Mathematical Programming 14 (1978) 224–248.
North-Holland Publishing Company

ALGORITHMS FOR NONLINEAR CONSTRAINTS THAT USE LAGRANGIAN FUNCTIONS*

M.J.D. POWELL

University of Cambridge, Cambridge, United Kingdom

Received 10 October 1976
Revised manuscript 23 August 1977

Lagrangian functions are the basis of many of the more successful methods for nonlinear constraints in optimization calculations. Sometimes they are used in conjunction with linear approximations to the constraints and sometimes penalty terms are included to allow the use of algorithms for unconstrained optimization. Much has been discovered about these techniques during the last eight years and this paper gives a view of the progress and understanding that has been achieved and its relevance to practical algorithms. A particular method is recommended that seems to be more powerful than the author believed to be possible at the beginning of 1976.

Key words: Augmented Lagrangian, Lagrangian Function, Nonlinear Constraints, Nonlinear Programming, Optimization Algorithm

1. Introduction

The general nonlinear programming problem is to calculate the least value of a function $F(\boldsymbol{x})$ of several variables subject to constraints on the variables, where each constraint has one of the forms

$$c_i(\boldsymbol{x}) = 0, \qquad c_i(\boldsymbol{x}) \geq 0. \tag{1.1}$$

When the constraint functions are nonlinear then a balance has to be achieved during the calculation between satisfying the constraints and reducing the objective function. Until recently most of the methods that were used were rather clumsy, but the properties of Lagrangian functions provide a natural balance which has led to the development of some excellent algorithms for constrained optimization. This work is surveyed. Most of its impact on practical problems is still to come.

In order to simplify the presentation we suppose that several convenient conditions are obtained. We assume third order differentiability whenever it is helpful to do so. Defining $\boldsymbol{x}^*$ to be the required solution and I^* to be the set of

[*] Presented at the Ninth International Symposium on Mathematical Programming, Budapest (August 1976).

indices i such that $c_i(\boldsymbol{x}^*)$ is zero, we assume that the constraint normals $\{\nabla c_i(\boldsymbol{x}); i \in I^*\}$ are linearly independent at x^*. Therefore there exist unique parameters λ_i^* $(i \in I^*)$ such that the equation

$$\nabla F(\boldsymbol{x}^*) - \sum_{i \in I^*} \lambda_i^* \nabla c_i(\boldsymbol{x}^*) = 0 \tag{1.2}$$

is satisfied. We require the parameters of the inequality constraints in I^* to be positive. We also require the existence of a positive constant, q say, such that the inequality

$$\boldsymbol{y}^{\mathrm{T}} G(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) \boldsymbol{y} \geq q \|\boldsymbol{y}\|^2, \tag{1.3}$$

holds for every vector $\boldsymbol{y}$ that satisfies the conditions

$$\boldsymbol{y}^{\mathrm{T}} \nabla c_i(\boldsymbol{x}^*) = 0, \quad i \in I^*. \tag{1.4}$$

Here the superscript T distinguishes a row vector from a column vector and $G(\boldsymbol{x}, \boldsymbol{\lambda})$ is the second derivative matrix with respect to $\boldsymbol{x}$ of the Lagrangian function

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = F(\boldsymbol{x}) - \sum_{i \in I^*} \lambda_i c_i(\boldsymbol{x}). \tag{1.5}$$

These conditions are required by many of the published algorithms for solving the general nonlinear programming problem. When they hold then second order convergence to $\boldsymbol{x}^*$ can be obtained by iterative methods that make use of first and second derivatives of $F(\boldsymbol{x})$ and the constraint functions. For example we can apply the classical Newton iteration to solve the nonlinear equations (1.2) and

$$c_i(\boldsymbol{x}^*) = 0, \quad i \in I^*, \tag{1.6}$$

where the unknowns are the components of $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$. Our approach to comparing published algorithms for the nonlinearly constrained minimization calculation is to relate them to the Newton iteration.

