		
RECIPE	RECIPE	3		1.50				1.50			1			
ROSENB	ROSENB	2	1				1					1.12		
ROSENB	SHNRSN10	10	1				1				1			
SARSEB	SARSEB	4	1				1				1			
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1				1			
SCHMVT	SCHMV500	500	1				1				1			
SCHMVT	SCHV1000	1000	1				1				1			
SISSER	SISSER	2	1				1					1.20		
SQRTMX	MSQRTB9	9		1.25					2.00		1			
SQRTMX	MSQRTB49	49	1						77.00			1.20		
TDQUAD	TDQ10	10	1				1				1			
TDQUAD	TDQ500	500	1				1					2.00		
TDQUAD	TDQ1000	1000	1				1					2.00		
TOIN2	TOIN2	3	1				1				1			
TOIN4	TOIN4	4	1				1					3.00		
TOINT	PSPTOINT	50		1.50			1						2.00	
TRIDIA	TRIDIA10	10	1				1					2.00		
TRIDIA	TRLN100	100	1							f		2.00		
TRIDIA	TRLN1000	1000	1							f		2.00		
TRIGTO	TRIGT50	50	1						2.12			1.12		
TRIGTO	TRIGT100	100	1					2.10			1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1				1			
VAROSB	VAROSBG1	50	1				1					3.00		
VAROSB	VAROSBG2	100	1				1					4.00		
WATSON	WATSON6	6	1				1				1			
WATSON	WATSON12	12	1				1				1			
WOODS	WOODS	4	1				1				1			
WOODS	WOODS80	80	1				1				1			
XTX	XTX2	2	1				1				1			
XTX	XTX16	16	1				1					2.00		
ZANGWL	ZANGWL1	3	1				1					2.00		
Totals			116	22	2	2	109	15	9	9	74	58	7	3

Table C.2: Place finishes for number of iterations, tolerance $= 10^{-6}$.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					2.00		
ARGQDO	ARGQO10	5	1				1					2.00		
ARGQDZ	ARGQDZ10	3	1				1					2.00		
ARTRIG	ARTRIG10	10		1.14						f	1			
AVRIEL	AVRIEL3	2	1				1					2.00		
BARD70	BARD70	3	1				1				1			
BEAL58	BEAL58KO	2	1					1.17			1			
BIGGS	BIGGS6	6		2.32			1							f
BOOTH	BOOTH	2	1				1					2.00		
BOX66	BOX662HL	2		1.50					1.67		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2		1.17					1.83		1			
BROWND	BROWND	4	1				1					1.33		
BROY7D	BROY7D	60				f		1.22			1			
BRWNAL	BRWNAL10	10	1				1					1.50		
BRWNAL	BRWNL100	100	1							f		2.00		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1					1.20		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1				1					1.33		
BRYTRI	BRYTRI10	600	1				1					1.33		
CHEBYQ	CHEBYQ8	8	1					1.18			1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10	1					1.09				1.09		
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2	1					1.29			1			
CRGLVY	CRGLVY	4	1				1				1			
CRGLVY	CRGLVY10	10	1				1				1			
CRGLVY	CRGLY500	500	1				1					1.20		
CRGLVY	CRGL1000	1000	1				1					1.10		
DIX7DG	DIX7DGA	15	1				1					1.20		
DIXON	DIXON	10	1				1					2.00		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1				1			
ENGVL1	ENGVL1B4	100	1				1					1.20		
ENGVL1	ENGVL1B6	1000	1				1				1			
ENGVL2	ENGVL2	3		1.20				1.20			1			
EXTRSN	EXTRAR10	10	1					1.06					1.11	
EXTRSN	EXTRAR50	50	1					1.06					1.33	
EXTRSN	EXTRA100	100	1					1.06					1.11	
EXTRSN	EXTR1000	1000	1					1.06					1.22	
FRANK	FRANK8	8	1				1					2.00		
FRANK	FRANK12	12	1				1					2.00		
FRDRTH	FRDRTH	2	1				1				1			
FRDRTH	FRDRTHB3	50	1				1					1.50		
FRDRTH	FRDRTHB4	100	1				1					1.25		
FRDRTH	FRDRTHB7	1000	1				1				1			
GENRSN	GENT2B	2	1					1.33			1			
GENRSN	GENT50A	50		1.03			1						1.39	
GENRSN	GENT500A	500		1.03			1						1.26	
GENRSN	GENT1000	1000		1.03			1						1.26	
GOTTFR	GOTTFR	2		2.25			1					2.25		
GULF	GULFSH2	3		1.25			1						1.44	
HELIX	HELIX	3			1.67		1					1.50		
HILBRT	HILBR10A	10		2.00			1					2.00		
HILBRT	HILBRT12	12		2.00			1					2.00		
HIMLN3	HIMLN3	2		1.50			1					1.50		
HIMM1	HIMM1	2	1				1					2.00		
HIMM25	HIMM25	2	1				1					2.00		
HIMM27	HIMM27	2		1.11			1				1			
HIMM28	HIMM28	2	1				1				1			
HIMM29	HIMM29	2	1				1				1			
HIMM30	HIMM30	3	1				1					1.12		
HIMM32	HIMM32	4	1					1.33				1.33		
HIMM33	HIMM33A	2	1						1.67			1.33		
HYPCIR	HYPCIR	2	1				1				1			
JENSMP	JENSMP	2	1				1					1.14		
KOWOSB	KOWOSB1	4		1.60			1					1.60		
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1				1			
MEYER	MEYER	3		1.21			1						1.77	
MNSRF1	NMSURF64	36			1.43		1					1.21		
MNSRF1	NMSUR484	400		3.79			1						7.00	
MNSRF2	SNMSUR64	36	1							f		1.24		
MNSRF2	SNMSR484	400		1.04						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1							f		1.50		
NONDIA	NONDIA10	10	1				1					1.24		
NONDIA	NONDIA20	20	1				1					1.06		
NONDIA	NONDI500	500	1				1					1.06		
NONDIA	NOND1000	1000	1				1					1.12		
OSBRN1	OSBRN1	5		4.00			1							f

OSBRN2	OSBRN2	11	1							f		1.27		
PENAL1	PEN1GM6	10	1				1				1			
PENAL1	PEN1LN1	50	1				1				1			
PENAL1	PEN1LN2	100	1				1				1			
PENAL1	PEN1LN3	1000	1				1				1			
PENAL2	PEN2GM6	4	1				1				1			
PENAL2	PEN2GM1	50	1							f		1.60		
PENAL2	PEN2GM2	100				f				f	1			
PENAL3	PENL3GM3	50	1				1					1.10		
PENAL3	PENL3GM4	100	1						1.20			1.10		
PENAL3	PENL3GM5	1000	1					1.36					1.55	
POWBSC	POWBSC	2	1					1.04					1.20	
POWBSC	POWBSC50	50	1					4.70			1			
POWBSC	POWBS500	500		1.43					43.57		1			
POWBSC	POWB1000	1000		1.43						f	1			
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1				1			
POWQUD	POWQUD8A	4	1				1				1			
POWSSQ	POWSSQ	2				f	1					2.50		
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1				1			
PWSING	PWSIN100	100	1				1					1.11		
PWSING	PWSI1000	1000	1				1				1			
QUARTC	QUARTC	25	1				1					1.22		
RECIPE	RECIPE	3		1.33				1.33			1			
ROSENB	ROSENB	2	1					1.06					1.17	
ROSENB	SHNRSN10	10	1				1				1			
SARSEB	SARSEB	4	1				1				1			
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1				1			
SCHMVT	SCHMV500	500	1				1					1.50		
SCHMVT	SCHV1000	1000	1				1				1			
SISSER	SISSER	2	1				1					1.11		
SQRTMX	MSQRTB9	9		1.40					1.80		1			
SQRTMX	MSQRTB49	49	1							f	1			
TDQUAD	TDQ10	10	1				1				1			
TDQUAD	TDQ500	500	1				1					2.00		
TDQUAD	TDQ1000	1000	1				1					2.00		
TOIN2	TOIN2	3	1				1					2.00		
TOIN4	TOIN4	4	1				1					3.00		
TOINT	PSPTOINT	50		1.44			1						1.67	
TRIDIA	TRIDIA10	10	1				1					2.00		
TRIDIA	TRLN100	100	1							f		2.00		
TRIDIA	TRLN1000	1000	1							f		2.00		
TRIGTO	TRIGT50	50	1						2.12			1.25		
TRIGTO	TRIGT100	100	1					2.00			1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1				1			
VAROSB	VAROSBG1	50	1				1					2.00		
VAROSB	VAROSBG2	100	1				1					2.50		
WATSON	WATSON6	6	1				1					1.11		
WATSON	WATSON12	12	1				1					1.10		
WOODS	WOODS	4	1				1					1.09		
WOODS	WOODS80	80	1				1					1.09		
XTX	XTX2	2	1				1				1			
XTX	XTX16	16	1				1					2.00		
ZANGWL	ZANGWL1	3	1				1					2.00		
Totals			113	24	2	3	104	18	7	13	60	65	14	3

Table C.3: Place finishes for number of iterations, tolerance = 10^{-9}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					2.00		
ARGQDO	ARGQO10	5	1				1					2.00		
ARGQDZ	ARGQDZ10	3	1				1					2.00		
ARTRIG	ARTRIG10	10	1							f	1			
AVRIEL	AVRIEL3	2	1				1					2.00		
BARD70	BARD70	3	1					1.33			1			
BEAL58	BEAL58KO	2	1					1.14			1			
BIGGS	BIGGS6	6		2.08			1							f
BOOTH	BOOTH	2	1				1					2.00		
BOX66	BOX662HL	2		1.43					1.57		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2	1					1.57			1			
BROWND	BROWND	4	1				1					1.29		
BROY7D	BROY7D	60				f		1.20			1			
BRWNAL	BRWNAL10	10	1				1					1.20		
BRWNAL	BRWNL100	100	1							f		2.00		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1					1.17		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1					1.33				1.33		
BRYTRI	BRYTRI10	600	1				1					1.25		
CHEBYQ	CHEBYQ8	8	1					1.27			1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10	1					1.08				1.08		
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2		1.25					1.38		1			
CRGLVY	CRGLVY	4		1.06				1.06			1			
CRGLVY	CRGLVY10	10	1				1					1.15		
CRGLVY	CRGLY500	500	1				1					1.31		
CRGLVY	CRGL1000	1000	1				1					1.08		
DIX7DG	DIX7DGA	15	1				1					1.20		
DIXON	DIXON	10	1				1					2.00		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1				1			
ENGVL1	ENGVL1B4	100	1				1					1.17		
ENGVL1	ENGVL1B6	1000	1				1				1			
ENGVL2	ENGVL2	3		1.17				1.17			1			
EXTRSN	EXTRAR10	10	1					1.05					1.11	
EXTRSN	EXTRAR50	50	1					1.05					1.32	
EXTRSN	EXTRA100	100	1					1.05					1.11	
EXTRSN	EXTR1000	1000	1					1.05					1.21	
FRANK	FRANK8	8	1				1					2.00		
FRANK	FRANK12	12	1				1					2.00		
FRDRTH	FRDRTH	2		1.20				1.20			1			
FRDRTH	FRDRTHB3	50	1				1					1.20		
FRDRTH	FRDRTHB4	100	1				1					1.20		
FRDRTH	FRDRTHB7	1000	1				1				1			
GENRSN	GENT2B	2	1					1.29			1			
GENRSN	GENT50A	50		1.01			1						1.37	
GENRSN	GENT500A	500		1.03			1						1.26	
GENRSN	GENT1000	1000		1.03			1						1.26	
GOTTFR	GOTTFR	2		1.71			1					1.71		
GULF	GULFSH2	3		1.17			1						1.39	
HELIX	HELIX	3		1.38			1					1.38		
HILBRT	HILBR10A	10		2.00			1					2.00		
HILBRT	HILBRT12	12		2.00			1					2.00		
HIMLN3	HIMLN3	2		1.50			1						2.00	
HIMM1	HIMM1	2	1				1					2.00		
HIMM25	HIMM25	2	1				1					2.00		
HIMM27	HIMM27	2			1.30		1					1.20		
HIMM28	HIMM28	2	1				1				1			
HIMM29	HIMM29	2	1				1				1			
HIMM30	HIMM30	3	1				1					1.11		
HIMM32	HIMM32	4	1					1.29				1.29		
HIMM33	HIMM33A	2	1					1.25				1.25		
HYPCIR	HYPCIR	2	1					1.25			1			
JENSMP	JENSMP	2	1				1					1.12		
KOWOSB	KOWOSB1	4		1.50			1					1.50		
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1				1			
MEYER	MEYER	3		1.12			1						1.52	
MNSRF1	NMSURF64	36		1.16						f	1			
MNSRF1	NMSUR484	400	1							f		1.84		
MNSRF2	SNMSUR64	36	1							f		1.22		
MNSRF2	SNMSR484	400		1.05						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1							f		1.50		
NONDIA	NONDIA10	10	1					1.06					1.28	
NONDIA	NONDIA20	20	1					1.06				1.06		
NONDIA	NONDI500	500	1					1.06				1.06		
NONDIA	NOND1000	1000	1					1.06					1.11	
OSBRN1	OSBRN1	5		1.81			1							f

OSBRN2	OSBRN2	11	1							f		1.25		
PENAL1	PEN1GM6	10	1				1					1.09		
PENAL1	PEN1LN1	50	1				1				1			
PENAL1	PEN1LN2	100	1				1				1			
PENAL1	PEN1LN3	1000	1				1					1.08		
PENAL2	PEN2GM6	4	1					1.04					1.06	
PENAL2	PEN2GM1	50	1							f		1.50		
PENAL2	PEN2GM2	100				f				f	1			
PENAL3	PENL3GM3	50	1					1.10					1.20	
PENAL3	PENL3GM4	100	1						1.18			1.09		
PENAL3	PENL3GM5	1000	1					1.25					1.42	
POWBSC	POWBSC	2	1					1.12				1.12		
POWBSC	POWBSC50	50		5.50						f	1			
POWBSC	POWBS500	500		6.00						f	1			
POWBSC	POWB1000	1000		7.91						f	1			
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1				1			
POWQUD	POWQUD8A	4	1				1				1			
POWSSQ	POWSSQ	2				f		4.00			1			
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1					1.08		
PWSING	PWSIN100	100	1				1					1.08		
PWSING	PWSI1000	1000	1				1					1.08		
QUARTC	QUARTC	25	1				1					1.15		
RECIPE	RECIPE	3		1.25				1.25			1			
ROSENB	ROSENB	2	1					1.05					1.16	
ROSENB	SHNRSN10	10	1					1.04					1.07	
SARSEB	SARSEB	4	1				1				1			
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1				1			
SCHMVT	SCHMV500	500	1				1				1			
SCHMVT	SCHV1000	1000	1				1				1			
SISSER	SISSER	2	1				1					1.08		
SQRTMX	MSQRTB9	9		1.17					1.67		1			
SQRTMX	MSQRTB49	49	1							f	1			
TDQUAD	TDQ10	10	1				1				1			
TDQUAD	TDQ500	500	1				1					2.00		
TDQUAD	TDQ1000	1000	1				1					2.00		
TOIN2	TOIN2	3	1				1					2.00		
TOIN4	TOIN4	4	1				1					3.00		
TOINT	PSPTOINT	50		1.40			1						1.60	
TRIDIA	TRIDIA10	10		2.00			1					2.00		
TRIDIA	TRLN100	100	1							f		2.00		
TRIDIA	TRLN1000	1000	1							f		2.00		
TRIGTO	TRIGT50	50	1						2.00			1.22		
TRIGTO	TRIGT100	100		1.09					2.00		1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1				1			
VAROSB	VAROSBG1	50	1				1					2.00		
VAROSB	VAROSBG2	100	1				1					2.50		
WATSON	WATSON6	6	1				1					1.10		
WATSON	WATSON12	12	1				1					1.27		
WOODS	WOODS	4	1				1					1.11		
WOODS	WOODS80	80	1				1					1.08		
XTX	XTX2	2	1				1				1			
XTX	XTX16	16	1				1					2.00		
ZANGWL	ZANGWL1	3	1				1					2.00		
Totals			110	28	1	3	89	30	6	17	55	66	18	3

