Chapter E04

Minimizing or Maximizing a Function

Contents

1 Scope of the Chapter

An optimization problem involves minimizing a function (called the objective function) of several variables. The NAG Parallel Library is concerned with function minimization only, since the problem of maximizing a given function can be transformed into a minimization problem simply by multiplying the function by -1.

This introduction is only a brief guide to the subject of optimization designed for the casual user. Anyone with a difficult or protracted problem to solve will find it beneficial to consult a more detailed text, such as Gill *et al.* [3] or Fletcher [2].

Readers who are unfamiliar with the mathematics of the subject may find some sections difficult at first reading; if so, they should concentrate on Section 2.1, Section 2.2, Section 2.5 and Section 2.6.

2 Background to the Problems

2.1 Types of Optimization Problems

In this release of the NAG Parallel Library, only unconstrained nonlinear least-squares problems and unconstrained nonlinear general problems are dealt with.

2.1.1 Unconstrained minimization

In unconstrained minimization problems there are no constraints on the variables. The problem can be stated mathematically as follows:

$$\min_x F(x)$$

where $x \in \mathbb{R}^n$, that is, $x = (x_1, x_2, ..., x_n)^T$.

2.1.2 Nonlinear least-squares problems

Special consideration is given to the problem for which the function to be minimized can be expressed as a sum of squared functions. The least-squares problem can be stated mathematically as follows:

$$\min_x F(x) = \sum_{i=1}^{m} f_i^2(x), \quad x \in \mathbb{R}^n,$$

where the ith element of the m-vector f is the function $f_i(x)$. The functions $f_i(x)$ are often referred to as residuals because they often represent the residuals in a nonlinear regression problem.

2.2 Geometric Representation and Terminology

To illustrate the nature of optimization problems it is useful to consider the following example in two dimensions:

$$F(x) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1).$$

Figure 1 is a contour diagram of $F(x)$. The contours labelled $F_0, F_1, ..., F_4$ are isovalue contours, or lines along which the function $F(x)$ takes specific constant values. The point x^* is a local unconstrained minimum, that is, the value of $F(x^*)$ is less than at all the neighbouring points. A function may have several such minima. The lowest of the local minima is termed a global minimum. In the problem illustrated in Figure 1, x^* is the only local minimum. The point $\bar{x}$ is said to be a saddle point because it is a minimum along the line AB, but a maximum along CD.

2.2.1 Gradient vector

The vector of first partial derivatives of $F(x)$ is called the gradient vector, and is denoted by $g(x)$, i.e.,

$$g(x) = \left[\frac{\partial F(x)}{\partial x_1}, \frac{\partial F(x)}{\partial x_2}, ..., \frac{\partial F(x)}{\partial x_n} \right]^T.$$

For the function illustrated in Figure 1,

$$g(x) = \begin{bmatrix} F(x) + e^{x_1}(8x_1 + 4x_2) \\ e^{x_1}(4x_2 + 4x_1 + 2) \end{bmatrix}.$$

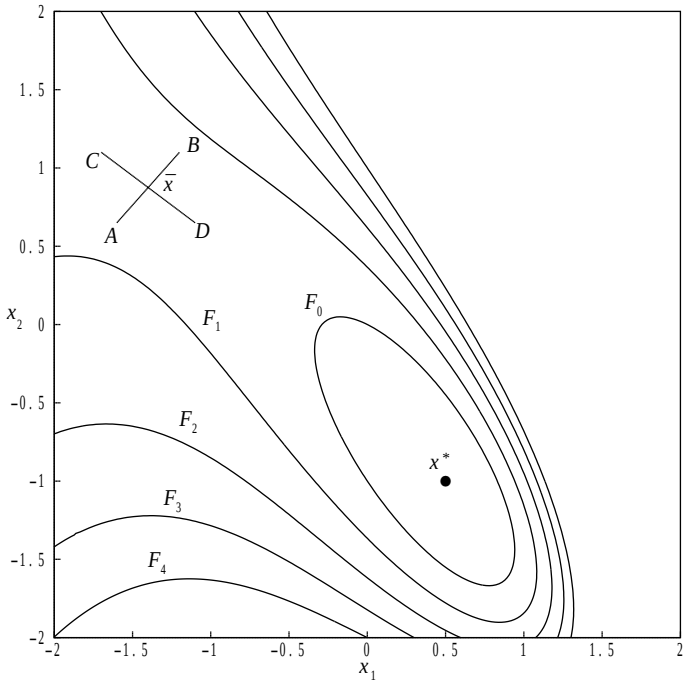

Figure 1

The gradient vector is of importance in optimization because it must be zero at an unconstrained minimum of any function with continuous first derivatives.