A similar approach is taken by Tapia [27] so basically his paper and the present one have much in common. However we take the view that local convergence theorems are less important than strategies that can force convergence from poor starting approximations. Therefore this paper is more concerned with techniques than with details. For example we make the useful remark that it is sometimes advantageous to make a positive definite approximation to the second derivative matrix $G(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ even when this second derivative matrix has some negative eigenvalues. Negative eigenvalues can occur because the left hand side of expression (1.3) is allowed to be negative when $\boldsymbol{y}$ does not satisfy the conditions (1.4).

Our survey does not study the early penalty function methods that do not make use of the Lagrangian parameters $\boldsymbol{\lambda}$, because they can achieve superlinear convergence only by introducing very ill-conditioned matrices or by an extrapolation procedure which has the disadvantage of requiring several accurate unconstrained minimization calculations. A good description of the development of penalty function methods is given by Ryan [25]. Also we do not study the generalized reduced gradient methods (see Sargent, [26], for instance), because they apply frequently a separate correction procedure to take account of nonlinearities in the constraints. This seems to be less efficient than some of the methods that we prefer, except on some large problems where the reduced gradient method can take advantage of sparsity in the definition of the constraint functions.

The Newton iteration for solving the nonlinear eqs. (1.2) and (1.6) is studied in Section 2. We note the derivatives that are used because later we make approximations to second derivatives in order that they do not have to be calculated. However often Newton's method converges to the required solution only when good initial estimates of $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$ are available. This problem is discussed. Also an extension to the Newton iteration is described to allow for inequality constraints.

An interesting method that treats successfully many of the calculations when Newton's method fails due to poor initial estimates of $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$ is suggested by Rosen and Kreuser [24] and by Robinson [21]. It is iterative. On each iteration an objective function is minimized in the space of the $\boldsymbol{x}$-variables subject to linear approximations to the constraints. The objective function is $F(\boldsymbol{x})$ augmented by some terms, which require estimates of the Lagrange multipliers, to allow for the curvature of the constraints. For instance the Lagrangian function (1.5) may be used. Some versions of this method are described and discussed in Section 3. We note that there are still some cases where failure may occur due to poor initial estimates of $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$.

Section 4 considers the method that I believed to be the best general technique until I started to prepare this paper a few months ago. It is the augmented Lagrangian method. Much has been published about its properties recently and the surveys by Bertsekas [3], Fletcher [10] and Rockafellar [23] are particularly recommended. The basis of the method is that the second order sufficiency conditions imply that $\boldsymbol{x}^*$ is a local unconstrained minimum of the augmented Lagrangian function

$$\Phi(\boldsymbol{x}, \boldsymbol{\lambda}, r) = F(\boldsymbol{x}) - \sum_{i \in I^*} \lambda_i c_i(\boldsymbol{x}) + r \sum_{i \in I^*} [c_i(\boldsymbol{x})]^2, \tag{1.7}$$

provided that $\lambda_i = \lambda_i^*$ $(i \in I^*)$ and that r is greater than a certain constant $\bar{r}$. Values of $\boldsymbol{\lambda}$ and r are chosen at the beginning of each iteration and then the function $\Phi(\boldsymbol{x}, \boldsymbol{\lambda}, r)$ is minimized with respect to $\boldsymbol{x}$. Unlike the method mentioned in the previous paragraph, there are now no constraints on $\boldsymbol{x}$ during each minimization calculation. The augmented Lagrangian method is suitable for general calculations because convergence from poor starting approximations can usually be achieved by choosing r large enough. The published papers on the algorithm are concerned with methods for adjusting $\boldsymbol{\lambda}$ and r, the tolerance of the unconstrained minimization calculations, convergence properties and extensions to inequality constraints. All of these topics are mentioned briefly in Section 4. Also the disadvantages of the method are discussed.