Table C.4: Place finishes for number of iterations, tolerance = 10^{-12}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					2.00		
ARGQDO	ARGQO10	5	1				1					2.00		
ARGQDZ	ARGQDZ10	3	1				1					2.00		
ARTRIG	ARTRIG10	10	1							f	1			
AVRIEL	AVRIEL3	2	1				1					2.00		
BARD70	BARD70	3	1					1.14			1			
BEAL58	BEAL58KO	2	1					1.12			1			
BIGGS	BIGGS6	6		2.02			1							f
BOOTH	BOOTH	2	1				1					2.00		
BOX66	BOX662HL	2		1.38					1.50		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2		1.14					1.57		1			
BROWND	BROWND	4	1				1					1.29		
BROY7D	BROY7D	60				f		1.27			1			
BRWNAL	BRWNAL10	10	1				1					1.17		
BRWNAL	BRWNL100	100	1							f		6.00		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1				1			
BRYTRI	BRYTRI10	600	1				1					1.25		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1				1					1.25		
CHEBYQ	CHEBYQ10	10	1							f				f
CHEBYQ	CHEBYQ8	8	1					1.17			1			
CHNRSN	CHNRSH10	10	1					1.08			1			
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2		1.50				1.50			1			
CRGLVY	CRGLVY	4		1.05				1.05			1			
CRGLVY	CRGLVY10	10	1				1					1.06		
CRGLVY	CRGLY500	500	1				1					1.43		
CRGLVY	CRGL1000	1000	1				1					1.14		
DIXON	DIXON	10	1				1					2.00		
DIX7DG	DIX7DGA	15	1				1				1			
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1				1			
ENGVL1	ENGVL1B4	100	1				1					1.14		
ENGVL1	ENGVL1B6	1000	1				1				1			
ENGVL2	ENGVL2	3		1.25				1.25			1			
EXTRSN	EXTRAR10	10	1					1.05				1.05		
EXTRSN	EXTRAR50	50	1					1.05					1.25	
EXTRSN	EXTRA100	100	1					1.05					1.10	
EXTRSN	EXTR1000	1000	1					1.05					1.20	
FRANK	FRANK12	12	1				1					2.00		
FRANK	FRANK8	8	1				1					2.00		
FRDRTH	FRDRTH	2	1				1				1			
FRDRTH	FRDRTHB3	50	1				1					1.17		
FRDRTH	FRDRTHB4	100	1				1				1			
FRDRTH	FRDRTHB7	1000	1				1				1			
GENRSN	GENT1000	1000		1.03			1						1.26	
GENRSN	GENT2B	2	1						1.29			1.14		
GENRSN	GENT50A	50		1.01			1						1.37	
GENRSN	GENT500A	500		1.03			1						1.26	
GOTTFR	GOTTFR	2		1.62			1					1.62		
GULF	GULFSH2	3		1.16			1						1.37	
HELIX	HELIX	3		1.38			1						1.50	
HILBRT	HILBRT12	12		2.00			1					2.00		
HILBRT	HILBR10A	10		2.00			1					2.00		
HIMLN3	HIMLN3	2		1.50			1						2.00	
HIMM1	HIMM1	2	1				1					2.00		
HIMM25	HIMM25	2	1				1					2.00		
HIMM27	HIMM27	2		1.27			1					1.27		
HIMM28	HIMM28	2	1				1				1			
HIMM29	HIMM29	2	1					1.06				1.06		
HIMM30	HIMM30	3	1				1				1			
HIMM32	HIMM32	4	1						1.43			1.29		
HIMM33	HIMM33A	2	1						1.50			1.25		
HYPCIR	HYPCIR	2	1				1				1			
JENSMP	JENSMP	2	1				1				1			
KOWOSB	KOWOSB1	4		1.50			1					1.50		
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1				1			
MEYER	MEYER	3		1.12			1						1.51	
MNSRF1	NMSURF64	36		1.16						f	1			
MNSRF1	NMSUR484	400	1							f		1.81		
MNSRF2	SNMSR484	400		1.05						f	1			
MNSRF2	SNMSUR64	36	1							f		1.21		
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MORBV998	998	1							f		1.33		
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
NONDIA	NONDIA10	10	1					1.05					1.26	
NONDIA	NONDIA20	20	1					1.05				1.05		
NONDIA	NONDI500	500	1					1.05				1.05		
NONDIA	NOND1000	1000	1					1.05					1.11	
OSBRN1	OSBRN1	5		1.82			1							f

OSBRN2	OSBRN2	11	1							f		1.23		
PENAL1	PEN1GM6	10	1					1.04					1.12	
PENAL1	PEN1LN1	50	1				1				1			
PENAL1	PEN1LN2	100	1				1				1			
PENAL1	PEN1LN3	1000	1				1				1			
PENAL2	PEN2GM1	50	1							f		1.50		
PENAL2	PEN2GM2	100				f				f	1			
PENAL2	PEN2GM6	4	1					1.04					1.05	
PENAL3	PENL3GM3	50	1				1					1.18		
PENAL3	PENL3GM4	100	1						1.27			1.09		
PENAL3	PENL3GM5	1000	1					1.23					1.38	
POWBSC	POWBSC	2	1						1.35			1.13		
POWBSC	POWBSC50	50	1							f				f
POWBSC	POWBS500	500	1							f				f
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1				1			
POWQUD	POWQUD8A	4	1				1				1			
POWSSQ	POWSSQ	2				f		1.95			1			
PWSING	PWSING4	4	1				1					1.06		
PWSING	PWSING60	60	1				1					1.06		
PWSING	PWSIN100	100	1				1					1.06		
PWSING	PWSI1000	1000	1				1					1.06		
QUARTC	QUARTC	25	1							f		1.11		
RECIPE	RECIPE	3		1.25				1.25			1			
ROSENB	ROSENB	2	1					1.05					1.15	
ROSENB	SHNRSN10	10	1						1.05			1.03		
SARSEB	SARSEB	4	1				1				1			
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1				1			
SCHMVT	SCHMV500	500	1				1					1.33		
SCHMVT	SCHV1000	1000	1				1				1			
SISSER	SISSER	2	1				1				1			
SQRTMX	MSQRTB49	49	1							f	1			
SQRTMX	MSQRTB9	9		1.14					1.43		1			
TDQUAD	TDQ10	10	1				1				1			
TDQUAD	TDQ1000	1000	1				1					2.00		
TDQUAD	TDQ500	500	1				1					2.00		
TOINT	PSPTOINT	50		1.27			1						1.45	
TOIN2	TOIN2	3	1				1					2.00		
TOIN4	TOIN4	4	1				1					3.00		
TRIDIA	TRIDIA10	10		2.00			1					2.00		
TRIDIA	TRLN100	100	1							f	1			
TRIDIA	TRLN1000	1000	1							f		2.00		
TRIGTO	TRIGT100	100	1					1.83			1			
TRIGTO	TRIGT50	50	1						2.00			1.22		
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1				1			
VAROSB	VAROSBG1	50	1				1					1.67		
VAROSB	VAROSBG2	100	1				1					2.00		
WATSON	WATSON12	12	1				1					1.50		
WATSON	WATSON6	6	1				1					1.09		
WOODS	WOODS	4	1				1					1.08		
WOODS	WOODS80	80	1				1					1.08		
XTX	XTX16	16	1				1					2.00		
XTX	XTX2	2	1				1				1			
ZANGWL	ZANGWL1	3	1				1					2.00		
Totals			113	25	0	3	90	24	10	17	57	62	17	5

Table C.5: Place finishes for number of function evaluations, tolerance = 10^{-3}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					2.00		
ARGQDO	ARGQO10	5	1				1					1.50		
ARGQDZ	ARGQDZ10	3	1				1					1.50		
ARTRIG	ARTRIG10	10	1					2.00			1			
AVRIEL	AVRIEL3	2	1				1					2.00		
BARD70	BARD70	3	1				1				1			
BEAL58	BEAL58KO	2	1					2.00			1			
BIGGS	BIGGS6	6		1.91			1							f
BOOTH	BOOTH	2	1				1					1.50		
BOX66	BOX662HL	2		1.12					3.50		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2	1					1.09					1.64	
BROWND	BROWND	4	1				1					1.33		
BROY7D	BROY7D	60				f		4.50			1			
BRWNAL	BRWNAL10	10	1				1					1.50		
BRWNAL	BRWNL100	100	1				1					2.00		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1				1			
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1					1.25			1			
BRYTRI	BRYTRI10	600	1				1					1.50		
CHEBYQ	CHEBYQ8	8	1					2.71			1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10		1.33					1.42		1			
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2	1						6.00			2.50		
CRGLVY	CRGLVY	4	1					1.12			1			
CRGLVY	CRGLVY10	10	1				1				1			
CRGLVY	CRGLY500	500	1				1					1.25		
CRGLVY	CRGL1000	1000	1				1					1.25		
DIX7DG	DIX7DGA	15	1						4.17			1.17		
DIXON	DIXON	10	1				1					2.00		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1					1.20		
ENGVL1	ENGVL1B4	100	1				1					1.20		
ENGVL1	ENGVL1B6	1000	1				1					1.40		
ENGVL2	ENGVL2	3		1.57					1.71		1			
EXTRSN	EXTRAR10	10		1.11					1.21		1			
EXTRSN	EXTRAR50	50	1					1.10					1.57	
EXTRSN	EXTRA100	100	1						1.10			1.05		
EXTRSN	EXTR1000	1000	1					1.10					1.48	
FRANK	FRANK8	8	1				1					2.50		
FRANK	FRANK12	12	1				1					3.00		
FRDRTH	FRDRTH	2		1.25				1.25			1			
FRDRTH	FRDRTHB3	50	1				1					1.75		
FRDRTH	FRDRTHB4	100	1				1					1.25		
FRDRTH	FRDRTHB7	1000	1				1					1.67		
GENRSN	GENT2B	2	1						4.00			1.20		
GENRSN	GENT50A	50		1.15			1						2.18	
GENRSN	GENT500A	500		1.23			1						2.13	
GENRSN	GENT1000	1000		1.21			1						2.15	
GOTTFR	GOTTFR	2	1					2.00			1			
GULF	GULFSH2	3	1						2.46			1.15		
HELIX	HELIX	3	1					1.29				1.29		
HILBRT	HILBR10A	10		1.50			1						2.00	
HILBRT	HILBRT12	12		1.50			1						2.00	
HIMLN3	HIMLN3	2			1.67			1.33			1			
HIMM1	HIMM1	2	1				1					1.50		
HIMM25	HIMM25	2	1				1					2.00		
HIMM27	HIMM27	2		1.14			1				1			
HIMM28	HIMM28	2	1					1.20				1.20		
HIMM29	HIMM29	2	1				1				1			
HIMM30	HIMM30	3	1				1				1			
HIMM32	HIMM32	4	1				1					1.29		
HIMM33	HIMM33A	2	1						2.67			1.33		
HYPCIR	HYPCIR	2	1					1.25				1.25		
JENSMP	JENSMP	2	1				1				1			
KOWOSB	KOWOSB1	4	1					1.89			1			
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1					2.00		
MEYER	MEYER	3		2.50			1				1			
MNSRF1	NMSURF64	36			11.47			1.47			1			
MNSRF1	NMSUR484	400			4.81		1					3.00		
MNSRF2	SNMSUR64	36		5.83						f	1			
MNSRF2	SNMSR484	400		4.16						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1				1					1.50		
NONDIA	NONDIA10	10	1					1.07					1.33	
NONDIA	NONDIA20	20		1.36					1.45		1			
NONDIA	NONDI500	500	1					1.14			1			
NONDIA	NOND1000	1000	1					1.33					1.58	
OSBRN1	OSBRN1	5		2.43			1							f

OSBRN2	OSBRN2	11	1							f	1			
PENAL1	PEN1GM6	10	1				1					1.17		
PENAL1	PEN1LN1	50	1				1					1.50		
PENAL1	PEN1LN2	100	1				1					1.67		
PENAL1	PEN1LN3	1000	1				1					2.00		
PENAL2	PEN2GM6	4	1				1				1			
PENAL2	PEN2GM1	50	1							f		2.00		
PENAL2	PEN2GM2	100	1							f		1.28		
PENAL3	PENL3GM3	50		1.08			1						1.23	
PENAL3	PENL3GM4	100		1.07					2.47		1			
PENAL3	PENL3GM5	1000	1						1.45			1.35		
POWBSC	POWBSC	2		1.20			1					1.20		
POWBSC	POWBSC50	50	1						67.00			1.50		
POWBSC	POWBS500	500	1						697.67			1.50		
POWBSC	POWB1000	1000	1							f		1.29		
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1					1.17		
POWQUD	POWQUD8A	4	1				1					1.50		
POWSSQ	POWSSQ	2				f	1					1.07		
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1					1.17		
PWSING	PWSIN100	100	1				1					1.17		
PWSING	PWSI1000	1000	1				1					1.33		
QUARTC	QUARTC	25	1				1					1.33		
RECIPE	RECIPE	3	1				1				1			
ROSENB	ROSENB	2	1					1.10					1.24	
ROSENB	SHNRSN10	10	1				1				1			
SARSEB	SARSEB	4	1				1					2.50		
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1					1.33		
SCHMVT	SCHMV500	500	1				1					1.33		
SCHMVT	SCHV1000	1000	1				1					1.67		
SISSER	SISSER	2	1				1					1.17		
SQRTMX	MSQRTB9	9		1.83					3.67		1			
SQRTMX	MSQRTB49	49	1					307.80			1			
TDQUAD	TDQ10	10	1				1					2.00		
TDQUAD	TDQ500	500	1				1					3.00		
TDQUAD	TDQ1000	1000	1				1					3.00		
TOIN2	TOIN2	3	1				1					1.50		
TOIN4	TOIN4	4	1				1					2.00		
TOINT	PSPTOINT	50			2.58		1					1.92		
TRIDIA	TRIDIA10	10	1				1					2.00		
TRIDIA	TRLN100	100	1							f		2.50		
TRIDIA	TRLN1000	1000	1							f		2.50		
TRIGTO	TRIGT50	50	1						5.00			1.08		
TRIGTO	TRIGT100	100		1.29					6.43		1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1					1.17		
VAROSB	VAROSBG1	50	1				1					2.50		
VAROSB	VAROSBG2	100	1				1					3.00		
WATSON	WATSON6	6	1				1				1			
WATSON	WATSON12	12	1				1				1			
WOODS	WOODS	4	1				1				1			
WOODS	WOODS80	80	1				1					1.40		
XTX	XTX2	2	1				1					1.50		
XTX	XTX16	16	1				1					2.00		
ZANGWL	ZANGWL1	3	1				1					3.00		
Totals			114	22	4	2	93	22	18	9	52	75	12	3

Table C.6: Place finishes for number of function evaluations, tolerance = 10^{-6}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					2.00		
ARGQDO	ARGQO10	5	1				1					1.50		
ARGQDZ	ARGQDZ10	3	1				1					1.50		
ARTRIG	ARTRIG10	10		1.11						f	1			
AVRIEL	AVRIEL3	2	1				1					2.00		
BARD70	BARD70	3	1				1				1			
BEAL58	BEAL58KO	2		1.11					2.00		1			
BIGGS	BIGGS6	6		1.44			1							f
BOOTH	BOOTH	2	1				1					2.00		
BOX66	BOX662HL	2		1.22					3.33		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2	1					1.25					1.58	
BROWND	BROWND	4	1				1					1.43		
BROY7D	BROY7D	60				f		3.73			1			
BRWNAL	BRWNAL10	10	1				1					1.67		
BRWNAL	BRWNL100	100	1							f		2.00		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1					1.17		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1				1				1			
BRYTRI	BRYTRI10	600	1				1					1.40		
CHEBYQ	CHEBYQ8	8	1					2.60			1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10		1.27					1.33		1			
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2	1						3.50			1.38		
CRGLVY	CRGLVY	4	1				1				1			
CRGLVY	CRGLVY10	10	1				1				1			
CRGLVY	CRGLY500	500	1				1					1.18		
CRGLVY	CRGL1000	1000	1				1					1.18		
DIX7DG	DIX7DGA	15	1						3.57			1.14		
DIXON	DIXON	10	1				1					2.00		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1					1.14		
ENGVL1	ENGVL1B4	100	1				1					1.33		
ENGVL1	ENGVL1B6	1000	1				1					1.33		
ENGVL2	ENGVL2	3		1.25					1.33		1			
EXTRSN	EXTRAR10	10	1					1.13			1			
EXTRSN	EXTRAR50	50	1					1.13					1.61	
EXTRSN	EXTRA100	100	1						1.13			1.09		
EXTRSN	EXTR1000	1000	1					1.13					1.52	
FRANK	FRANK8	8	1				1					2.50		
FRANK	FRANK12	12	1				1					3.00		
FRDRTH	FRDRTH	2	1				1				1			
FRDRTH	FRDRTHB3	50	1				1					1.80		
FRDRTH	FRDRTHB4	100	1				1					1.40		
FRDRTH	FRDRTHB7	1000	1				1					1.40		
GENRSN	GENT2B	2	1						3.14			1.14		
GENRSN	GENT50A	50		1.16			1						2.28	
GENRSN	GENT500A	500		1.23			1						2.14	
GENRSN	GENT1000	1000		1.21			1						2.15	
GOTTFR	GOTTFR	2	1						1.58			1.17		
GULF	GULFSH2	3	1						1.24			1.09		
HELIX	HELIX	3		1.10			1						1.30	
HILBRT	HILBR10A	10		1.50			1						2.00	
HILBRT	HILBRT12	12		1.50			1						2.00	
HIMLN3	HIMLN3	2		1.50			1				1			
HIMM1	HIMM1	2	1				1					1.50		
HIMM25	HIMM25	2	1				1					2.00		
HIMM27	HIMM27	2		1.10			1				1			
HIMM28	HIMM28	2	1					1.17				1.17		
HIMM29	HIMM29	2	1				1				1			
HIMM30	HIMM30	3	1				1					1.09		
HIMM32	HIMM32	4	1					1.25					1.62	
HIMM33	HIMM33A	2	1						2.25			1.25		
HYPCIR	HYPCIR	2	1					1.20				1.20		
JENSMP	JENSMP	2	1				1					1.12		
KOWOSB	KOWOSB1	4	1					1.64			1			
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1					1.67		
MEYER	MEYER	3		1.14			1						1.82	
MNSRF1	NMSURF64	36			10.00			1.41			1			
MNSRF1	NMSUR484	400			4.72		1					2.72		
MNSRF2	SNMSUR64	36		8.00						f	1			
MNSRF2	SNMSR484	400		4.10						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1							f		1.33		
NONDIA	NONDIA10	10	1					1.09					1.50	
NONDIA	NONDIA20	20		1.21					1.26		1			
NONDIA	NONDI500	500	1						1.14			1.05		
NONDIA	NOND1000	1000	1					1.20					1.35	
OSBRN1	OSBRN1	5		3.44			1							f