2.2.2 Hessian matrix

The matrix of second partial derivatives of a function is termed its Hessian matrix. The Hessian matrix of $F(x)$ is denoted by $G(x)$, and its (i,j)th element is given by $\dfrac{\partial^2 F(x)}{\partial x_i \partial x_j}$. If $F(x)$ has continuous second derivatives, then $G(x)$ must be positive semi-definite at any unconstrained minimum of F.

2.2.3 Jacobian matrix

In nonlinear least-squares problems, the matrix of first partial derivatives of the vector-valued function $f(x)$ is termed the Jacobian matrix of $f(x)$ and its (i,j)th component is $\dfrac{\partial f_i}{\partial x_j}$. In this case the gradient vector g is given by $g = 2J^T f$, where J is the Jacobian matrix.

2.3 Sufficient Conditions for a Solution

All nonlinear functions will be assumed to have continuous second derivatives in the neighbourhood of the solution.

2.3.1 Unconstrained minimization

The following conditions are sufficient for the point x^* to be an unconstrained local minimum of $F(x)$:

(i) $\|g(x^*)\| = 0$; and

(ii) $G(x^*)$ is positive-definite,

where $\|g\|$ denotes the Euclidean norm of g.

2.4 Background to Optimization Methods

The algorithm contained in this chapter generates an iterative sequence $x^{(k)}$ that converges to the solution x^* in the limit. To terminate computation of the sequence, a convergence test is performed to determine

whether the current estimate of the solution is an adequate approximation. The convergence tests are discussed in Section 2.6.

The method constructs a sequence $x^{(k)}$ satisfying

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)},$$

where the vector $p^{(k)}$ is termed the direction of search, and $\alpha^{(k)}$ is the steplength. The steplength $\alpha^{(k)}$ is chosen so that $F(x^{(k+1)}) < F(x^{(k)})$ and is computed using a technique for 1-d optimization referred to in Gill *et al.* [3].

2.4.1 Methods for unconstrained optimization

The algorithm used in this chapter is a Newton-type method.

Newton-type methods use the Hessian matrix $G(x^{(k)})$, or a finite difference approximation to $G(x^{(k)})$, to define the search direction.

Newton-type methods are the most powerful methods available for general problems and will find the minimum of a quadratic function in one iteration. See Sections 4.4 and 4.5.1 of Gill *et al.* [3].

2.4.2 Methods for nonlinear least-squares problems

These methods exploit the special structure of the Hessian matrix to give improved computational efficiency.

Since

$$F(x) = \sum_{i=1}^{m} f_i^2(x)$$

the Hessian matrix $G(x)$ is of the form

$$G(x) = 2\left(J(x)^T J(x) + \sum_{i=1}^{m} f_i(x) G_i(x)\right),$$

where $J(x)$ is the Jacobian matrix of $f(x)$, and $G_i(x)$ is the Hessian matrix of $f_i(x)$.

In the neighbourhood of the solution, $\|f(x)\|$ is often small compared to $\|J(x)^T J(x)\|$ (for example, when $f(x)$ represents the goodness of fit of a nonlinear model to observed data). In such cases, $2J(x)^T J(x)$ may be an adequate approximation to $G(x)$, thereby avoiding the need to compute or approximate second derivatives of $f_i(x)$. Such methods are called Gauss–Newton methods, see Section 4.7 of Gill *et al.* [3]. The routine in the NAG Parallel Library approximates $J(x^{(k)})$ by finite differences.

2.5 Scaling

Scaling (in a broadly defined sense) often has a significant influence on the performance of optimization methods. Since convergence tolerances and other criteria are necessarily based on an implicit definition of 'small' and 'large', problems with unusual or unbalanced scaling may cause difficulties for some algorithms. There are currently no scaling routines in the Library. In light of the present state of the art, it is considered that sensible scaling by the user is likely to be more effective than any automatic routine. The following sections present some general comments on problem scaling.

2.5.1 Transformation of variables

One method of scaling is to transform the variables from their original representation, which may reflect the physical nature of the problem, to variables that have certain desirable properties in terms of optimization. It is generally helpful for the following conditions to be satisfied:

(i) the variables are all of similar magnitude in the region of interest;

(ii) a fixed change in any of the variables results in similar changes in $F(x)$. Ideally, a unit change in any variable produces a unit change in $F(x)$;

(iii) the variables are transformed so as to avoid cancellation error in the evaluation of $F(x)$.

Normally, users should restrict themselves to linear transformations of variables, although occasionally nonlinear transformations are possible. The most common such transformation (and often the most appropriate) is of the form

$$x_{\text{new}} = Dx_{\text{old}},$$

where D is a diagonal matrix with constant coefficients. Our experience suggests that more use should be made of the transformation

$$x_{\text{new}} = Dx_{\text{old}} + v,$$

where v is a constant vector.