The final section of this paper tries to combine the advantages of all the given methods into a single efficient algorithm. A particular algorithm is recommended, due mainly to Han [15], which is the Newton method except that the second derivative matrix $G(\boldsymbol{x}, \boldsymbol{\lambda}^*)$ is approximated by a positive definite matrix and a safeguard is included to avoid divergence of the iterations. Numerical results are reported for one test problem and they are very encouraging. However the version of the algorithm that is described should be regarded as preliminary because very little practical experience has been obtained. Also some important theoretical questions are still open and the organisation of the matrix calculations has not yet been studied properly.

2. Newton's method

For most of this section we consider the case when the only constraints that are present are equality constraints. Therefore we require to calculate the values of the variables $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ that satisfy the nonlinear equations

$$\nabla F(\boldsymbol{x}) - \sum_{i=1}^{m} \lambda_i \nabla c_i(\boldsymbol{x}) = 0 \tag{2.1}$$

and

$$c_i(\boldsymbol{x}) = 0, \quad i = 1, 2, \dots, m, \tag{2.2}$$

where for convenience we have assumed that I^* is the set of integers $\{1, 2, \dots, m\}$. Some of the equality constraints may be inequality constraints that are binding at the solution. Later we interpret Newton's method in a way that allows us to include also inequality constraints whose boundaries are away from the solution $\boldsymbol{x}^*$.

In Newton's method for solving a system of nonlinear equations we obtain linear approximations to the residuals of the equations by calculating first derivatives. These linear approximations are equated to zero, which provides a new estimate of the unknowns. The process is repeated iteratively. Thus for the system of eqs. (2.1) and (2.2) we replace the estimate $(\boldsymbol{x}, \boldsymbol{\lambda})$ of the solution by the estimate $(\boldsymbol{x}^\dagger, \boldsymbol{\lambda}^\dagger)$, which is obtained by solving the linear system

$$\begin{pmatrix} G(\boldsymbol{x}, \boldsymbol{\lambda}) & -N(\boldsymbol{x}) \\ -N(\boldsymbol{x})^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^\dagger - \boldsymbol{x} \\ \boldsymbol{\lambda}^\dagger - \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\nabla F(\boldsymbol{x}) + N(\boldsymbol{x})\boldsymbol{\lambda} \\ \boldsymbol{c}(\boldsymbol{x}) \end{pmatrix}, \tag{2.3}$$

where $G(\boldsymbol{x}, \boldsymbol{\lambda})$ was defined in Section 1 to be the second derivative matrix

$$G(\boldsymbol{x}, \boldsymbol{\lambda}) = \nabla^2 F(\boldsymbol{x}) - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i(\boldsymbol{x}) \tag{2.4}$$

and where $N(x)$ is the matrix

$$N(\boldsymbol{x}) = \{\nabla c_1(\boldsymbol{x}), \nabla c_2(\boldsymbol{x}), \dots, \nabla c_m(\boldsymbol{x})\}, \tag{2.5}$$

whose columns are the constraint normals at $\boldsymbol{x}$. The conditions of Section 1 ensure that the Newton iteration has a quadratic rate of convergence (see Ortega and Rheinboldt [19], for instance).

There is much useful information in the iteration (2.3) which is important to the development of good algorithms for constrained optimization calculations. First we notice the derivative information that is needed to obtain second order convergence. We require first derivatives of the functions $F(\boldsymbol{x})$ and $c_i(\boldsymbol{x})$ $\{i = 1, 2, \dots, m\}$ and we require the second derivative matrix $G(\boldsymbol{x}, \boldsymbol{\lambda})$. When the constraints are curved, then $\nabla^2 F(\boldsymbol{x})$ is not very helpful unless the second derivatives $\nabla^2 c_i(\boldsymbol{x})$ $\{i = 1, 2, \dots, m\}$ are available also. However we do not need all the second derivatives separately but only the combination given in expression (2.4). Therefore in our discussion of algorithms we refer frequently to the second derivatives of the Lagrangian function but the second derivatives of the objective function are seldom mentioned. Eq. (2.3) shows that this attitude is appropriate and its advantages are confirmed by numerical examples [4].