OSBRN2	OSBRN2	11	1							f	1			
PENAL1	PEN1GM6	10	1				1					1.10		
PENAL1	PEN1LN1	50	1				1					1.30		
PENAL1	PEN1LN2	100	1				1					1.40		
PENAL1	PEN1LN3	1000	1				1					1.60		
PENAL2	PEN2GM6	4	1				1				1			
PENAL2	PEN2GM1	50	1							f		1.88		
PENAL2	PEN2GM2	100				f				f	1			
PENAL3	PENL3GM3	50		1.07			1						1.13	
PENAL3	PENL3GM4	100		1.06					2.24		1			
PENAL3	PENL3GM5	1000	1						1.41			1.32		
POWBSC	POWBSC	2	1					1.06					1.16	
POWBSC	POWBSC50	50	1						39.27			1.18		
POWBSC	POWBS500	500	1						393.36			1.18		
POWBSC	POWB1000	1000	1							f		1.18		
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1					1.10		
POWQUD	POWQUD8A	4	1				1					1.50		
POWSSQ	POWSSQ	2				f		1.18			1			
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1					1.10		
PWSING	PWSIN100	100	1				1					1.20		
PWSING	PWSI1000	1000	1				1					1.20		
QUARTC	QUARTC	25	1				1					1.20		
RECIPE	RECIPE	3	1				1				1			
ROSENB	ROSENB	2	1					1.13					1.26	
ROSENB	SHNRSN10	10		1.14					1.24		1			
SARSEB	SARSEB	4	1				1					2.50		
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1					1.33		
SCHMVT	SCHMV500	500	1				1					1.67		
SCHMVT	SCHV1000	1000	1				1					1.67		
SISSER	SISSER	2	1				1					1.10		
SQRTMX	MSQRTB9	9		1.86					3.29		1			
SQRTMX	MSQRTB49	49		1.07						f	1			
TDQUAD	TDQ10	10	1				1					2.00		
TDQUAD	TDQ500	500	1				1					3.00		
TDQUAD	TDQ1000	1000	1				1					3.00		
TOIN2	TOIN2	3	1				1					2.00		
TOIN4	TOIN4	4	1				1					2.00		
TOINT	PSPTOINT	50			2.67		1					1.73		
TRIDIA	TRIDIA10	10	1				1					2.00		
TRIDIA	TRLN100	100	1							f		2.50		
TRIDIA	TRLN1000	1000	1							f		2.50		
TRIGTO	TRIGT50	50	1						5.00			1.15		
TRIGTO	TRIGT100	100		1.27					6.07		1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1					1.10		
VAROSB	VAROSBG1	50	1				1					2.00		
VAROSB	VAROSBG2	100	1				1					2.33		
WATSON	WATSON6	6	1				1					1.10		
WATSON	WATSON12	12	1				1					1.09		
WOODS	WOODS	4	1					1.02					1.09	
WOODS	WOODS80	80	1					1.02					1.16	
XTX	XTX2	2	1				1					1.50		
XTX	XTX16	16	1				1					2.00		
ZANGWL	ZANGWL1	3	1				1					3.00		
Totals			111	25	3	3	90	18	21	13	45	76	18	3

Table C.7: Place finishes for number of function evaluations, tolerance = 10^{-9}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					2.00		
ARGQDO	ARGQO10	5	1				1					1.50		
ARGQDZ	ARGQDZ10	3	1				1					1.50		
ARTRIG	ARTRIG10	10	1							f	1			
AVRIEL	AVRIEL3	2	1				1					2.00		
BARD70	BARD70	3	1					1.29			1			
BEAL58	BEAL58KO	2		1.10					1.90		1			
BIGGS	BIGGS6	6		1.39			1							f
BOOTH	BOOTH	2	1				1					2.00		
BOX66	BOX662HL	2		1.20					3.10		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2	1					1.25					1.67	
BROWND	BROWND	4	1				1					1.38		
BROY7D	BROY7D	60				f		3.56			1			
BRWNAL	BRWNAL10	10	1				1					1.50		
BRWNAL	BRWNL100	100	1							f		2.00		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1					1.14		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1					1.20			1			
BRYTRI	BRYTRI10	600	1				1					1.33		
CHEBYQ	CHEBYQ8	8	1					2.67			1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10		1.25					1.31		1			
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2	1						2.73			1.09		
CRGLVY	CRGLVY	4		1.06				1.06			1			
CRGLVY	CRGLVY10	10	1				1					1.14		
CRGLVY	CRGLY500	500	1				1					1.29		
CRGLVY	CRGL1000	1000	1				1					1.14		
DIX7DG	DIX7DGA	15	1						3.57			1.14		
DIXON	DIXON	10	1				1					2.00		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1					1.12		
ENGVL1	ENGVL1B4	100	1				1					1.29		
ENGVL1	ENGVL1B6	1000	1				1					1.29		
ENGVL2	ENGVL2	3		1.21					1.29		1			
EXTRSN	EXTRAR10	10	1					1.12			1			
EXTRSN	EXTRAR50	50	1					1.12					1.58	
EXTRSN	EXTRA100	100	1						1.12			1.08		
EXTRSN	EXTR1000	1000	1					1.12					1.50	
FRANK	FRANK8	8	1				1					2.50		
FRANK	FRANK12	12	1				1					3.00		
FRDRTH	FRDRTH	2		1.17				1.17			1			
FRDRTH	FRDRTHB3	50	1				1					1.50		
FRDRTH	FRDRTHB4	100	1				1					1.33		
FRDRTH	FRDRTHB7	1000	1				1					1.33		
GENRSN	GENT2B	2	1						2.88			1.12		
GENRSN	GENT50A	50		1.14			1						2.24	
GENRSN	GENT500A	500		1.23			1						2.13	
GENRSN	GENT1000	1000		1.21			1						2.15	
GOTTFR	GOTTFR	2	1						1.47			1.13		
GULF	GULFSH2	3	1						1.26			1.12		
HELIX	HELIX	3	1				1					1.25		
HILBRT	HILBR10A	10		1.50			1						2.00	
HILBRT	HILBRT12	12		1.50			1						2.00	
HIMLN3	HIMLN3	2			1.50		1					1.25		
HIMM1	HIMM1	2	1				1					1.50		
HIMM25	HIMM25	2	1				1					2.00		
HIMM27	HIMM27	2			1.33		1					1.08		
HIMM28	HIMM28	2	1					1.17				1.17		
HIMM29	HIMM29	2	1						1.18			1.06		
HIMM30	HIMM30	3	1				1					1.08		
HIMM32	HIMM32	4	1					1.22					1.56	
HIMM33	HIMM33A	2	1						1.80			1.20		
HYPCIR	HYPCIR	2	1						1.40			1.20		
JENSMP	JENSMP	2	1				1					1.11		
KOWOSB	KOWOSB1	4	1					1.58			1			
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1					1.67		
MEYER	MEYER	3		1.03			1						1.51	
MNSRF1	NMSURF64	36		9.25						f	1			
MNSRF1	NMSUR484	400		1.73						f	1			
MNSRF2	SNMSUR64	36		7.74						f	1			
MNSRF2	SNMSR484	400		4.08						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1							f		1.33		
NONDIA	NONDIA10	10	1					1.13					1.52	
NONDIA	NONDIA20	20		1.20					1.30		1			
NONDIA	NONDI500	500	1						1.18			1.05		
NONDIA	NOND1000	1000	1					1.24					1.33	
OSBRN1	OSBRN1	5		1.78			1							f

OSBRN2	OSBRN2	11	1							f	1			
PENAL1	PEN1GM6	10	1				1					1.17		
PENAL1	PEN1LN1	50	1				1					1.21		
PENAL1	PEN1LN2	100	1				1					1.29		
PENAL1	PEN1LN3	1000	1				1					1.50		
PENAL2	PEN2GM6	4		1.04					1.21		1			
PENAL2	PEN2GM1	50	1							f		1.78		
PENAL2	PEN2GM2	100				f				f	1			
PENAL3	PENL3GM3	50	1				1					1.12		
PENAL3	PENL3GM4	100		1.06					2.17		1			
PENAL3	PENL3GM5	1000	1						1.35			1.26		
POWBSC	POWBSC	2	1						1.15			1.06		
POWBSC	POWBSC50	50		5.53						f	1			
POWBSC	POWBS500	500		4.59						f	1			
POWBSC	POWB1000	1000		5.71						f	1			
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1					1.07		
POWQUD	POWQUD8A	4	1				1					1.50		
POWSSQ	POWSSQ	2				f		4.00			1			
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1					1.14		
PWSING	PWSIN100	100	1				1					1.14		
PWSING	PWSI1000	1000	1				1					1.21		
QUARTC	QUARTC	25	1				1					1.14		
RECIPE	RECIPE	3	1				1				1			
ROSENB	ROSENB	2	1					1.12					1.25	
ROSENB	SHNRSN10	10		1.04					1.24		1			
SARSEB	SARSEB	4	1				1					2.50		
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1					1.25		
SCHMVT	SCHMV500	500	1				1					1.25		
SCHMVT	SCHV1000	1000	1				1					1.50		
SISSER	SISSER	2	1				1					1.07		
SQRTMX	MSQRTB9	9		1.62					3.00		1			
SQRTMX	MSQRTB49	49		1.07						f	1			
TDQUAD	TDQ10	10	1				1					2.00		
TDQUAD	TDQ500	500	1				1					3.00		
TDQUAD	TDQ1000	1000	1				1					3.00		
TOIN2	TOIN2	3	1				1					2.00		
TOIN4	TOIN4	4	1				1					2.00		
TOINT	PSPTOINT	50			2.56		1					1.69		
TRIDIA	TRIDIA10	10		1.50			1						2.00	
TRIDIA	TRLN100	100	1							f		2.50		
TRIDIA	TRLN1000	1000	1							f		2.50		
TRIGTO	TRIGT50	50	1						4.71			1.14		
TRIGTO	TRIGT100	100		1.33					6.07		1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1					1.07		
VAROSB	VAROSBG1	50	1				1					2.00		
VAROSB	VAROSBG2	100	1				1					2.33		
WATSON	WATSON6	6	1				1					1.09		
WATSON	WATSON12	12	1				1					1.33		
WOODS	WOODS	4	1					1.02					1.11	
WOODS	WOODS80	80	1					1.02					1.15	
XTX	XTX2	2	1				1					1.50		
XTX	XTX16	16	1				1					2.00		
ZANGWL	ZANGWL1	3	1				1					3.00		
Totals			107	29	3	3	83	19	23	17	45	78	16	3

Table C.8: Place finishes for number of function evaluations, tolerance = 10^{-12}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					2.00		
ARGQDO	ARGQO10	5	1				1					1.50		
ARGQDZ	ARGQDZ10	3	1				1					1.50		
ARTRIG	ARTRIG10	10	1							f	1			
AVRIEL	AVRIEL3	2	1				1					2.00		
BARD70	BARD70	3	1					1.12			1			
BEAL58	BEAL58KO	2		1.09					1.82		1			
BIGGS	BIGGS6	6		1.38			1							f
BOOTH	BOOTH	2	1				1					2.00		
BOX66	BOX662HL	2		1.18					2.91		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2	1					1.15					1.54	
BROWND	BROWND	4	1				1					1.38		
BROY7D	BROY7D	60				f		3.47			1			
BRWNAL	BRWNAL10	10	1				1					1.43		
BRWNAL	BRWNL100	100	1							f		4.00		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1				1			
BRYTRI	BRYTRI10	600	1				1					1.33		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1				1				1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHEBYQ	CHEBYQ8	8	1					2.50			1			
CHNRSN	CHNRSH10	10		1.31					1.38		1			
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2		1.08					2.58		1			
CRGLVY	CRGLVY	4		1.04				1.04			1			
CRGLVY	CRGLVY10	10	1				1					1.06		
CRGLVY	CRGLY500	500	1				1					1.40		
CRGLVY	CRGL1000	1000	1				1					1.20		
DIXON	DIXON	10	1				1					2.00		
DIX7DG	DIX7DGA	15	1					3.25			1			
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1					1.12		
ENGVL1	ENGVL1B4	100	1				1					1.25		
ENGVL1	ENGVL1B6	1000	1				1					1.25		
ENGVL2	ENGVL2	3		1.29					1.36		1			
EXTRSN	EXTRAR10	10		1.04					1.17		1			
EXTRSN	EXTRAR50	50	1					1.12					1.52	
EXTRSN	EXTRA100	100	1						1.12			1.08		
EXTRSN	EXTR1000	1000	1					1.12					1.48	
FRANK	FRANK12	12	1				1					3.00		
FRANK	FRANK8	8	1				1					2.50		
FRDRTH	FRDRTH	2	1				1				1			
FRDRTH	FRDRTHB3	50	1				1					1.43		
FRDRTH	FRDRTHB4	100	1				1					1.14		
FRDRTH	FRDRTHB7	1000	1				1					1.33		
GENRSN	GENT1000	1000		1.21			1						2.15	
GENRSN	GENT2B	2	1						2.88			1.25		
GENRSN	GENT50A	50		1.14			1						2.23	
GENRSN	GENT500A	500		1.23			1						2.13	
GOTTFR	GOTTFR	2	1						1.44			1.12		
GULF	GULFSH2	3	1						1.26			1.11		
HELIX	HELIX	3	1				1					1.33		
HILBRT	HILBRT12	12		1.50			1						2.00	
HILBRT	HILBR10A	10		1.50			1						2.00	
HIMLN3	HIMLN3	2			1.50		1					1.25		
HIMM1	HIMM1	2	1				1					1.50		
HIMM25	HIMM25	2	1				1					2.00		
HIMM27	HIMM27	2			1.31		1					1.15		
HIMM28	HIMM28	2	1					1.14				1.14		
HIMM29	HIMM29	2	1						1.21			1.11		
HIMM30	HIMM30	3	1				1				1			
HIMM32	HIMM32	4	1					1.33					1.56	
HIMM33	HIMM33A	2	1						2.00			1.20		
HYPCIR	HYPCIR	2	1					1.17				1.17		
JENSMP	JENSMP	2	1				1				1			
KOWOSB	KOWOSB1	4	1					1.58			1			
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1					1.50		
MEYER	MEYER	3		1.04			1						1.50	
MNSRF1	NMSURF64	36		9.25						f	1			
MNSRF1	NMSUR484	400		1.74						f	1			
MNSRF2	SNMSR484	400		4.04						f	1			
MNSRF2	SNMSUR64	36		7.50						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MORBV998	998	1							f		1.25		
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
NONDIA	NONDIA10	10	1					1.12					1.50	
NONDIA	NONDIA20	20		1.19					1.29		1			
NONDIA	NONDI500	500	1						1.17			1.04		
NONDIA	NOND1000	1000	1					1.23					1.32	
OSBRN1	OSBRN1	5		1.79			1							f

OSBRN2	OSBRN2	11	1							f	1			
PENAL1	PEN1GM6	10	1					1.14					1.29	
PENAL1	PEN1LN1	50	1				1					1.18		
PENAL1	PEN1LN2	100	1				1					1.22		
PENAL1	PEN1LN3	1000	1				1					1.32		
PENAL2	PEN2GM1	50	1							f		1.78		
PENAL2	PEN2GM2	100				f				f	1			
PENAL2	PEN2GM6	4		1.08					1.25		1			
PENAL3	PENL3GM3	50		1.06			1						1.19	
PENAL3	PENL3GM4	100		1.06					2.22		1			
PENAL3	PENL3GM5	1000	1						1.33			1.25		
POWBSC	POWBSC	2	1						1.58			1.10		
POWBSC	POWBSC50	50	1							f				f
POWBSC	POWBS500	500	1							f				f
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1					1.05		
POWQUD	POWQUD8A	4	1				1					1.50		
POWSSQ	POWSSQ	2				f		2.33			1			
PWSING	PWSING4	4	1				1					1.06		
PWSING	PWSING60	60	1				1					1.11		
PWSING	PWSIN100	100	1				1					1.11		
PWSING	PWSI1000	1000	1				1					1.17		
QUARTC	QUARTC	25	1							f		1.11		
RECIPE	RECIPE	3	1				1				1			
ROSENB	ROSENB	2	1					1.12					1.24	
ROSENB	SHNRSN10	10		1.12					1.37		1			
SARSEB	SARSEB	4	1				1					2.50		
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1					1.25		
SCHMVT	SCHMV500	500	1				1					1.50		
SCHMVT	SCHV1000	1000	1				1					1.50		
SISSER	SISSER	2	1				1				1			
SQRTMX	MSQRTB49	49		1.06						f	1			
SQRTMX	MSQRTB9	9		1.56					2.67		1			
TDQUAD	TDQ10	10	1				1					2.00		
TDQUAD	TDQ1000	1000	1				1					3.00		
TDQUAD	TDQ500	500	1				1					3.00		
TOINT	PSPTOINT	50			2.41		1					1.59		
TOIN2	TOIN2	3	1				1					2.00		
TOIN4	TOIN4	4	1				1					2.00		
TRIDIA	TRIDIA10	10		1.50			1						2.00	
TRIDIA	TRLN100	100	1							f		1.67		
TRIDIA	TRLN1000	1000	1							f		2.50		
TRIGTO	TRIGT100	100		1.25					5.69		1			
TRIGTO	TRIGT50	50	1						4.71			1.14		
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1					1.05		
VAROSB	VAROSBG1	50	1				1					1.75		
VAROSB	VAROSBG2	100	1				1					2.00		
WATSON	WATSON12	12	1				1					1.62		
WATSON	WATSON6	6	1				1					1.08		
WOODS	WOODS	4	1					1.02					1.09	
WOODS	WOODS80	80	1					1.02					1.15	
XTX	XTX16	16	1				1					2.00		
XTX	XTX2	2	1				1					1.50		
ZANGWL	ZANGWL1	3	1				1					3.00		
Totals			107	28	3	3	83	19	22	17	47	71	18	5