Consider, for example, a problem in which the variable x_3 represents the position of the peak of a Gaussian curve to be fitted to data for which the extreme values are 150 and 170; therefore x_3 is known to lie in the range 150–170. One possible scaling would be to define a new variable $\bar{x}_3$, given by

$$\bar{x}_3 = \frac{x_3}{170}.$$

A better transformation, however, is given by defining $\bar{x}_3$ as

$$\bar{x}_3 = \frac{x_3 - 160}{10}.$$

Frequently, an improvement in the accuracy of evaluation of $F(x)$ can result if the variables are scaled before the routines to evaluate $F(x)$ are coded. For instance, in the above problem of Gaussian curve fitting, x_3 may always occur in terms of the form $(x_3 - x_{\text{m}})$, where x_{m} is a constant representing the mean peak position.

2.5.2 Scaling the objective function

The objective function has already been mentioned in the discussion of scaling the variables. The solution of a given problem is unaltered if $F(x)$ is multiplied by a positive constant, or if a constant value is added to $F(x)$. It is generally preferable for the objective function to be of the order of unity in the region of interest; thus, if in the original formulation $F(x)$ is always of the order of 10^{+5} (say), then the value of $F(x)$ should be multiplied by 10^{-5} when evaluating the function within the optimization routines. If a constant is added or subtracted in the computation of $F(x)$, usually it should be omitted – i.e., it is better to formulate $F(x)$ as $x_1^2 + x_2^2$ rather than as $x_1^2 + x_2^2 + 1000$ or even $x_1^2 + x_2^2 + 1$. The inclusion of such a constant in the calculation of $F(x)$ can result in a loss of significant figures.

2.6 Analysis of Computed Results

2.6.1 Convergence criteria

The convergence criteria inevitably vary between different classes of problems. Nonetheless, the underlying principles of the various criteria are the same; in non-mathematical terms, they are:

(i) is the sequence $x^{(k)}$ converging?

(ii) is the sequence $F(x^{(k)})$ converging?

(iii) are the necessary and sufficient conditions for the solution satisfied?

The decision as to whether a sequence is converging is necessarily speculative. The criterion used is to assume convergence if the relative change occurring between two successive iterations is less than some prescribed quantity. Criterion (iii) is the most reliable but often the conditions cannot be checked fully because not all the required information may be available.

2.6.2 Checking results

Little *a priori* guidance can be given as to the quality of the solution found by a nonlinear optimization algorithm, since no guarantees can be given that the methods will always work. Therefore, it is necessary for the user to check the computed solution even if the routine reports success. Frequently a 'solution' may have been found even when the routine does not report a success. The reason for this apparent contradiction is that the routine needs to assess the accuracy of the solution. This assessment is not an exact process and consequently may be unduly pessimistic. Any 'solution' is in general only an

approximation to the exact solution, and it is possible that the accuracy specified by the user is too stringent.

Further confirmation can be sought by trying to check whether or not convergence tests are almost satisfied, or whether or not some of the sufficient conditions are nearly satisfied. When it is thought that a routine has returned a non-zero value of IFAIL only because the requirements for 'success' were too stringent it may be worth restarting with an increased convergence tolerance.

Confidence in a solution may be increased by resolving the problem with a different initial approximation to the solution. See Section 8.3 of Gill *et al.* [3] for further information.

2.6.3 Block Algorithms

The optimization method used in this chapter for the least-squares problem requires the solution of a linear least-squares problem at each iteration. The optimization method used in this chapter for the general nonlinear problem requires the solution of a system of linear equations. The solutions to these systems of equations are obtained using routines from Chapter F07 or Chapter F08, and involve the use of block algorithms for performing an LU or QR factorization. A cyclic 2-d block distribution of the matrices with a row and column block size of N_b is used. It is not necessary for the use of this chapter to understand the details of the algorithm or the distribution, other than to be aware that the performance of a block algorithm varies to some extent with the block size. In general this is a machine dependent parameter. See Chapter F07 or Chapter F08 for further details of the solution of these systems of equations.

2.7 References

[1] Bard Y (1974) *Nonlinear Parameter Estimation* Academic Press

[2] Fletcher R (1987) *Practical Methods of Optimization* Wiley (2nd Edition)

[3] Gill P E, Murray W and Wright M H (1981) *Practical Optimization* Academic Press

[4] Wolberg J R (1967) *Prediction Analysis* Van Nostrand

3 Recommendations on Choice and Use of Available Routines

Note: Refer to the Users' Note for your implementation to check that a routine is available.

In this release of the Library there are two easy-to-use routines: E04FDFP for the unconstrained nonlinear least-squares problems, and E04JBFP for the general unconstrained problem.