The second interesting feature of eq. (2.3) is that the term on the right-hand side that involves $\boldsymbol{\lambda}$ cancels with a term on the left-hand side. Thus $\boldsymbol{\lambda}^\dagger$ and $\boldsymbol{x}^\dagger$ depend on $\boldsymbol{\lambda}$ only through the matrix $G(\boldsymbol{x}, \boldsymbol{\lambda})$. If we do not show the dependence of vectors and matrices on $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ and if the matrix G is non-singular then the eqs. (2.3) have the solution

$$\boldsymbol{\lambda}^\dagger = (N^{\mathrm{T}} G^{-1} N)^{-1} \{N^{\mathrm{T}} G^{-1} \nabla F - \boldsymbol{c}\}, \tag{2.6}$$

$$\boldsymbol{x}^\dagger - \boldsymbol{x} = -G^{-1} N (N^{\mathrm{T}} G^{-1} N)^{-1} \boldsymbol{c} + \{G^{-1} N (N^{\mathrm{T}} G^{-1} N)^{-1} N^{\mathrm{T}} G^{-1} - G^{-1}\} \nabla F. \tag{2.7}$$

It should be noted that we require G to be non-singular only in order that this form of the solution is valid. It is not a necessary condition for the linear system (2.3) to be solvable.

Besides eq. (2.6) many other methods have been proposed for revising an estimate of $\boldsymbol{\lambda}^*$. Whatever method is used, eq. (2.7) is often a good formula for improving an approximation to $\boldsymbol{x}^*$. One reason is that, if the sequence $\{\boldsymbol{\lambda}_k, k = 1, 2, 3, \dots\}$ converges to $\boldsymbol{\lambda}^*$, if $\boldsymbol{x}_{k+1}$ is the value of $\boldsymbol{x}^\dagger$ that is obtained by giving $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ the values $\boldsymbol{x}_k$ and $\boldsymbol{\lambda}_k$ in eq. (2.7) and if $\{\boldsymbol{x}_k; k = 1, 2, 3, \dots\}$ converges to $\boldsymbol{x}^*$, then, even if $\{\boldsymbol{\lambda}_k; k = 1, 2, 3, \dots\}$ approaches $\boldsymbol{\lambda}^*$ rather slowly, the rate of convergence of the sequence $\{\boldsymbol{x}_k; k = 1, 2, 3, \dots\}$ is superlinear.

A way of showing the truth of this statement is to let J_k be the matrix

$$J_k = \begin{pmatrix} G(\boldsymbol{x}_k, \boldsymbol{\lambda}_k) & -N(\boldsymbol{x}_k) \\ -N(\boldsymbol{x}_k)^{\mathrm{T}} & 0 \end{pmatrix}, \tag{2.8}$$

to let $\boldsymbol{\mu}_1$ be an arbitrary m-component vector and to calculate the sequence

$\{(\boldsymbol{x}_k, \boldsymbol{\mu}_k); k = 1, 2, 3, \ldots\}$ by solving the equations

$$J_k\begin{pmatrix} \boldsymbol{x}_{k+1} - \boldsymbol{x}_k \\ \boldsymbol{\mu}_{k+1} - \boldsymbol{\mu}_k \end{pmatrix} = \begin{pmatrix} -\nabla F(\boldsymbol{x}_k) + N(\boldsymbol{x}_k)\boldsymbol{\mu}_k \\ \boldsymbol{c}(\boldsymbol{x}_k) \end{pmatrix}. \tag{2.9}$$