Table C.9: Place finishes for number of gradient evaluations, tolerance = 10^{-3}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					1.50		
ARGQDO	ARGQO10	5	1				1					1.50		
ARGQDZ	ARGQDZ10	3	1				1					1.50		
ARTRIG	ARTRIG10	10	1					2.29			1			
AVRIEL	AVRIEL3	2	1				1					1.50		
BARD70	BARD70	3	1				1				1			
BEAL58	BEAL58KO	2	1						3.20			1.20		
BIGGS	BIGGS6	6		1.82			1							f
BOOTH	BOOTH	2	1				1				1			
BOX66	BOX662HL	2		1.50					4.67		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2		1.17					2.00		1			
BROWND	BROWND	4	1				1					1.17		
BROY7D	BROY7D	60				f		7.71			1			
BRWNAL	BRWNAL10	10	1				1					1.50		
BRWNAL	BRWNL100	100	1				1					1.50		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1				1			
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1						1.67			1.33		
BRYTRI	BRYTRI10	600	1					1.33				1.33		
CHEBYQ	CHEBYQ8	8		1.18					3.45		1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10	1						1.89			1.11		
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2	1						6.00			1.75		
CRGLVY	CRGLVY	4	1					1.29			1			
CRGLVY	CRGLVY10	10	1				1				1			
CRGLVY	CRGLY500	500	1				1					1.25		
CRGLVY	CRGL1000	1000	1				1					1.12		
DIX7DG	DIX7DGA	15	1						5.00			1.20		
DIXON	DIXON	10	1				1					1.50		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1				1			
ENGVL1	ENGVL1B4	100	1				1				1			
ENGVL1	ENGVL1B6	1000	1				1				1			
ENGVL2	ENGVL2	3		1.83					2.00		1			
EXTRSN	EXTRAR10	10	1					1.35			1			
EXTRSN	EXTRAR50	50	1						1.35			1.24		
EXTRSN	EXTRA100	100	1						1.35			1.06		
EXTRSN	EXTR1000	1000	1						1.35			1.12		
FRANK	FRANK8	8	1				1					1.50		
FRANK	FRANK12	12	1				1					1.50		
FRDRTH	FRDRTH	2		1.25				1.25			1			
FRDRTH	FRDRTHB3	50	1				1					1.25		
FRDRTH	FRDRTHB4	100	1				1				1			
FRDRTH	FRDRTHB7	1000	1				1				1			
GENRSN	GENT2B	2	1					4.00			1			
GENRSN	GENT50A	50	1					1.04					1.25	
GENRSN	GENT500A	500	1					1.02					1.15	
GENRSN	GENT1000	1000	1					1.01					1.15	
GOTTFR	GOTTFR	2	1					2.80			1			
GULF	GULFSH2	3		1.50					4.00		1			
HELIX	HELIX	3		1.17					1.50		1			
HILBRT	HILBR10A	10		1.50			1					1.50		
HILBRT	HILBRT12	12		1.50			1					1.50		
HIMLN3	HIMLN3	2		1.33				1.33			1			
HIMM1	HIMM1	2	1				1					1.50		
HIMM25	HIMM25	2	1				1					1.50		
HIMM27	HIMM27	2		1.14			1				1			
HIMM28	HIMM28	2	1					1.20			1			
HIMM29	HIMM29	2	1				1				1			
HIMM30	HIMM30	3		1.14					1.29		1			
HIMM32	HIMM32	4	1					1.17			1			
HIMM33	HIMM33A	2	1						2.67			1.33		
HYPCIR	HYPCIR	2	1					1.25			1			
JENSMP	JENSMP	2	1				1				1			
KOWOSB	KOWOSB1	4	1					2.43			1			
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1				1			
MEYER	MEYER	3			1.67			1.33			1			
MNSRF1	NMSURF64	36			10.67			1.87			1			
MNSRF1	NMSUR484	400		1.64			1						2.79	
MNSRF2	SNMSUR64	36		2.65						f	1			
MNSRF2	SNMSR484	400		1.15						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1				1					1.50		
NONDIA	NONDIA10	10	1						1.45			1.18		
NONDIA	NONDIA20	20	1					1.45			1			
NONDIA	NONDI500	500	1					1.45			1			
NONDIA	NOND1000	1000	1					1.33			1			
OSBRN1	OSBRN1	5		1.29			1							f

OSBRN2	OSBRN2	11	1							f		1.33		
PENAL1	PEN1GM6	10	1				1				1			
PENAL1	PEN1LN1	50	1				1				1			
PENAL1	PEN1LN2	100	1				1				1			
PENAL1	PEN1LN3	1000	1				1				1			
PENAL2	PEN2GM6	4	1				1				1			
PENAL2	PEN2GM1	50	1							f		1.33		
PENAL2	PEN2GM2	100		1.19						f	1			
PENAL3	PENL3GM3	50		1.09					1.18		1			
PENAL3	PENL3GM4	100		1.30					3.70		1			
PENAL3	PENL3GM5	1000		1.06					1.81		1			
POWBSC	POWBSC	2	1					1.25				1.25		
POWBSC	POWBSC50	50	1						67.00			1.17		
POWBSC	POWBS500	500		1.20					837.20		1			
POWBSC	POWB1000	1000		1.40						f	1			
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1				1			
POWQUD	POWQUD8A	4	1				1				1			
POWSSQ	POWSSQ	2				f		1.50			1			
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1				1			
PWSING	PWSIN100	100	1				1				1			
PWSING	PWSI1000	1000	1				1				1			
QUARTC	QUARTC	25	1				1					1.33		
RECIPE	RECIPE	3		1.33				1.33			1			
ROSENB	ROSENB	2	1						1.35			1.12		
ROSENB	SHNRSN10	10	1				1				1			
SARSEB	SARSEB	4	1				1				1			
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1				1			
SCHMVT	SCHMV500	500	1				1				1			
SCHMVT	SCHV1000	1000	1				1				1			
SISSER	SISSER	2	1				1					1.17		
SQRTMX	MSQRTB9	9		1.60					4.40		1			
SQRTMX	MSQRTB49	49		1.29					439.71		1			
TDQUAD	TDQ10	10	1				1				1			
TDQUAD	TDQ500	500	1				1					1.50		
TDQUAD	TDQ1000	1000	1				1					1.50		
TOIN2	TOIN2	3	1				1				1			
TOIN4	TOIN4	4	1				1					2.00		
TOINT	PSPTOINT	50	1					1.09					1.18	
TRIDIA	TRIDIA10	10	1				1					1.50		
TRIDIA	TRLN100	100	1							f		1.50		
TRIDIA	TRLN1000	1000	1							f		1.50		
TRIGTO	TRIGT50	50		1.30					6.50		1			
TRIGTO	TRIGT100	100		1.64					8.18		1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1				1			
VAROSB	VAROSBG1	50	1				1					2.00		
VAROSB	VAROSBG2	100	1				1					2.50		
WATSON	WATSON6	6	1				1				1			
WATSON	WATSON12	12	1				1				1			
WOODS	WOODS	4	1				1				1			
WOODS	WOODS80	80	1				1				1			
XTX	XTX2	2	1				1				1			
XTX	XTX16	16	1				1					1.50		
ZANGWL	ZANGWL1	3	1				1					1.50		
Totals			110	28	2	2	81	25	27	9	88	46	5	3

Table C.10: Place finishes for number of gradient evaluations, tolerance = 10^{-6}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					1.50		
ARGQDO	ARGQO10	5	1				1					1.50		
ARGQDZ	ARGQDZ10	3	1				1					1.50		
ARTRIG	ARTRIG10	10		1.12						f	1			
AVRIEL	AVRIEL3	2	1				1					1.50		
BARD70	BARD70	3	1				1				1			
BEAL58	BEAL58KO	2	1					2.57			1			
BIGGS	BIGGS6	6		1.29			1							f
BOOTH	BOOTH	2	1				1					1.50		
BOX66	BOX662HL	2		1.57					4.29		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2		1.14					2.14		1			
BROWND	BROWND	4	1				1					1.29		
BROY7D	BROY7D	60				f		5.60			1			
BRWNAL	BRWNAL10	10	1				1					1.33		
BRWNAL	BRWNL100	100	1							f		1.50		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1					1.17		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1					1.25				1.25		
BRYTRI	BRYTRI10	600	1					1.25				1.25		
CHEBYQ	CHEBYQ8	8		1.17					3.25		1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10	1						1.67			1.08		
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2	1					3.50			1			
CRGLVY	CRGLVY	4	1					1.08			1			
CRGLVY	CRGLVY10	10	1				1				1			
CRGLVY	CRGLY500	500	1				1					1.18		
CRGLVY	CRGL1000	1000	1				1					1.09		
DIX7DG	DIX7DGA	15	1						4.17			1.17		
DIXON	DIXON	10	1				1					1.50		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1				1			
ENGVL1	ENGVL1B4	100	1				1					1.17		
ENGVL1	ENGVL1B6	1000	1				1				1			
ENGVL2	ENGVL2	3		1.36					1.45		1			
EXTRSN	EXTRAR10	10	1						1.37			1.11		
EXTRSN	EXTRAR50	50	1						1.37			1.32		
EXTRSN	EXTRA100	100	1						1.37			1.11		
EXTRSN	EXTR1000	1000	1						1.37			1.21		
FRANK	FRANK8	8	1				1					1.50		
FRANK	FRANK12	12	1				1					1.50		
FRDRTH	FRDRTH	2	1				1				1			
FRDRTH	FRDRTHB3	50	1				1					1.40		
FRDRTH	FRDRTHB4	100	1				1					1.20		
FRDRTH	FRDRTHB7	1000	1				1				1			
GENRSN	GENT2B	2	1					3.14			1			
GENRSN	GENT50A	50	1					1.02					1.26	
GENRSN	GENT500A	500	1					1.03					1.16	
GENRSN	GENT1000	1000	1					1.02					1.15	
GOTTFR	GOTTFR	2	1					1.90			1			
GULF	GULFSH2	3		1.17					1.71		1			
HELIX	HELIX	3		1.10			1				1			
HILBRT	HILBR10A	10		1.50			1					1.50		
HILBRT	HILBRT12	12		1.50			1					1.50		
HIMLN3	HIMLN3	2		1.25			1				1			
HIMM1	HIMM1	2	1				1					1.50		
HIMM25	HIMM25	2	1				1					1.50		
HIMM27	HIMM27	2		1.10			1				1			
HIMM28	HIMM28	2	1					1.17			1			
HIMM29	HIMM29	2	1				1				1			
HIMM30	HIMM30	3	1					1.10			1			
HIMM32	HIMM32	4	1						1.43			1.29		
HIMM33	HIMM33A	2	1						2.25			1.25		
HYPCIR	HYPCIR	2	1					1.20			1			
JENSMP	JENSMP	2	1				1					1.12		
KOWOSB	KOWOSB1	4	1					2.00			1			
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1				1			
MEYER	MEYER	3	1					1.28					1.45	
MNSRF1	NMSURF64	36			9.00			1.72			1			
MNSRF1	NMSUR484	400		1.60			1						2.53	
MNSRF2	SNMSUR64	36		3.05						f	1			
MNSRF2	SNMSR484	400		1.14						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1							f		1.33		
NONDIA	NONDIA10	10	1						1.33			1.22		
NONDIA	NONDIA20	20	1						1.33			1.06		
NONDIA	NONDI500	500	1						1.33			1.06		
NONDIA	NOND1000	1000	1						1.26			1.05		
OSBRN1	OSBRN1	5		2.33			1							f

OSBRN2	OSBRN2	11	1							f		1.25		
PENAL1	PEN1GM6	10	1				1				1			
PENAL1	PEN1LN1	50	1				1				1			
PENAL1	PEN1LN2	100	1				1				1			
PENAL1	PEN1LN3	1000	1				1				1			
PENAL2	PEN2GM6	4	1				1				1			
PENAL2	PEN2GM1	50	1							f		1.29		
PENAL2	PEN2GM2	100				f				f	1			
PENAL3	PENL3GM3	50		1.17					1.25		1			
PENAL3	PENL3GM4	100		1.25					3.17		1			
PENAL3	PENL3GM5	1000		1.06					1.72		1			
POWBSC	POWBSC	2	1						1.31			1.19		
POWBSC	POWBSC50	50	1					39.27			1			
POWBSC	POWBS500	500		1.38					540.88		1			
POWBSC	POWB1000	1000		1.38						f	1			
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1				1			
POWQUD	POWQUD8A	4	1				1				1			
POWSSQ	POWSSQ	2				f		1.82			1			
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1				1			
PWSING	PWSIN100	100	1				1					1.10		
PWSING	PWSI1000	1000	1				1				1			
QUARTC	QUARTC	25	1				1					1.20		
RECIPE	RECIPE	3		1.25				1.25			1			
ROSENB	ROSENB	2	1						1.37			1.16		
ROSENB	SHNRSN10	10	1					1.24			1			
SARSEB	SARSEB	4	1				1				1			
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1				1			
SCHMVT	SCHMV500	500	1				1					1.33		
SCHMVT	SCHV1000	1000	1				1				1			
SISSER	SISSER	2	1				1					1.10		
SQRTMX	MSQRTB9	9		1.67					3.83		1			
SQRTMX	MSQRTB49	49		1.27						f	1			
TDQUAD	TDQ10	10	1				1				1			
TDQUAD	TDQ500	500	1				1					1.50		
TDQUAD	TDQ1000	1000	1				1					1.50		
TOIN2	TOIN2	3	1				1					1.50		
TOIN4	TOIN4	4	1				1					2.00		
TOINT	PSPTOINT	50		1.07			1					1.07		
TRIDIA	TRIDIA10	10	1				1					1.50		
TRIDIA	TRLN100	100	1							f		1.50		
TRIDIA	TRLN1000	1000	1							f		1.50		
TRIGTO	TRIGT50	50		1.18					5.91		1			
TRIGTO	TRIGT100	100		1.58					7.58		1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1				1			
VAROSB	VAROSBG1	50	1				1					1.67		
VAROSB	VAROSBG2	100	1				1					2.00		
WATSON	WATSON6	6	1				1					1.10		
WATSON	WATSON12	12	1				1					1.09		
WOODS	WOODS	4	1						1.25			1.06		
WOODS	WOODS80	80	1						1.25			1.06		
XTX	XTX2	2	1				1				1			
XTX	XTX16	16	1				1					1.50		
ZANGWL	ZANGWL1	3	1				1					1.50		
Totals			111	27	1	3	80	21	28	13	73	61	5	3

Table C.11: Place finishes for number of gradient evaluations, tolerance = 10^{-9}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					1.50		
ARGQDO	ARGQO10	5	1				1					1.50		
ARGQDZ	ARGQDZ10	3	1				1					1.50		
ARTRIG	ARTRIG10	10	1							f	1			
AVRIEL	AVRIEL3	2	1				1					1.50		
BARD70	BARD70	3	1					1.29			1			
BEAL58	BEAL58KO	2	1					2.38			1			
BIGGS	BIGGS6	6		1.24			1							f
BOOTH	BOOTH	2	1				1					1.50		
BOX66	BOX662HL	2		1.50					3.88		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2	1					1.88			1			
BROWND	BROWND	4	1				1					1.25		
BROY7D	BROY7D	60				f		5.18			1			
BRWNAL	BRWNAL10	10	1				1					1.17		
BRWNAL	BRWNL100	100	1							f		1.50		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1					1.14		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1						1.50			1.25		
BRYTRI	BRYTRI10	600	1					1.20				1.20		
CHEBYQ	CHEBYQ8	8		1.17					3.33		1			
CHEBYQ	CHEBYQ10	10	1							f				f
CHNRSN	CHNRSH10	10	1						1.62			1.08		
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2		1.22					3.33		1			
CRGLVY	CRGLVY	4		1.06					1.12		1			
CRGLVY	CRGLVY10	10	1				1					1.14		
CRGLVY	CRGLY500	500	1				1					1.29		
CRGLVY	CRGL1000	1000	1				1					1.07		
DIX7DG	DIX7DGA	15	1						4.17			1.17		
DIXON	DIXON	10	1				1					1.50		
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1				1			
ENGVL1	ENGVL1B4	100	1				1					1.14		
ENGVL1	ENGVL1B6	1000	1				1				1			
ENGVL2	ENGVL2	3		1.31					1.38		1			
EXTRSN	EXTRAR10	10	1						1.35			1.10		
EXTRSN	EXTRAR50	50	1						1.35			1.30		
EXTRSN	EXTRA100	100	1						1.35			1.10		
EXTRSN	EXTR1000	1000	1						1.35			1.20		
FRANK	FRANK8	8	1				1					1.50		
FRANK	FRANK12	12	1				1					1.50		
FRDRTH	FRDRTH	2		1.17				1.17			1			
FRDRTH	FRDRTHB3	50	1				1					1.17		
FRDRTH	FRDRTHB4	100	1				1					1.17		
FRDRTH	FRDRTHB7	1000	1				1				1			
GENRSN	GENT2B	2	1					2.88			1			
GENRSN	GENT50A	50	1					1.03					1.26	
GENRSN	GENT500A	500	1					1.03					1.16	
GENRSN	GENT1000	1000	1					1.02					1.15	
GOTTFR	GOTTFR	2	1					1.69			1			
GULF	GULFSH2	3		1.12					1.65		1			
HELIX	HELIX	3	1				1				1			
HILBRT	HILBR10A	10		1.50			1					1.50		
HILBRT	HILBRT12	12		1.50			1					1.50		
HIMLN3	HIMLN3	2		1.25			1					1.25		
HIMM1	HIMM1	2	1				1					1.50		
HIMM25	HIMM25	2	1				1					1.50		
HIMM27	HIMM27	2			1.17		1					1.08		
HIMM28	HIMM28	2	1					1.17			1			
HIMM29	HIMM29	2	1					1.25			1			
HIMM30	HIMM30	3	1					1.09			1			
HIMM32	HIMM32	4	1						1.38			1.25		
HIMM33	HIMM33A	2	1						1.80			1.20		
HYPCIR	HYPCIR	2	1					1.40			1			
JENSMP	JENSMP	2	1				1					1.11		
KOWOSB	KOWOSB1	4	1					1.90			1			
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1				1			
MEYER	MEYER	3	1						1.36			1.35		
MNSRF1	NMSURF64	36		8.20						f	1			
MNSRF1	NMSUR484	400	1							f		1.57		
MNSRF2	SNMSUR64	36		2.96						f	1			
MNSRF2	SNMSR484	400		1.15						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
MOREBV	MORBV998	998	1							f		1.33		
NONDIA	NONDIA10	10	1						1.37			1.26		
NONDIA	NONDIA20	20	1						1.37			1.05		
NONDIA	NONDI500	500	1						1.37			1.05		
NONDIA	NOND1000	1000	1						1.30			1.05		
OSBRN1	OSBRN1	5		1.30			1							f

OSBRN2	OSBRN2	11	1							f		1.23		
PENAL1	PEN1GM6	10	1				1					1.08		
PENAL1	PEN1LN1	50	1				1				1			
PENAL1	PEN1LN2	100	1				1				1			
PENAL1	PEN1LN3	1000	1				1					1.07		
PENAL2	PEN2GM6	4	1						1.50			1.06		
PENAL2	PEN2GM1	50	1							f		1.25		
PENAL2	PEN2GM2	100				f				f	1			
PENAL3	PENL3GM3	50		1.08					1.23		1			
PENAL3	PENL3GM4	100		1.23					3.00		1			
PENAL3	PENL3GM5	1000		1.11					1.72		1			
POWBSC	POWBSC	2	1						1.41			1.11		
POWBSC	POWBSC50	50		5.38						f	1			
POWBSC	POWBS500	500		5.58						f	1			
POWBSC	POWB1000	1000		7.33						f	1			
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1				1			
POWQUD	POWQUD8A	4	1				1				1			
POWSSQ	POWSSQ	2				f		6.00			1			
PWSING	PWSING4	4	1				1				1			
PWSING	PWSING60	60	1				1					1.07		
PWSING	PWSIN100	100	1				1					1.07		
PWSING	PWSI1000	1000	1				1					1.07		
QUARTC	QUARTC	25	1				1					1.14		
RECIPE	RECIPE	3		1.20				1.20			1			
ROSENB	ROSENB	2	1						1.35			1.15		
ROSENB	SHNRSN10	10	1						1.49			1.07		
SARSEB	SARSEB	4	1				1				1			
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1				1			
SCHMVT	SCHMV500	500	1				1				1			
SCHMVT	SCHV1000	1000	1				1				1			
SISSER	SISSER	2	1				1					1.07		
SQRTMX	MSQRTB9	9		1.43					3.43		1			
SQRTMX	MSQRTB49	49		1.25						f	1			
TDQUAD	TDQ10	10	1				1				1			
TDQUAD	TDQ500	500	1				1					1.50		
TDQUAD	TDQ1000	1000	1				1					1.50		
TOIN2	TOIN2	3	1				1					1.50		
TOIN4	TOIN4	4	1				1					2.00		
TOINT	PSPTOINT	50		1.06			1					1.06		
TRIDIA	TRIDIA10	10		1.50			1					1.50		
TRIDIA	TRLN100	100	1							f		1.50		
TRIDIA	TRLN1000	1000	1							f		1.50		
TRIGTO	TRIGT50	50		1.17					5.50		1			
TRIGTO	TRIGT100	100		1.67					7.58		1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1				1			
VAROSB	VAROSBG1	50	1				1					1.67		
VAROSB	VAROSBG2	100	1				1					2.00		
WATSON	WATSON6	6	1				1					1.09		
WATSON	WATSON12	12	1				1					1.25		
WOODS	WOODS	4	1						1.24			1.08		
WOODS	WOODS80	80	1						1.24			1.05		
XTX	XTX2	2	1				1				1			
XTX	XTX16	16	1				1					1.50		
ZANGWL	ZANGWL1	3	1				1					1.50		
Totals			110	28	1	3	75	18	32	17	65	71	3	3

Table C.12: Place finishes for number of gradient evaluations, tolerance = 10^{-12}.

Function	Problem	Size	UMINH				E04LBF				DMNH			
			1	2	3	f	1	2	3	f	1	2	3	f
ARGAUS	ARGAUS	3	1				1				1			
ARGQDN	ARGQDN50	5	1				1					1.50		
ARGQDO	ARGQO10	5	1				1					1.50		
ARGQDZ	ARGQDZ10	3	1				1					1.50		
ARTRIG	ARTRIG10	10	1							f	1			
AVRIEL	AVRIEL3	2	1				1					1.50		
BARD70	BARD70	3	1					1.12			1			
BEAL58	BEAL58KO	2	1					2.22			1			
BIGGS	BIGGS6	6		1.23			1							f
BOOTH	BOOTH	2	1				1					1.50		
BOX66	BOX662HL	2		1.44					3.56		1			
BRKMCC	BRKMCC	2	1				1				1			
BROWNB	BROWNB	2		1.12					1.88		1			
BROWND	BROWND	4	1				1					1.25		
BROY7D	BROY7D	60				f		4.92			1			
BRWNAL	BRWNAL10	10	1				1					1.14		
BRWNAL	BRWNL100	100	1							f		3.50		
BRYBND	BRYBND	10	1				1				1			
BRYBND	BRYBND18	100	1				1				1			
BRYTRI	BRYTRI10	600	1					1.20				1.20		
BRYTRI	BRYTRI2	5	1				1				1			
BRYTRI	BRYTRI6	20	1					1.20				1.20		
CHEBYQ	CHEBYQ10	10	1							f				f
CHEBYQ	CHEBYQ8	8		1.15					3.08		1			
CHNRSN	CHNRSH10	10	1					1.57			1			
CLIFF	CLIFF	2	1				1				1			
CLUSTR	CLUSTR	2		1.44					3.44		1			
CRGLVY	CRGLVY	4		1.05					1.09		1			
CRGLVY	CRGLVY10	10	1				1					1.06		
CRGLVY	CRGLY500	500	1				1					1.40		
CRGLVY	CRGL1000	1000	1				1					1.13		
DIXON	DIXON	10	1				1					1.50		
DIX7DG	DIX7DGA	15	1					3.71			1			
ENGVL1	ENGVL1A	2	1				1				1			
ENGVL1	ENGVL1B2	10	1				1				1			
ENGVL1	ENGVL1B4	100	1				1					1.12		
ENGVL1	ENGVL1B6	1000	1				1				1			
ENGVL2	ENGVL2	3		1.38					1.46		1			
EXTRSN	EXTRAR10	10	1						1.33			1.05		
EXTRSN	EXTRAR50	50	1						1.33			1.24		
EXTRSN	EXTRA100	100	1						1.33			1.10		
EXTRSN	EXTR1000	1000	1						1.33			1.19		
FRANK	FRANK12	12	1				1					1.50		
FRANK	FRANK8	8	1				1					1.50		
FRDRTH	FRDRTH	2	1				1				1			
FRDRTH	FRDRTHB3	50	1				1					1.14		
FRDRTH	FRDRTHB4	100	1				1				1			
FRDRTH	FRDRTHB7	1000	1				1				1			
GENRSN	GENT1000	1000	1					1.02					1.15	
GENRSN	GENT2B	2	1						2.88			1.12		
GENRSN	GENT50A	50	1					1.03					1.26	
GENRSN	GENT500A	500	1					1.02					1.16	
GOTTFR	GOTTFR	2	1					1.64			1			
GULF	GULFSH2	3		1.11					1.63		1			
HELIX	HELIX	3	1				1					1.08		
HILBRT	HILBRT12	12		1.50			1					1.50		
HILBRT	HILBR10A	10		1.50			1					1.50		
HIMLN3	HIMLN3	2		1.25			1					1.25		
HIMM1	HIMM1	2	1				1					1.50		
HIMM25	HIMM25	2	1				1					1.50		
HIMM27	HIMM27	2		1.15			1					1.15		
HIMM28	HIMM28	2	1					1.14			1			
HIMM29	HIMM29	2	1						1.28			1.06		
HIMM30	HIMM30	3		1.09					1.18		1			
HIMM32	HIMM32	4	1						1.50			1.25		
HIMM33	HIMM33A	2	1						2.00			1.20		
HYPCIR	HYPCIR	2	1					1.17			1			
JENSMP	JENSMP	2	1				1				1			
KOWOSB	KOWOSB1	4	1					1.90			1			
MANCIN	MANCIN10	10	1				1				1			
MANCIN	MANCIN50	50	1				1				1			
MEYER	MEYER	3	1						1.35			1.34		
MNSRF1	NMSURF64	36		8.20						f	1			
MNSRF1	NMSUR484	400	1							f		1.55		
MNSRF2	SNMSR484	400		1.15						f	1			
MNSRF2	SNMSUR64	36		2.88						f	1			
MORCIN	MORCIN10	10	1				1				1			
MOREBV	MORBV998	998	1							f		1.25		
MOREBV	MOREBV10	10	1				1				1			
MOREBV	MOREBV18	18	1				1				1			
NONDIA	NONDIA10	10	1						1.35			1.25		
NONDIA	NONDIA20	20	1						1.35			1.05		
NONDIA	NONDI500	500	1						1.35			1.05		
NONDIA	NOND1000	1000	1						1.29			1.05		
OSBRN1	OSBRN1	5		1.33			1							f

OSBRN2	OSBRN2	11	1							f		1.21		
PENAL1	PEN1GM6	10	1						1.23			1.12		
PENAL1	PEN1LN1	50	1				1				1			
PENAL1	PEN1LN2	100	1				1				1			
PENAL1	PEN1LN3	1000	1				1				1			
PENAL2	PEN2GM1	50	1							f		1.25		
PENAL2	PEN2GM2	100				f				f	1			
PENAL2	PEN2GM6	4	1						1.50			1.05		
PENAL3	PENL3GM3	50		1.07					1.14		1			
PENAL3	PENL3GM4	100		1.23					3.08		1			
PENAL3	PENL3GM5	1000		1.11					1.68		1			
POWBSC	POWBSC	2	1						1.93			1.13		
POWBSC	POWBSC50	50	1							f				f
POWBSC	POWBS500	500	1							f				f
POWER	POWER10	10	1				1				1			
POWER	POWER75	75	1				1				1			
POWQUD	POWQUD8A	4	1				1				1			
POWSSQ	POWSSQ	2				f		2.84			1			
PWSING	PWSING4	4	1				1					1.06		
PWSING	PWSING60	60	1				1					1.06		
PWSING	PWSIN100	100	1				1					1.06		
PWSING	PWSI1000	1000	1				1					1.06		
QUARTC	QUARTC	25	1							f		1.11		
RECIPE	RECIPE	3		1.20				1.20			1			
ROSENB	ROSENB	2	1						1.33			1.14		
ROSENB	SHNRSN10	10	1						1.54			1.03		
SARSEB	SARSEB	4	1				1				1			
SCHMVT	SCHMVT	3	1				1				1			
SCHMVT	SCHMVT50	50	1				1				1			
SCHMVT	SCHMV500	500	1				1					1.25		
SCHMVT	SCHV1000	1000	1				1				1			
SISSER	SISSER	2	1				1				1			
SQRTMX	MSQRTB49	49		1.23						f	1			
SQRTMX	MSQRTB9	9		1.38					3.00		1			
TDQUAD	TDQ10	10	1				1				1			
TDQUAD	TDQ1000	1000	1				1					1.50		
TDQUAD	TDQ500	500	1				1					1.50		
TOINT	PSPTOINT	50	1				1				1			
TOIN2	TOIN2	3	1				1					1.50		
TOIN4	TOIN4	4	1				1					2.00		
TRIDIA	TRIDIA10	10		1.50			1					1.50		
TRIDIA	TRLN100	100	1							f	1			
TRIDIA	TRLN1000	1000	1							f		1.50		
TRIGTO	TRIGT100	100		1.54					7.00		1			
TRIGTO	TRIGT50	50		1.17					5.50		1			
VARDIM	VARDIM	10	1				1				1			
VARDIM	VARDM100	100	1				1				1			
VAROSB	VAROSBG1	50	1				1					1.50		
VAROSB	VAROSBG2	100	1				1					1.75		
WATSON	WATSON12	12	1				1					1.46		
WATSON	WATSON6	6	1				1					1.08		
WOODS	WOODS	4	1						1.23			1.05		
WOODS	WOODS80	80	1						1.23			1.05		
XTX	XTX16	16	1				1					1.50		
XTX	XTX2	2	1				1				1			
ZANGWL	ZANGWL1	3	1				1					1.50		
Totals			112	26	0	3	74	16	34	17	66	67	3	5

Table C.13: Place finishes for number of iterations, tolerance = 10^{-3}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10	1						2.50			1.17		
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10	1					2.00			1			
BRWNAL	BRWNL100	100	1					2.00			1			
BRYBND	BRYBND	10	1				1							f
BRYBND	BRYBND18	100	1					1.25				1.25		
BRYTRI	BRYTRI6	20	1						6.50			2.00		
BRYTRI	BRYTRI10	600	1						7.00			1.50		
CHEBYQ	CHEBYQ10	10		1.33					3.00		1			
CHNRSN	CHNRSH10	10		2.25					5.00		1			
CRGLVY	CRGLVY10	10		1.75					2.50		1			
CRGLVY	CRGLY500	500	1						1.71			1.14		
CRGLVY	CRGL1000	1000	1						1.71			1.29		
DIX7DG	DIX7DGA	15		1.67					2.33		1			
DIXON	DIXON	10	1						7.00			2.00		
ENGVL1	ENGVL1B2	10		1.33					2.00		1			
ENGVL1	ENGVL1B4	100		1.33					2.00		1			
ENGVL1	ENGVL1B6	1000	1					1.50				1.50		
EXTRSN	EXTRAR50	50		1.90					3.00		1			
EXTRSN	EXTRA100	100		1.90					2.60		1			
EXTRSN	EXTR1000	1000		1.73					2.36		1			
FRANK	FRANK12	12	1						15.00			5.00		
FRDRTH	FRDRTHB3	50	1						3.25			1.25		
FRDRTH	FRDRTHB4	100	1					2.40			1			
FRDRTH	FRDRTHB7	1000	1						2.00			1.50		
GENRSN	GENT50A	50		1.14					3.36		1			
GENRSN	GENT500A	500		1.12					3.67		1			
GENRSN	GENT1000	1000		1.11					3.70		1			
HILBRT	HILBRT12	12		2.00					4.00		1			
MANCIN	MANCIN50	50	1				1					1.50		
MNSRF1	NMSURF64	36	1						3.00			1.64		
MNSRF1	NMSUR484	400			2.22			1.65			1			
MNSRF2	SNMSUR64	36	1						4.29			2.43		
MNSRF2	SNMSR484	400			2.47			1.61			1			
MORCIN	MORCIN10	10		2.00				2.00			1			
MOREBV	MOREBV18	18	1						11.22			1.22		
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20		1.88					2.00		1			
NONDIA	NONDI500	500		2.43					2.57		1			
NONDIA	NOND1000	1000		2.43				2.43			1			
OSBRN2	OSBRN2	11		1.08					4.46		1			
PENAL1	PEN1LN1	50	1					1.40					3.60	
PENAL1	PEN1LN2	100	1					1.40					9.60	
PENAL1	PEN1LN3	1000	1					1.40					300.40	
PENAL2	PEN2GM1	50	1						6.75			3.25		
PENAL2	PEN2GM2	100	1							f				f
PENAL3	PENL3GM3	50	1						2.00			1.44		
PENAL3	PENL3GM4	100	1					2.33					3.89	
PENAL3	PENL3GM5	1000	1					1.85						f
POWBSC	POWBSC50	50	1					1.40						f
POWBSC	POWBS500	500	1					1.40						f
POWBSC	POWB1000	1000	1					1.40						f
POWER	POWER75	75		1.25					2.00		1			
PWSING	PWSING60	60	1						2.00			1.20		
PWSING	PWSIN100	100	1					2.00			1			
PWSING	PWSI1000	1000	1						2.00			1.20		
QUARTC	QUARTC	25	1						1.40			1.20		
ROSENB	SHNRSN10	10	1					1.25				1.25		
SCHMVT	SCHMVT50	50	1						4.50			1.50		
SCHMVT	SCHMV500	500	1					4.50			1			
SCHMVT	SCHV1000	1000	1					5.00			1			
SQRTMX	MSQRTB49	49	1						2.80			1.40		
TDQUAD	TDQ10	10	1					4.00			1			
TDQUAD	TDQ500	500	1					2.00					7.00	
TDQUAD	TDQ1000	1000	1					2.00					10.00	
TOINT	PSPTOINT	50		1.25					2.62		1			
TRIDIA	TRIDIA10	10	1						3.50			1.50		
TRIDIA	TRLN100	100	1						6.33			1.33		
TRIDIA	TRLN1000	1000	1						7.00			2.67		
TRIGTO	TRIGT50	50	1						2.12			1.25		
TRIGTO	TRIGT100	100	1						2.50			1.17		
VARDIM	VARDIM	10		2.50				2.50			1			
VARDIM	VARDM100	100		2.50					3.50		1			
VAROSB	VAROSBG1	50	1						10.40			2.00		
VAROSB	VAROSBG2	100	1						11.00			1.78		
WATSON	WATSON12	12	1					3.20			1			
WOODS	WOODS80	80	1						3.33			1.67		
XTX	XTX16	16	1					2.00			1			
Totals			52	22	2	2	2	27	46	3	33	31	6	8