It follows from eq. (2.7) that the sequence $\{\boldsymbol{x}_k; k = 1, 2, 3, \ldots\}$ is the same as before. It can also be seen that the iteration (2.9) is Newton's method applied to the equations

$$\nabla F(\boldsymbol{x}) - N(\boldsymbol{x})\boldsymbol{\mu} = 0, \qquad \boldsymbol{c}(\boldsymbol{x}) = 0, \tag{2.10}$$

where the unknowns are $\boldsymbol{x}$ and $\boldsymbol{\mu}$, except that on each iteration the Jacobian is approximated by the matrix (2.8) where k is the iteration number. Since it may be shown that $\{\boldsymbol{\mu}_k; k = 1, 2, 3, \ldots\}$ converges to $\boldsymbol{\lambda}^*$, it follows that the error in the Jacobian approximation tends to zero. Thus the sequence $\{\boldsymbol{x}_k; k = 1, 2, 3, \ldots\}$ converges to $\boldsymbol{x}^*$ at a superlinear rate, which is the required result. The sequence $\{\boldsymbol{\mu}_k; k = 1, 2, 3, \ldots\}$ also converges at a superlinear rate, which is unimportant. Here we are using a theorem that may be found in Ortega and Rheinboldt [19] under the heading "consistent approximations".

Another interesting and similar property is noted by Garcia–Palomares and Mangasarian [13]. It is that there is a positive number ϵ such that, if $\boldsymbol{\lambda}$ is within distance ϵ of $\boldsymbol{\lambda}^*$, then $\boldsymbol{x}^*$ is a point of attraction of the procedure for revising $\boldsymbol{x}$ that is given by eq. (2.7). Therefore we can obtain convergence to $\boldsymbol{x}^*$ from formula (2.7) even when the error $\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^*\|$ is bounded away from zero. However we will see in Section 4 that this nice feature is not obtained by the augmented Lagrangian method.

The extension of the Newton iteration to take account of inequality constraints on the variables arises from the fact that the value of $\boldsymbol{x}^\dagger - \boldsymbol{x}$ that solves the eqs. (2.3) can also be found by solving a quadratic programming problem. Specifically $\boldsymbol{x}^\dagger - \boldsymbol{x}$ is the value of $\boldsymbol{\delta}$ that makes the quadratic function

$$Q(\boldsymbol{\delta}) = \boldsymbol{\delta}^{\mathrm{T}}\nabla F(\boldsymbol{x}) + \tfrac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}G(\boldsymbol{x}, \boldsymbol{\lambda})\boldsymbol{\delta} \tag{2.11}$$

stationary, subject to the linear constraints

$$N(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\delta} + \boldsymbol{c}(\boldsymbol{x}) = 0. \tag{2.12}$$

Further, the value of $\boldsymbol{\lambda}^\dagger$ in eq. (2.3) is equal to the vector of Lagrange multipliers of this quadratic programming problem. When inequality constraints are present we continue to let $\boldsymbol{x}^\dagger - \boldsymbol{x}$ be the value of $\boldsymbol{\delta}$ that minimizes the function (2.11), but expression (2.12) is extended to include an inequality condition for each inequality constraint. For example the inequalities

$$h_i(\boldsymbol{x}) \geq 0, \quad i = 1, 2, \ldots, m', \tag{2.13}$$

provide the linear conditions

$$\nabla h_i(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\delta} + h_i(\boldsymbol{x}) \geq 0, \quad i = 1, 2, \ldots, m', \tag{2.14}$$

on $\boldsymbol{\delta}$. This extension of the Newton iteration is due to Wilson [28] and it is the basis of his algorithm called SOLVER. Note that, if $F(\boldsymbol{x})$ is a quadratic function and if all the constraint functions are linear, then $\boldsymbol{x}^*$ is obtained in one iteration for all initial estimates of $\boldsymbol{x}$ and $\boldsymbol{\lambda}$.

If this method were successful consistently then the subject of constrained optimization would be much easier and there would be little need for the more sophisticated techniques that are described later. However Newton's method is usually good only when the user can provide a close estimate of the solution. Therefore it is not suitable for a general purpose algorithm. The following remarks indicate the difficulties that we wish to overcome.