Table C.14: Place finishes for number of iterations, tolerance = 10^{-6}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10	1						2.56			1.11		
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10	1						3.00			2.00		
BRWNAL	BRWNL100	100	1					2.00			1			
BRYBND	BRYBND	10	1					1.17						f
BRYBND	BRYBND18	100	1					1.50						f
BRYTRI	BRYTRI6	20	1						5.75			1.50		
BRYTRI	BRYTRI10	600	1						6.25			2.00		
CHEBYQ	CHEBYQ10	10		1.25					3.38		1			
CHNRSN	CHNRSH10	10		1.62					3.25		1			
CRGLVY	CRGLVY10	10		1.09					1.91		1			
CRGLVY	CRGLY500	500	1					2.30					2.40	
CRGLVY	CRGL1000	1000	1					1.90					3.00	
DIX7DG	DIX7DGA	15		1.75					2.25		1			
DIXON	DIXON	10	1						5.00			1.75		
ENGVL1	ENGVL1B2	10		1.17					1.83		1			
ENGVL1	ENGVL1B4	100		1.17					1.83		1			
ENGVL1	ENGVL1B6	1000	1					1.67				1.67		
EXTRSN	EXTRAR50	50		2.00					2.77		1			
EXTRSN	EXTRA100	100		2.00					2.46		1			
EXTRSN	EXTR1000	1000		1.86					2.36		1			
FRANK	FRANK12	12	1						32.00			7.00		
FRDRTH	FRDRTHB3	50		1.67					2.50		1			
FRDRTH	FRDRTHB4	100		1.29					2.14		1			
FRDRTH	FRDRTHB7	1000	1						1.86			1.43		
GENRSN	GENT50A	50		1.15					3.47		1			
GENRSN	GENT500A	500		1.13					3.75		1			
GENRSN	GENT1000	1000		1.13					3.75		1			
HILBRT	HILBRT12	12		1.33					3.00		1			
MANCIN	MANCIN50	50	1				1					1.67		
MNSRF1	NMSURF64	36	1						4.93			1.60		
MNSRF1	NMSUR484	400		2.27					3.84		1			
MNSRF2	SNMSUR64	36	1						7.10			2.10		
MNSRF2	SNMSR484	400		2.71					3.87		1			
MORCIN	MORCIN10	10	1					1.50			1			
MOREBV	MOREBV18	18	1						13.50			1.50		
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20			2.50			2.42			1			
NONDIA	NONDI500	500			2.91			2.73			1			
NONDIA	NOND1000	1000		2.50				2.50			1			
OSBRN2	OSBRN2	11		1.22					8.56		1			
PENAL1	PEN1LN1	50	1					1.44					2.33	
PENAL1	PEN1LN2	100	1					1.44					6.33	
PENAL1	PEN1LN3	1000	1					1.44					196.56	
PENAL2	PEN2GM1	50	1						6.00			2.33		
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50	1						2.27			1.27		
PENAL3	PENL3GM4	100	1					2.42						f
PENAL3	PENL3GM5	1000	1					1.87						f
POWBSC	POWBSC50	50	1					1.44						f
POWBSC	POWBS500	500	1					1.56						f
POWBSC	POWB1000	1000	1					1.56						f
POWER	POWER75	75		2.00					3.00		1			
PWSING	PWSING60	60	1					1.89			1			
PWSING	PWSIN100	100	1					1.78			1			
PWSING	PWSI1000	1000		1.12					2.12		1			
QUARTC	QUARTC	25		1.29					1.86		1			
ROSENB	SHNRSN10	10		5.86					7.86		1			
SCHMVT	SCHMVT50	50	1						5.25			1.75		
SCHMVT	SCHMV500	500	1						5.25			1.50		
SCHMVT	SCHV1000	1000	1						5.25			1.50		
SQRTMX	MSQRTB49	49	1						6.40			1.30		
TDQUAD	TDQ10	10	1					4.00			1			
TDQUAD	TDQ500	500	1					2.33				2.33		
TDQUAD	TDQ1000	1000	1					2.00					3.33	
TOINT	PSPTOINT	50	1					3.67			1			
TRIDIA	TRIDIA10	10	1					3.00			1			
TRIDIA	TRLN100	100	1						9.20			1.60		
TRIDIA	TRLN1000	1000	1						10.50			1.30		
TRIGTO	TRIGT50	50	1						3.33			1.22		
TRIGTO	TRIGT100	100	1						3.93			1.07		
VARDIM	VARDIM	10		3.00					4.00		1			
VARDIM	VARDM100	100		3.00					4.33		1			
VAROSB	VAROSBG1	50	1						8.90			1.40		
VAROSB	VAROSBG2	100		1.55					7.82		1			
WATSON	WATSON12	12	1						10.31			1.08		
WOODS	WOODS80	80		1.89					3.00		1			
XTX	XTX16	16	1					2.00			1			
Totals			46	27	2	3	1	26	48	3	37	25	6	10

Table C.15: Place finishes for number of iterations, tolerance = 10^{-9}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10	1						2.90			1.10		
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10		1.33					4.00		1			
BRWNAL	BRWNL100	100	1					2.00			1			
BRYBND	BRYBND	10	1					1.29						f
BRYBND	BRYBND18	100	1					1.57						f
BRYTRI	BRYTRI6	20	1						5.50			1.67		
BRYTRI	BRYTRI10	600	1						7.00			2.60		
CHEBYQ	CHEBYQ10	10		1.20					3.00		1			
CHNRSN	CHNRSH10	10		1.56					3.33		1			
CRGLVY	CRGLVY10	10	1					2.94			1			
CRGLVY	CRGLY500	500	1						2.73			2.20		
CRGLVY	CRGL1000	1000	1						2.79			2.71		
DIX7DG	DIX7DGA	15		2.00					2.50		1			
DIXON	DIXON	10	1						4.83			1.33		
ENGVL1	ENGVL1B2	10		1.14					1.86		1			
ENGVL1	ENGVL1B4	100	1					1.62			1			
ENGVL1	ENGVL1B6	1000	1					1.62				1.62		
EXTRSN	EXTRAR50	50		2.00					2.64		1			
EXTRSN	EXTRA100	100		2.00					2.43		1			
EXTRSN	EXTR1000	1000		1.87					2.33		1			
FRANK	FRANK12	12	1						44.00			10.00		
FRDRTH	FRDRTHB3	50		1.25					2.12		1			
FRDRTH	FRDRTHB4	100	1					1.89			1			
FRDRTH	FRDRTHB7	1000	1					1.36				1.36		
GENRSN	GENT50A	50		1.16					3.49		1			
GENRSN	GENT500A	500		1.14					3.75		1			
GENRSN	GENT1000	1000		1.13					3.75		1			
HILBRT	HILBRT12	12		1.25					7.00		1			
MANCIN	MANCIN50	50	1				1					1.75		
MNSRF1	NMSURF64	36	1						6.44			1.44		
MNSRF1	NMSUR484	400		2.43					5.49		1			
MNSRF2	SNMSUR64	36	1						9.50			2.08		
MNSRF2	SNMSR484	400		2.80					5.42		1			
MORCIN	MORCIN10	10	1					1.33			1			
MOREBV	MOREBV18	18	1						12.96			1.25		
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20			2.36			2.21			1			
NONDIA	NONDI500	500			3.17			2.67			1			
NONDIA	NOND1000	1000			2.92			2.54			1			
OSBRN2	OSBRN2	11		1.18					10.18		1			
PENAL1	PEN1LN1	50	1					1.46					1.62	
PENAL1	PEN1LN2	100	1					1.46					4.46	
PENAL1	PEN1LN3	1000	1					1.46					139.77	
PENAL2	PEN2GM1	50	1						5.88			2.00		
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50	1						2.62			1.23		
PENAL3	PENL3GM4	100	1					2.92						f
PENAL3	PENL3GM5	1000	1					2.08						f
POWBSC	POWBSC50	50	1					8.40						f
POWBSC	POWBS500	500	1					11.65						f
POWBSC	POWB1000	1000	1					6.33						f
POWER	POWER75	75		2.50					4.00		1			
PWSING	PWSING60	60		1.27					2.27		1			
PWSING	PWSIN100	100		1.40					3.60		1			
PWSING	PWSI1000	1000		1.27					2.91		1			
QUARTC	QUARTC	25		1.62					2.50		1			
ROSENB	SHNRSN10	10			24.25			17.42			1			
SCHMVT	SCHMVT50	50	1						6.00			1.67		
SCHMVT	SCHMV500	500	1						5.50			1.67		
SCHMVT	SCHV1000	1000	1						6.60			2.00		
SQRTMX	MSQRTB49	49		1.18					8.00		1			
TDQUAD	TDQ10	10		1.50					5.00		1			
TDQUAD	TDQ500	500	1						3.00			2.33		
TDQUAD	TDQ1000	1000	1					2.67					3.33	
TOINT	PSPTOINT	50	1						5.23			1.15		
TRIDIA	TRIDIA10	10	1						3.40			1.20		
TRIDIA	TRLN100	100	1						10.14			1.57		
TRIDIA	TRLN1000	1000		1.11					9.89		1			
TRIGTO	TRIGT50	50	1						3.91			1.09		
TRIGTO	TRIGT100	100	1					5.06			1			
VARDIM	VARDIM	10		2.40					3.20		1			
VARDIM	VARDM100	100		3.25					4.75		1			
VAROSB	VAROSBG1	50	1						7.80			1.20		
VAROSB	VAROSBG2	100		2.27					10.00		1			
WATSON	WATSON12	12	1						425.79			75.58		
WOODS	WOODS80	80		1.97					2.97		1			
XTX	XTX16	16	1					2.00			1			
Totals			43	28	4	3	1	24	50	3	39	25	4	10

Table C.16: Place finishes for number of iterations, tolerance = 10^{-12}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10	1						3.08			1.17		
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10		1.75					3.50		1			
BRWNAL	BRWNL100	100	1					2.00			1			
BRYBND	BRYBND	10	1					1.38						f
BRYBND	BRYBND18	100	1					1.44						f
BRYTRI	BRYTRI6	20	1						6.00			1.71		
BRYTRI	BRYTRI10	600	1						6.57			2.57		
CHEBYQ	CHEBYQ10	10	1						2.77			1.31		
CHNRSN	CHNRSH10	10		1.45					3.64		1			
CRGLVY	CRGLVY10	10	1						3.74			1.05		
CRGLVY	CRGLY500	500	1						4.43			1.76		
CRGLVY	CRGL1000	1000	1						3.81			2.19		
DIX7DG	DIX7DGA	15		2.50					3.00		1			
DIXON	DIXON	10	1						4.50			1.12		
ENGVL1	ENGVL1B2	10	1					1.67			1			
ENGVL1	ENGVL1B4	100	1					1.50			1			
ENGVL1	ENGVL1B6	1000	1					1.67					1.89	
EXTRSN	EXTRAR50	50		2.00					2.53		1			
EXTRSN	EXTRA100	100		2.00					2.33		1			
EXTRSN	EXTR1000	1000		1.88					2.25		1			
FRANK	FRANK12	12	1						52.00			13.00		
FRDRTH	FRDRTHB3	50		1.30					1.90		1			
FRDRTH	FRDRTHB4	100	1					1.58			1			
FRDRTH	FRDRTHB7	1000	1					1.55					1.82	
GENRSN	GENT50A	50		1.16					3.49		1			
GENRSN	GENT500A	500		1.13					3.75		1			
GENRSN	GENT1000	1000		1.13					3.75		1			
HILBRT	HILBRT12	12	1						15.17			1.17		
MANCIN	MANCIN50	50		1.20			1						1.80	
MNSRF1	NMSURF64	36	1						7.75			1.45		
MNSRF1	NMSUR484	400		2.55					6.85		1			
MNSRF2	SNMSUR64	36	1						9.27			1.80		
MNSRF2	SNMSR484	400		2.84					6.66		1			
MORCIN	MORCIN10	10	1					1.50			1			
MOREBV	MOREBV18	18	1						13.84			1.32		
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20			2.50			2.29			1			
NONDIA	NONDI500	500			3.15			2.54			1			
NONDIA	NOND1000	1000			2.93			2.36			1			
OSBRN2	OSBRN2	11		1.12					10.54		1			
PENAL1	PEN1LN1	50	1						1.50			1.44		
PENAL1	PEN1LN2	100	1					1.47					3.47	
PENAL1	PEN1LN3	1000	1					1.44					101.39	
PENAL2	PEN2GM1	50	1						5.70			1.80		
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50	1						2.73			1.20		
PENAL3	PENL3GM4	100	1					3.14						f
PENAL3	PENL3GM5	1000	1					2.32						f
POWBSC	POWBSC50	50	1							f				f
POWBSC	POWBS500	500	1							f				f
POWBSC	POWB1000	1000	1							f				f
POWER	POWER75	75		2.38					4.12		1			
PWSING	PWSING60	60		1.29					3.57		1			
PWSING	PWSIN100	100		1.50					4.17		1			
PWSING	PWSI1000	1000		1.38					3.69		1			
QUARTC	QUARTC	25		2.00					2.89		1			
ROSENB	SHNRSN10	10			9.06			5.49			1			
SCHMVT	SCHMVT50	50	1						6.00			1.50		
SCHMVT	SCHMV500	500	1						6.86			2.00		
SCHMVT	SCHV1000	1000	1						5.86			2.00		
SQRTMX	MSQRTB49	49		1.43					8.29		1			
TDQUAD	TDQ10	10	1					4.00			1			
TDQUAD	TDQ500	500	1						2.50			1.75		
TDQUAD	TDQ1000	1000	1					2.50				2.50		
TOINT	PSPTOINT	50	1						6.33			1.27		
TRIDIA	TRIDIA10	10	1						2.86			1.14		
TRIDIA	TRLN100	100	1						9.20			1.30		
TRIDIA	TRLN1000	1000		1.41					11.45		1			
TRIGTO	TRIGT50	50	1						4.08			1.08		
TRIGTO	TRIGT100	100		1.06					5.61		1			
VARDIM	VARDIM	10		2.17					2.83		1			
VARDIM	VARDM100	100		3.60					5.00		1			
VAROSB	VAROSBG1	50	1						8.74			1.37		
VAROSB	VAROSBG2	100		2.71					10.48		1			
WATSON	WATSON12	12	1							f				f
WOODS	WOODS80	80		1.94					2.87		1			
XTX	XTX16	16	1					2.00			1			
Totals			45	26	4	3	1	20	50	7	36	26	5	11

Table C.17: Place finishes for number of function evaluations, tolerance = 10^{-3}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10	1					2.75					4.75	
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10	1					1.50					2.00	
BRWNAL	BRWNL100	100	1					1.50				1.50		
BRYBND	BRYBND	10	1				1							f
BRYBND	BRYBND18	100	1					1.20					7.60	
BRYTRI	BRYTRI6	20	1					4.00					6.50	
BRYTRI	BRYTRI10	600	1					4.00					6.00	
CHEBYQ	CHEBYQ10	10	1					1.77					2.69	
CHNRSN	CHNRSH10	10	1					1.26				1.26		
CRGLVY	CRGLVY10	10	1					1.38					2.75	
CRGLVY	CRGLY500	500	1					1.75					6.12	
CRGLVY	CRGL1000	1000	1					1.75					7.62	
DIX7DG	DIX7DGA	15	1					1.33					2.33	
DIXON	DIXON	10	1					5.00					6.33	
ENGVL1	ENGVL1B2	10	1					1.40					4.00	
ENGVL1	ENGVL1B4	100	1					1.40					2.20	
ENGVL1	ENGVL1B6	1000	1					1.60					5.00	
EXTRSN	EXTRAR50	50	1					1.18					1.55	
EXTRSN	EXTRA100	100	1					1.09					1.88	
EXTRSN	EXTR1000	1000	1					1.09					1.61	
FRANK	FRANK12	12	1					8.00					13.50	
FRDRTH	FRDRTHB3	50	1					1.67					2.22	
FRDRTH	FRDRTHB4	100	1					1.44					2.22	
FRDRTH	FRDRTHB7	1000	1					2.00					3.67	
GENRSN	GENT50A	50	1					2.64					4.97	
GENRSN	GENT500A	500	1					2.88					5.27	
GENRSN	GENT1000	1000	1					2.98					5.53	
HILBRT	HILBRT12	12	1						1.67			1.33		
MANCIN	MANCIN50	50	1					1.33					5.33	
MNSRF1	NMSURF64	36		1.30			1						2.51	
MNSRF1	NMSUR484	400		4.05			1						4.91	
MNSRF2	SNMSUR64	36	1					1.10					3.03	
MNSRF2	SNMSR484	400		4.63			1						4.71	
MORCIN	MORCIN10	10	1					1.33				1.33		
MOREBV	MOREBV18	18	1						10.80			10.00		
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20		1.48			1						2.22	
NONDIA	NONDI500	500			1.72		1					1.24		
NONDIA	NOND1000	1000			1.96		1					1.30		
OSBRN2	OSBRN2	11	1					4.06					4.88	
PENAL1	PEN1LN1	50	1					1.67					7.83	
PENAL1	PEN1LN2	100	1					1.67					17.00	
PENAL1	PEN1LN3	1000	1					2.17					501.17	
PENAL2	PEN2GM1	50	1					5.83					7.50	
PENAL2	PEN2GM2	100	1							f				f
PENAL3	PENL3GM3	50	1					2.13					2.93	
PENAL3	PENL3GM4	100	1					1.94					6.35	
PENAL3	PENL3GM5	1000	1					1.68						f
POWBSC	POWBSC50	50	1					1.33						f
POWBSC	POWBS500	500	1					1.33						f
POWBSC	POWB1000	1000	1					1.50						f
POWER	POWER75	75	1					1.50					5.00	
PWSING	PWSING60	60	1					2.00					4.83	
PWSING	PWSIN100	100	1					2.00					4.33	
PWSING	PWSI1000	1000	1					1.83					5.17	
QUARTC	QUARTC	25	1					1.50					5.17	
ROSENB	SHNRSN10	10	1					1.20					5.80	
SCHMVT	SCHMVT50	50	1					4.00					7.33	
SCHMVT	SCHMV500	500	1						4.33			3.00		
SCHMVT	SCHV1000	1000	1						4.67			3.00		
SQRTMX	MSQRTB49	49	1					1.90					4.80	
TDQUAD	TDQ10	10	1						2.50			2.00		
TDQUAD	TDQ500	500	1					2.50					10.00	
TDQUAD	TDQ1000	1000	1					2.50					13.00	
TOINT	PSPTOINT	50	1					1.06					1.53	
TRIDIA	TRIDIA10	10	1					2.67					4.00	
TRIDIA	TRLN100	100	1						5.25			3.25		
TRIDIA	TRLN1000	1000	1					6.00					16.50	
TRIGTO	TRIGT50	50	1					1.62					6.54	
TRIGTO	TRIGT100	100	1					1.80					3.85	
VARDIM	VARDIM	10	1				1					1.17		
VARDIM	VARDM100	100	1					1.33					1.67	
VAROSB	VAROSBG1	50	1					8.83					14.00	
VAROSB	VAROSBG2	100	1					10.10					15.90	
WATSON	WATSON12	12	1					3.33					4.00	
WOODS	WOODS80	80	1					2.75					5.50	
XTX	XTX16	16	1					1.50				1.50		
Totals			70	4	2	2	8	61	6	3	0	13	57	8

Table C.18: Place finishes for number of function evaluations, tolerance = 10^{-6}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10	1					2.82					4.36	
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10	1					2.00					5.00	
BRWNAL	BRWNL100	100	1					1.50				1.50		
BRYBND	BRYBND	10	1					1.14						f
BRYBND	BRYBND18	100	1					1.43						f
BRYTRI	BRYTRI6	20	1					4.50					6.00	
BRYTRI	BRYTRI10	600	1					4.67					13.67	
CHEBYQ	CHEBYQ10	10	1					2.20					3.00	
CHNRSN	CHNRSH10	10	1					1.35					1.74	
CRGLVY	CRGLVY10	10	1					1.69					5.15	
CRGLVY	CRGLY500	500	1					2.36					19.73	
CRGLVY	CRGL1000	1000	1					2.09					21.82	
DIX7DG	DIX7DGA	15	1					1.25					2.25	
DIXON	DIXON	10	1					4.20					6.60	
ENGVL1	ENGVL1B2	10	1					1.50					3.75	
ENGVL1	ENGVL1B4	100	1					1.50					3.00	
ENGVL1	ENGVL1B6	1000	1					1.71					6.57	
EXTRSN	EXTRAR50	50		1.13			1						1.41	
EXTRSN	EXTRA100	100		1.21			1						1.70	
EXTRSN	EXTR1000	1000		1.18			1						1.52	
FRANK	FRANK12	12	1					17.50					19.50	
FRDRTH	FRDRTHB3	50	1				1					1.41		
FRDRTH	FRDRTHB4	100	1					1.31					2.08	
FRDRTH	FRDRTHB7	1000	1					1.55					4.27	
GENRSN	GENT50A	50	1					2.73					4.96	
GENRSN	GENT500A	500	1					2.94					5.27	
GENRSN	GENT1000	1000	1					2.96					5.39	
HILBRT	HILBRT12	12	1						2.20			2.00		
MANCIN	MANCIN50	50	1					1.25					5.75	
MNSRF1	NMSURF64	36	1					1.46					2.59	
MNSRF1	NMSUR484	400		1.64			1						2.43	
MNSRF2	SNMSUR64	36	1					2.27					3.70	
MNSRF2	SNMSR484	400		1.91			1						2.35	
MORCIN	MORCIN10	10	1					1.67					2.33	
MOREBV	MOREBV18	18	1						13.35			12.29		
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20		1.70			1						1.80	
NONDIA	NONDI500	500			2.34		1					1.25		
NONDIA	NOND1000	1000			2.32		1					1.32		
OSBRN2	OSBRN2	11	1						5.73			3.60		
PENAL1	PEN1LN1	50	1					1.60					5.70	
PENAL1	PEN1LN2	100	1					1.60					12.10	
PENAL1	PEN1LN3	1000	1					1.90					354.10	
PENAL2	PEN2GM1	50	1					5.62					6.12	
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50	1					2.29					2.82	
PENAL3	PENL3GM4	100	1					2.10						f
PENAL3	PENL3GM5	1000	1					1.72						f
POWBSC	POWBSC50	50	1					1.40						f
POWBSC	POWBS500	500	1					1.50						f
POWBSC	POWB1000	1000	1					1.80						f
POWER	POWER75	75	1					1.45					3.18	
PWSING	PWSING60	60	1					1.90					4.50	
PWSING	PWSIN100	100	1					1.90					4.60	
PWSING	PWSI1000	1000	1					1.80					4.10	
QUARTC	QUARTC	25	1					1.50					3.60	
ROSENB	SHNRSN10	10		1.53					2.00		1			
SCHMVT	SCHMVT50	50	1					5.00					7.60	
SCHMVT	SCHMV500	500	1					5.00					9.00	
SCHMVT	SCHV1000	1000	1					5.20					9.20	
SQRTMX	MSQRTB49	49	1					4.73					7.13	
TDQUAD	TDQ10	10	1						4.00			2.33		
TDQUAD	TDQ500	500	1					3.25					5.00	
TDQUAD	TDQ1000	1000	1					2.75					6.50	
TOINT	PSPTOINT	50	1					1.69					2.11	
TRIDIA	TRIDIA10	10	1					2.60					3.00	
TRIDIA	TRLN100	100	1						8.00			6.67		
TRIDIA	TRLN1000	1000	1					10.27					10.64	
TRIGTO	TRIGT50	50	1					2.50					6.29	
TRIGTO	TRIGT100	100	1					2.77					3.68	
VARDIM	VARDIM	10	1						1.30			1.10		
VARDIM	VARDM100	100	1					1.40				1.40		
VAROSB	VAROSBG1	50	1					8.36					10.09	
VAROSB	VAROSBG2	100	1					5.09					6.34	
WATSON	WATSON12	12	1						8.33			3.78		
WOODS	WOODS80	80	1					1.06					1.71	
XTX	XTX16	16	1					1.50				1.50		
Totals			66	7	2	3	9	58	8	3	1	13	54	10

Table C.19: Place finishes for number of function evaluations, tolerance = 10^{-9}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10	1					3.17					4.42	
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10	1						1.75			1.62		
BRWNAL	BRWNL100	100	1					1.50				1.50		
BRYBND	BRYBND	10	1					1.25						f
BRYBND	BRYBND18	100	1					1.50						f
BRYTRI	BRYTRI6	20	1					4.62					6.62	
BRYTRI	BRYTRI10	600	1					5.57					19.86	
CHEBYQ	CHEBYQ10	10	1					2.18					3.12	
CHNRSN	CHNRSH10	10	1					1.46					1.83	
CRGLVY	CRGLVY10	10	1					2.94					5.47	
CRGLVY	CRGLY500	500	1					2.94					22.12	
CRGLVY	CRGL1000	1000	1					2.93					26.80	
DIX7DG	DIX7DGA	15	1					1.22					2.00	
DIXON	DIXON	10	1					4.43					5.14	
ENGVL1	ENGVL1B2	10	1					1.56					3.67	
ENGVL1	ENGVL1B4	100	1					1.56					3.56	
ENGVL1	ENGVL1B6	1000	1					1.67					6.89	
EXTRSN	EXTRAR50	50		1.19			1						1.45	
EXTRSN	EXTRA100	100		1.24			1						1.69	
EXTRSN	EXTR1000	1000		1.22			1						1.52	
FRANK	FRANK12	12	1					24.00					25.50	
FRDRTH	FRDRTHB3	50	1					1.12					1.76	
FRDRTH	FRDRTHB4	100	1					1.46					2.62	
FRDRTH	FRDRTHB7	1000	1					1.27					5.00	
GENRSN	GENT50A	50	1					2.74					4.90	
GENRSN	GENT500A	500	1					2.93					5.26	
GENRSN	GENT1000	1000	1					2.96					5.38	
HILBRT	HILBRT12	12	1						5.50			2.67		
MANCIN	MANCIN50	50	1					1.20					5.80	
MNSRF1	NMSURF64	36	1					2.14					2.67	
MNSRF1	NMSUR484	400		1.12			1						1.81	
MNSRF2	SNMSUR64	36	1					3.46					4.26	
MNSRF2	SNMSR484	400		1.30			1						1.76	
MORCIN	MORCIN10	10	1					1.50					2.50	
MOREBV	MOREBV18	18	1						13.04			10.48		
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20		1.81			1						1.88	
NONDIA	NONDI500	500			2.87		1					1.26		
NONDIA	NOND1000	1000			3.06		1					1.30		
OSBRN2	OSBRN2	11	1						7.14			3.63		
PENAL1	PEN1LN1	50	1					1.57					4.07	
PENAL1	PEN1LN2	100	1					1.57					8.86	
PENAL1	PEN1LN3	1000	1					1.79					259.79	
PENAL2	PEN2GM1	50	1						5.60			5.50		
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50	1					2.58					2.95	
PENAL3	PENL3GM4	100	1					2.57						f
PENAL3	PENL3GM5	1000	1					1.84						f
POWBSC	POWBSC50	50	1					4.18						f
POWBSC	POWBS500	500	1					11.20						f
POWBSC	POWB1000	1000	1					6.41						f
POWER	POWER75	75	1					1.56					2.56	
PWSING	PWSING60	60	1					1.80					3.67	
PWSING	PWSIN100	100	1					2.73					3.27	
PWSING	PWSI1000	1000	1					2.40					3.53	
QUARTC	QUARTC	25	1					1.57					2.93	
ROSENB	SHNRSN10	10			10.34			3.71			1			
SCHMVT	SCHMVT50	50	1					5.86					7.14	
SCHMVT	SCHMV500	500	1					5.43					11.43	
SCHMVT	SCHV1000	1000	1					6.50					13.50	
SQRTMX	MSQRTB49	49	1					5.88					6.60	
TDQUAD	TDQ10	10	1						3.50			1.75		
TDQUAD	TDQ500	500	1					3.75					5.00	
TDQUAD	TDQ1000	1000	1					3.50					6.50	
TOINT	PSPTOINT	50	1					2.41					2.62	
TRIDIA	TRIDIA10	10	1					3.00					3.50	
TRIDIA	TRLN100	100	1						9.25			6.75		
TRIDIA	TRLN1000	1000	1						8.90			6.76		
TRIGTO	TRIGT50	50	1					3.12					5.69	
TRIGTO	TRIGT100	100	1						3.62			3.54		
VARDIM	VARDIM	10	1					1.31					1.38	
VARDIM	VARDM100	100	1						1.43			1.29		
VAROSB	VAROSBG1	50	1					7.56					10.06	
VAROSB	VAROSBG2	100	1						4.53			4.42		
WATSON	WATSON12	12	1						263.63			232.71		
WOODS	WOODS80	80	1					1.01					1.63	
XTX	XTX16	16	1					1.50				1.50		
Totals			66	6	3	3	8	55	12	3	1	16	51	10

Table C.20: Place finishes for number of function evaluations, tolerance = 10^{-12}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10	1					3.29					4.64	
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10		1.12			1				1			
BRWNAL	BRWNL100	100	1					1.50				1.50		
BRYBND	BRYBND	10	1					1.33						f
BRYBND	BRYBND18	100	1					1.40						f
BRYTRI	BRYTRI6	20	1					5.11					6.78	
BRYTRI	BRYTRI10	600	1					5.67					21.56	
CHEBYQ	CHEBYQ10	10	1					2.44					4.94	
CHNRSN	CHNRSH10	10	1					1.77					2.04	
CRGLVY	CRGLVY10	10	1					3.70					5.35	
CRGLVY	CRGLY500	500	1					4.68					17.50	
CRGLVY	CRGL1000	1000	1					3.91					24.59	
DIX7DG	DIX7DGA	15	1					1.18					1.64	
DIXON	DIXON	10	1					4.22					4.44	
ENGVL1	ENGVL1B2	10	1					1.60					3.90	
ENGVL1	ENGVL1B4	100	1					1.45					3.55	
ENGVL1	ENGVL1B6	1000	1					1.70					8.20	
EXTRSN	EXTRAR50	50		1.21			1						1.48	
EXTRSN	EXTRA100	100		1.26			1						1.72	
EXTRSN	EXTR1000	1000		1.23			1						1.57	
FRANK	FRANK12	12	1					28.00					32.50	
FRDRTH	FRDRTHB3	50		1.05			1						1.76	
FRDRTH	FRDRTHB4	100	1					1.24					2.82	
FRDRTH	FRDRTHB7	1000	1					1.40					6.33	
GENRSN	GENT50A	50	1					2.78					4.92	
GENRSN	GENT500A	500	1					2.93					5.26	
GENRSN	GENT1000	1000	1					2.96					5.38	
HILBRT	HILBRT12	12	1						14.71			4.57		
MANCIN	MANCIN50	50	1				1					5.00		
MNSRF1	NMSURF64	36	1					2.75					2.86	
MNSRF1	NMSUR484	400	1					1.14					1.75	
MNSRF2	SNMSUR64	36	1					3.89					4.16	
MNSRF2	SNMSR484	400		1.00			1						1.48	
MORCIN	MORCIN10	10	1					1.60					2.60	
MOREBV	MOREBV18	18	1						14.06			10.97		
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20			1.88		1					1.84		
NONDIA	NONDI500	500			3.02		1					1.30		
NONDIA	NOND1000	1000			3.34		1					1.36		
OSBRN2	OSBRN2	11	1						7.82			3.77		
PENAL1	PEN1LN1	50	1					1.59					3.65	
PENAL1	PEN1LN2	100	1					1.56					7.11	
PENAL1	PEN1LN3	1000	1					1.68					192.32	
PENAL2	PEN2GM1	50	1						5.58			5.08		
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50	1					2.71					3.05	
PENAL3	PENL3GM4	100	1					2.73						f
PENAL3	PENL3GM5	1000	1					1.98						f
POWBSC	POWBSC50	50	1							f				f
POWBSC	POWBS500	500	1							f				f
POWBSC	POWB1000	1000	1							f				f
POWER	POWER75	75	1					1.70					2.65	
PWSING	PWSING60	60	1					2.84					3.63	
PWSING	PWSIN100	100	1					3.05				3.05		
PWSING	PWSI1000	1000	1					2.79					3.26	
QUARTC	QUARTC	25	1					1.47					2.42	
ROSENB	SHNRSN10	10			3.89			1.02			1			
SCHMVT	SCHMVT50	50	1					6.11					6.44	
SCHMVT	SCHMV500	500	1					6.75					14.25	
SCHMVT	SCHV1000	1000	1					5.88					14.62	
SQRTMX	MSQRTB49	49	1					5.34					5.94	
TDQUAD	TDQ10	10	1						4.00			2.50		
TDQUAD	TDQ500	500	1					3.20					4.00	
TDQUAD	TDQ1000	1000	1					3.20					5.20	
TOINT	PSPTOINT	50	1					3.03					3.10	
TRIDIA	TRIDIA10	10	1					2.62					3.62	
TRIDIA	TRLN100	100	1						8.73			5.82		
TRIDIA	TRLN1000	1000	1						8.31			5.06		
TRIGTO	TRIGT50	50	1					3.39					5.39	
TRIGTO	TRIGT100	100	1						4.00			3.37		
VARDIM	VARDIM	10	1					1.29					1.43	
VARDIM	VARDM100	100	1						1.37			1.16		
VAROSB	VAROSBG1	50	1					8.65					11.60	
VAROSB	VAROSBG2	100	1					4.00					4.26	
WATSON	WATSON12	12	1							f				f
WOODS	WOODS80	80	1					1.01					1.66	
XTX	XTX16	16	1					1.50				1.50		
Totals			65	6	4	3	10	52	9	7	2	16	49	11

Table C.21: Place finishes for number of gradient and Hessian-vector product evaluations, tolerance = 10^{-3}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10			1.86		1					1.73		
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10	1				1					1.33		
BRWNAL	BRWNL100	100	1				1				1			
BRYBND	BRYBND	10		2.00			1							f
BRYBND	BRYBND18	100		1.50			1						6.33	
BRYTRI	BRYTRI6	20	1				1					1.62		
BRYTRI	BRYTRI10	600	1				1					1.50		
CHEBYQ	CHEBYQ10	10			1.91		1					1.52		
CHNRSN	CHNRSH10	10		2.38			1				1			
CRGLVY	CRGLVY10	10			2.55		1					2.00		
CRGLVY	CRGLY500	500		1.43			1						3.50	
CRGLVY	CRGL1000	1000		1.43			1						4.36	
DIX7DG	DIX7DGA	15		1.75			1					1.75		
DIXON	DIXON	10		1.07			1						1.27	
ENGVL1	ENGVL1B2	10		1.43			1						2.86	
ENGVL1	ENGVL1B4	100		1.29			1						1.57	
ENGVL1	ENGVL1B6	1000		1.12			1						3.12	
EXTRSN	EXTRAR50	50			1.62		1					1.31		
EXTRSN	EXTRA100	100			1.75		1					1.72		
EXTRSN	EXTR1000	1000			1.75		1					1.47		
FRANK	FRANK12	12	1					1.14					1.93	
FRDRTH	FRDRTHB3	50	1				1					1.33		
FRDRTH	FRDRTHB4	100		1.46			1						1.54	
FRDRTH	FRDRTHB7	1000	1					1.20					2.20	
GENRSN	GENT50A	50			2.76		1					1.89		
GENRSN	GENT500A	500			3.10		1					1.83		
GENRSN	GENT1000	1000			3.09		1					1.85		
HILBRT	HILBRT12	12			1.50			1.25			1			
MANCIN	MANCIN50	50		1.25			1						4.00	
MNSRF1	NMSURF64	36			3.62		1					2.51		
MNSRF1	NMSUR484	400			21.50		1					4.91		
MNSRF2	SNMSUR64	36			3.73		1					2.76		
MNSRF2	SNMSR484	400			24.66		1					4.71		
MORCIN	MORCIN10	10		1.50			1				1			
MOREBV	MOREBV18	18			1.49			1.08			1			
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20			2.43		1					2.22		
NONDIA	NONDI500	500			2.55		1					1.24		
NONDIA	NOND1000	1000			2.89		1					1.30		
OSBRN2	OSBRN2	11			1.62		1					1.20		
PENAL1	PEN1LN1	50		1.10			1						4.70	
PENAL1	PEN1LN2	100		1.10			1						10.20	
PENAL1	PEN1LN3	1000	1					1.18					273.36	
PENAL2	PEN2GM1	50	1				1					1.29		
PENAL2	PEN2GM2	100	1							f				f
PENAL3	PENL3GM3	50		1.31			1						1.38	
PENAL3	PENL3GM4	100		1.33			1						3.27	
PENAL3	PENL3GM5	1000		1.52			1							f
POWBSC	POWBSC50	50		1.38			1							f
POWBSC	POWBS500	500		1.38			1							f
POWBSC	POWB1000	1000		1.22			1							f
POWER	POWER75	75		1.56			1						3.33	
PWSING	PWSING60	60		1.58			1						2.42	
PWSING	PWSIN100	100		1.58			1						2.17	
PWSING	PWSI1000	1000		1.73			1						2.82	
QUARTC	QUARTC	25		1.78			1						3.44	
ROSENB	SHNRSN10	10		1.67			1						4.83	
SCHMVT	SCHMVT50	50		1.25			1						1.83	
SCHMVT	SCHMV500	500		1.11					1.44		1			
SCHMVT	SCHV1000	1000	1					1.56			1			
SQRTMX	MSQRTB49	49			2.63		1					2.53		
TDQUAD	TDQ10	10	1					1.25			1			
TDQUAD	TDQ500	500	1					1.67					6.67	
TDQUAD	TDQ1000	1000	1					1.67					8.67	
TOINT	PSPTOINT	50			1.83		1					1.44		
TRIDIA	TRIDIA10	10			1.62		1					1.50		
TRIDIA	TRLN100	100			2.46			1.62			1			
TRIDIA	TRLN1000	1000		1.42			1						2.75	
TRIGTO	TRIGT50	50			5.38		1					4.05		
TRIGTO	TRIGT100	100			4.92		1					2.14		
VARDIM	VARDIM	10			1.83		1					1.17		
VARDIM	VARDM100	100			1.38		1					1.25		
VAROSB	VAROSBG1	50		1.28			1						1.58	
VAROSB	VAROSBG2	100		1.25			1						1.57	
WATSON	WATSON12	12	1				1					1.20		
WOODS	WOODS80	80	1					1.38					2.75	
XTX	XTX16	16	1				1				1			
Totals			17	32	27	2	63	11	1	3	10	31	29	8

Table C.22: Place finishes for number of gradient and Hessian-vector product evaluations, tolerance $= 10^{-6}$.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10			1.87		1					1.55		
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10	1					1.33					3.33	
BRWNAL	BRWNL100	100	1				1				1			
BRYBND	BRYBND	10		2.00			1							f
BRYBND	BRYBND18	100		1.50			1							f
BRYTRI	BRYTRI6	20		1.07			1						1.33	
BRYTRI	BRYTRI10	600		1.04			1						2.93	
CHEBYQ	CHEBYQ10	10			1.67		1					1.36		
CHNRSN	CHNRSH10	10			2.87		1					1.29		
CRGLVY	CRGLVY10	10		3.00			1						3.05	
CRGLVY	CRGLY500	500		1.35			1						8.35	
CRGLVY	CRGL1000	1000		1.52			1						10.43	
DIX7DG	DIX7DGA	15		1.80			1					1.80		
DIXON	DIXON	10		1.19			1						1.57	
ENGVL1	ENGVL1B2	10		1.83			1						2.50	
ENGVL1	ENGVL1B4	100		1.75			1						2.00	
ENGVL1	ENGVL1B6	1000		1.33			1						3.83	
EXTRSN	EXTRAR50	50			2.02		1					1.41		
EXTRSN	EXTRA100	100			2.16		1					1.70		
EXTRSN	EXTR1000	1000			2.11		1					1.52		
FRANK	FRANK12	12	1					2.50					2.79	
FRDRTH	FRDRTHB3	50			2.53		1					1.41		
FRDRTH	FRDRTHB4	100			2.29		1					1.59		
FRDRTH	FRDRTHB7	1000		1.59			1						2.76	
GENRSN	GENT50A	50			2.96		1					1.82		
GENRSN	GENT500A	500			3.09		1					1.80		
GENRSN	GENT1000	1000			3.11		1					1.82		
HILBRT	HILBRT12	12			1.50			1.10			1			
MANCIN	MANCIN50	50		1.40			1						4.60	
MNSRF1	NMSURF64	36			2.23		1					1.77		
MNSRF1	NMSUR484	400			9.57		1					2.43		
MNSRF2	SNMSUR64	36			2.15		1					1.63		
MNSRF2	SNMSR484	400			11.26		1					2.35		
MORCIN	MORCIN10	10		1.20			1						1.40	
MOREBV	MOREBV18	18			1.28			1.09			1			
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20			2.80		1					1.80		
NONDIA	NONDI500	500			3.41		1					1.25		
NONDIA	NOND1000	1000			3.34		1					1.32		
OSBRN2	OSBRN2	11			1.79			1.59			1			
PENAL1	PEN1LN1	50		1.19			1						3.56	
PENAL1	PEN1LN2	100		1.19			1						7.56	
PENAL1	PEN1LN3	1000	1				1					186.37		
PENAL2	PEN2GM1	50			1.11		1					1.09		
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50			1.26		1					1.23		
PENAL3	PENL3GM4	100		1.38			1							f
PENAL3	PENL3GM5	1000		1.54			1							f
POWBSC	POWBSC50	50		1.71			1							f
POWBSC	POWBS500	500		1.40			1							f
POWBSC	POWB1000	1000		1.22			1							f
POWER	POWER75	75		2.06			1						2.19	
PWSING	PWSING60	60		2.05			1						2.37	
PWSING	PWSIN100	100		2.05			1						2.42	
PWSING	PWSI1000	1000		2.17			1						2.28	
QUARTC	QUARTC	25		1.87			1						2.40	
ROSENB	SHNRSN10	10			6.72			2.00			1			
SCHMVT	SCHMVT50	50		1.16			1						1.52	
SCHMVT	SCHMV500	500	1				1					1.80		
SCHMVT	SCHV1000	1000	1					1.08					1.92	
SQRTMX	MSQRTB49	49			1.56		1					1.51		
TDQUAD	TDQ10	10		1.29					1.71		1			
TDQUAD	TDQ500	500	1					1.30					2.00	
TDQUAD	TDQ1000	1000	1					1.10					2.60	
TOINT	PSPTOINT	50			1.43		1					1.25		
TRIDIA	TRIDIA10	10			1.92		1					1.15		
TRIDIA	TRLN100	100			1.42			1.20			1			
TRIDIA	TRLN1000	1000			1.27		1					1.04		
TRIGTO	TRIGT50	50			3.46		1					2.51		
TRIGTO	TRIGT100	100			3.28		1					1.33		
VARDIM	VARDIM	10			1.73			1.18			1			
VARDIM	VARDM100	100		1.36			1				1			
VAROSB	VAROSBG1	50			1.55		1					1.21		
VAROSB	VAROSBG2	100			3.10		1					1.25		
WATSON	WATSON12	12		1.01					2.21		1			
WOODS	WOODS80	80			2.29		1					1.61		
XTX	XTX16	16	1				1				1			
Totals			9	31	35	3	62	11	2	3	11	32	25	10

Table C.23: Place finishes for number of gradient and Hessian-vector product evaluations, tolerance = 10^{-9}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10			1.68		1					1.39		
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10		1.08				1.08			1			
BRWNAL	BRWNL100	100	1				1				1			
BRYBND	BRYBND	10		2.00			1							f
BRYBND	BRYBND18	100		1.50			1							f
BRYTRI	BRYTRI6	20		1.14			1						1.43	
BRYTRI	BRYTRI10	600	1					1.05					3.76	
CHEBYQ	CHEBYQ10	10			1.81		1					1.43		
CHNRSN	CHNRSH10	10			2.80		1					1.26		
CRGLVY	CRGLVY10	10			2.06		1					1.86		
CRGLVY	CRGLY500	500		2.11			1						7.53	
CRGLVY	CRGL1000	1000		1.86			1						9.14	
DIX7DG	DIX7DGA	15			1.82		1					1.64		
DIXON	DIXON	10			1.23		1					1.16		
ENGVL1	ENGVL1B2	10		1.86			1						2.36	
ENGVL1	ENGVL1B4	100		1.79			1						2.29	
ENGVL1	ENGVL1B6	1000		1.73			1						4.13	
EXTRSN	EXTRAR50	50			2.13		1					1.45		
EXTRSN	EXTRA100	100			2.22		1					1.69		
EXTRSN	EXTR1000	1000			2.17		1					1.52		
FRANK	FRANK12	12	1					3.43					3.64	
FRDRTH	FRDRTHB3	50			2.26		1					1.58		
FRDRTH	FRDRTHB4	100			2.05		1					1.79		
FRDRTH	FRDRTHB7	1000		2.42			1						3.95	
GENRSN	GENT50A	50			2.98		1					1.79		
GENRSN	GENT500A	500			3.09		1					1.79		
GENRSN	GENT1000	1000			3.11		1					1.82		
HILBRT	HILBRT12	12		1.31					2.06		1			
MANCIN	MANCIN50	50		1.50			1						4.83	
MNSRF1	NMSURF64	36			1.82		1					1.25		
MNSRF1	NMSUR484	400			7.19		1					1.81		
MNSRF2	SNMSUR64	36			1.55		1					1.23		
MNSRF2	SNMSR484	400			8.27		1					1.76		
MORCIN	MORCIN10	10		1.50			1						1.67	
MOREBV	MOREBV18	18			1.54			1.24			1			
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20			2.98		1					1.88		
NONDIA	NONDI500	500			4.09		1					1.26		
NONDIA	NOND1000	1000			4.28		1					1.30		
OSBRN2	OSBRN2	11		1.83					1.97		1			
PENAL1	PEN1LN1	50		1.23			1						2.59	
PENAL1	PEN1LN2	100		1.23			1						5.64	
PENAL1	PEN1LN3	1000		1.08			1						145.48	
PENAL2	PEN2GM1	50			1.22			1.02			1			
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50			1.16		1					1.14		
PENAL3	PENL3GM4	100		1.15			1							f
PENAL3	PENL3GM5	1000		1.47			1							f
POWBSC	POWBSC50	50		1.48			1							f
POWBSC	POWBS500	500		1.36			1							f
POWBSC	POWB1000	1000		2.55			1							f
POWER	POWER75	75			2.12		1					1.64		
PWSING	PWSING60	60			2.37		1					2.04		
PWSING	PWSIN100	100			1.56		1					1.20		
PWSING	PWSI1000	1000			1.78		1					1.47		
QUARTC	QUARTC	25		1.82			1						1.86	
ROSENB	SHNRSN10	10			36.87			3.71			1			
SCHMVT	SCHMVT50	50		1.10			1						1.22	
SCHMVT	SCHMV500	500		1.05			1						2.11	
SCHMVT	SCHV1000	1000	1					1.18					2.45	
SQRTMX	MSQRTB49	49			1.91		1					1.12		
TDQUAD	TDQ10	10		1.71					2.00		1			
TDQUAD	TDQ500	500	1					1.50					2.00	
TDQUAD	TDQ1000	1000	1					1.40					2.60	
TOINT	PSPTOINT	50			1.11		1					1.09		
TRIDIA	TRIDIA10	10			1.72		1					1.17		
TRIDIA	TRLN100	100			1.50			1.37			1			
TRIDIA	TRLN1000	1000			2.20			1.32			1			
TRIGTO	TRIGT50	50			2.76		1					1.82		
TRIGTO	TRIGT100	100			2.64			1.02			1			
VARDIM	VARDIM	10			1.47		1					1.06		
VARDIM	VARDM100	100			1.50			1.11			1			
VAROSB	VAROSBG1	50			1.76		1					1.33		
VAROSB	VAROSBG2	100			3.68			1.03			1			
WATSON	WATSON12	12	1						70.98			62.65		
WOODS	WOODS80	80			2.44		1					1.62		
XTX	XTX16	16	1				1				1			
Totals			8	26	41	3	57	14	4	3	14	34	20	10

Table C.24: Place finishes for number of gradient and Hessian-vector product evaluations, tolerance = 10^{-12}.

Function	Problem	Size	UMINHV				L-BFGS				TN			
			1	2	3	f	1	2	3	f	1	2	3	f
ARTRIG	ARTRIG10	10			1.67		1					1.41		
BROY7D	BROY7D	60				f				f				f
BRWNAL	BRWNAL10	10		1.81			1				1			
BRWNAL	BRWNL100	100	1				1				1			
BRYBND	BRYBND	10		1.92			1							f
BRYBND	BRYBND18	100		1.79			1							f
BRYTRI	BRYTRI6	20		1.07			1						1.33	
BRYTRI	BRYTRI10	600	1				1					3.80		
CHEBYQ	CHEBYQ10	10		1.68			1						2.02	
CHNRSN	CHNRSH10	10			2.41		1					1.15		
CRGLVY	CRGLVY10	10			1.77		1					1.45		
CRGLVY	CRGLY500	500		1.80			1						3.74	
CRGLVY	CRGL1000	1000		2.17			1						6.29	
DIX7DG	DIX7DGA	15			1.85		1					1.38		
DIXON	DIXON	10			1.39		1					1.05		
ENGVL1	ENGVL1B2	10		1.94			1						2.44	
ENGVL1	ENGVL1B4	100		2.12			1						2.44	
ENGVL1	ENGVL1B6	1000		1.71			1						4.82	
EXTRSN	EXTRAR50	50			2.19		1					1.48		
EXTRSN	EXTRA100	100			2.28		1					1.72		
EXTRSN	EXTR1000	1000			2.23		1					1.57		
FRANK	FRANK12	12	1					4.00					4.64	
FRDRTH	FRDRTHB3	50			2.81		1					1.76		
FRDRTH	FRDRTHB4	100			2.52		1					2.29		
FRDRTH	FRDRTHB7	1000		2.19			1						4.52	
GENRSN	GENT50A	50			2.95		1					1.77		
GENRSN	GENT500A	500			3.09		1					1.80		
GENRSN	GENT1000	1000			3.11		1					1.82		
HILBRT	HILBRT12	12	1						3.68			1.14		
MANCIN	MANCIN50	50		1.86			1						5.00	
MNSRF1	NMSURF64	36			1.54		1					1.04		
MNSRF1	NMSUR484	400			6.03		1					1.53		
MNSRF2	SNMSUR64	36			1.53		1					1.07		
MNSRF2	SNMSR484	400			6.85		1					1.48		
MORCIN	MORCIN10	10		1.38			1						1.62	
MOREBV	MOREBV18	18			1.49			1.28			1			
MOREBV	MORBV998	998				f				f				f
NONDIA	NONDIA20	20			3.09		1					1.84		
NONDIA	NONDI500	500			4.32		1					1.30		
NONDIA	NOND1000	1000			4.66		1					1.36		
OSBRN2	OSBRN2	11		1.76					2.07		1			
PENAL1	PEN1LN1	50		1.22			1						2.30	
PENAL1	PEN1LN2	100		1.25			1						4.57	
PENAL1	PEN1LN3	1000		1.16			1						114.19	
PENAL2	PEN2GM1	50			1.34			1.10			1			
PENAL2	PEN2GM2	100				f				f				f
PENAL3	PENL3GM3	50			1.19		1					1.12		
PENAL3	PENL3GM4	100		1.13			1							f
PENAL3	PENL3GM5	1000		1.40			1							f
POWBSC	POWBSC50	50	1							f				f
POWBSC	POWBS500	500	1							f				f
POWBSC	POWB1000	1000	1							f				f
POWER	POWER75	75			2.09		1					1.56		
PWSING	PWSING60	60			1.56		1					1.28		
PWSING	PWSIN100	100		1.45			1				1			
PWSING	PWSI1000	1000			1.58		1					1.17		
QUARTC	QUARTC	25			1.96		1					1.64		
ROSENB	SHNRSN10	10			12.29			1.02			1			
SCHMVT	SCHMVT50	50			1.09		1					1.05		
SCHMVT	SCHMV500	500	1					1.12					2.38	
SCHMVT	SCHV1000	1000	1				1					2.49		
SQRTMX	MSQRTB49	49			2.34		1					1.11		
TDQUAD	TDQ10	10		1.20					1.60		1			
TDQUAD	TDQ500	500	1					1.33					1.67	
TDQUAD	TDQ1000	1000	1					1.33					2.17	
TOINT	PSPTOINT	50	1					1.01					1.03	
TRIDIA	TRIDIA10	10			2.05		1					1.38		
TRIDIA	TRLN100	100			1.77			1.50			1			
TRIDIA	TRLN1000	1000			3.09			1.64			1			
TRIGTO	TRIGT50	50			2.52		1					1.59		
TRIGTO	TRIGT100	100			2.84			1.19			1			
VARDIM	VARDIM	10			1.50		1					1.11		
VARDIM	VARDM100	100			1.68			1.18			1			
VAROSB	VAROSBG1	50			1.55		1					1.34		
VAROSB	VAROSBG2	100			4.12		1					1.06		
WATSON	WATSON12	12	1							f				f
WOODS	WOODS80	80			2.47		1					1.65		
XTX	XTX16	16	1				1				1			
Totals			14	21	40	3	56	12	3	7	13	36	18	11

Bibliography

[1] Melvyn S. Berger. *Nonlinearity and Functional Analysis: Lectures on Nonlinear Problems in Mathematical Analysis.* Academic Press, 1977.

[2] Christian Bischof, Alan Carle, Peyvand Khademi, and Andrew Mauer. The ADIFOR 2.0 system for the automatic differentiation of Fortran 77 programs. Technical Report ANL-MCS-P481-1194, Argonne National Laboratory, 1994.

[3] Christian Bischof, Alan Carle, Peyvand Khademi, Andrew Mauer, and Paul Hovland. ADIFOR 2.0 user's guide. Technical Report ANL-MCS-TM-192, Argonne National Laboratory, 1995.

[4] Albert G. Buckley. Test functions for unconstrained minimization. Technical report, Department of Mathematics, Royal Roads Military College, 1994.

[5] Augustin Cauchy. Méthode générale pour la résolution des systèmes d'équations simultanées. *C. R. Acad. Sci. Paris*, 25(16):536–538, 1847.

[6] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bull. Amer. Math. Soc.*, 49:1–23, 1943.

[7] Nelson Dunford and Jacob T. Schwartz. *Linear Operators: General Theory*, volume 1. Wiley, 1958.

[8] David M. Gay. Usage summary for selected optimization routines. Technical Report CSTR 153, AT&T Bell Laboratories, 1990.

[9] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization.* Academic Press, 1981.

[10] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[11] Jacques Hadamard. Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées. *Mémoires présentés par divers savants à l'Académie des Sciences de l'Institut National de France, Series 2*, 33(4):1–128, 1908.

[12] Uwe Helmke and John B. Moore. *Optimization and Dynamical Systems*. Springer-Verlag, 1994.

[13] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards*, 45(4):255–282, 1950.

[14] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45(3):503–528, 1989.

[15] Jorge J. Moré and Stephen J. Wright. *Optimization Software Guide*. Society for Industrial and Applied Mathematics, 1993.

[16] Stephen G. Nash. Newton-type minimization via the Lanczos method. *SIAM J. Numer. Anal.*, 21(4):770–788, 1984.

[17] Stephen G. Nash and Jorge Nocedal. A numerical study of the limited memory BFGS method and the truncated-Newton method for large scale optimization. *SIAM J. Optim.*, 1(3):358–372, 1991.

[18] Numerical Algorithms Group. *The NAG Fortran Library Manual, Mark 17*, 1996.

[19] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12(4):617–629, 1975.

[20] R. S. Palais and S. Smale. A generalized Morse theory. *Bull. Amer. Math. Soc.*, 70:165–172, 1964.

[21] Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Prentice-Hall, 1980.

[22] Michael Reed and Barry Simon. *Methods of Modern Mathematical Physics : Functional Analysis*, volume 1. Academic Press, 1980.

[23] Horst D. Simon. Analysis of the symmetric Lanczos algorithm with reorthogonalization methods. *Linear Algebra Appl.*, 61:101–131, 1984.

[24] Horst D. Simon. The Lanczos algorithm with partial reorthogonalization. *Math. Comp.*, 42(165):115–142, 1984.

[25] Stephen Wolfram. *The Mathematica Book*. Cambridge University Press, 1996.