The nonlinear eqs. (2.1) and (2.2) hold not only at constrained minima but also at constrained maxima and constrained saddle points of $F(\boldsymbol{x})$. Therefore the Newton iteration can converge to all types of stationary points and there is no bias towards a minimization calculation. Also many starting points cause the sequence $\{\boldsymbol{x}_k;\, k = 1, 2, 3, ...\}$ that is calculated by the iterative method to cycle or to diverge. The best way of using second derivative values when no constraints are present is still a subject of research in the case when the objective function is not convex. Some sophisticated methods have been suggested recently (by Fletcher [12] and McCormick [18], for example) and similar techniques are likely to be helpful to an algorithm for constrained optimization. Moreover the quadratic programming problem that occurs on each iteration of Wilson's [28] extension of Newton's method may have several solutions when inequality constraints are present and when the matrix $G(\boldsymbol{x}, \boldsymbol{\lambda})$ of expression (2.11) is indefinite. Here we are faced with the difficulties of non-convex quadratic programming.

The methods that are studied in the remainder of this paper were developed mainly because of these disadvantages. They are much more suitable than Newton's method when good initial estimates of $\boldsymbol{x}^*$ and $\boldsymbol{\lambda}^*$ are not available and some of them do not require the calculation of second derivatives.

3. Linear approximations to the constraints

Good algorithms are available for calculating the least value of a general differentiable function $F(\boldsymbol{x})$ when there are linear equality and inequality constraints on the variables. For example the quasi-Newton method described by Buckley [6] is suitable. Therefore an obvious approach when there are nonlinear constraints is to seek the least value of $F(\boldsymbol{x})$ subject to linear approximations to the constraints. Sometimes this technique is successful but often the curvature of the constraints is so important that it cannot be neglected. The following example shows this point rather well.

Let $F(\boldsymbol{x})$ be the function of two variables

$$F(\boldsymbol{x}) = x_1 + x_2 \tag{3.1}$$

and let there be one constraint

$$c(\boldsymbol{x}) = x_1^2 + x_2^2 - 2 = 0. \tag{3.2}$$

The required solution is unique and occurs at the point $(x_1, x_2) = (-1, -1)$. However, if the constraint is replaced by a linear approximation and the objective function is unchanged, then we have the problem of minimizing a linear function of two variables subject to one linear constraint which does not have a proper solution.

The Newton iteration (2.3) can solve this problem successfully. We recall that it takes account of the curvature of the constraints by using the second derivatives of the Lagrangian function

$$F(\boldsymbol{x}) - \sum_i \lambda_i c_i(\boldsymbol{x}), \tag{3.3}$$

instead of the second derivatives of $F(\boldsymbol{x})$. Therefore, in algorithms that minimize a function subject to linear approximations to the constraints, it is much better to let expression (3.3) be the objective function instead of $F(\boldsymbol{x})$. Thus the problem of the previous paragraph can be solved successfully, in general the rate of convergence becomes quadratic instead of linear, and the required solution is always a point of attraction of the process instead of sometimes being a point of repulsion which can happen when $F(\boldsymbol{x})$ alone is the objective function of the linearly constrained minimization calculation.

The following iterative algorithm that uses the objective function (3.3) is proposed by Rosen and Kreuser [24]. At the beginning of each iteration a value of $\boldsymbol{x}$ is given. Linear approximations to the constraints are formed from the values of the constraint functions and their first derivatives at $\boldsymbol{x}$. By using $\nabla F(\boldsymbol{x})$ also values of the parameters λ_i for use in expression (3.3) are obtained. Then the function (3.3) is minimized subject to the linear approximations to the constraints. The calculated value of $\boldsymbol{x}$ is the starting point for the next iteration.

This method has two main advantages over Newton's method to make up for the extra work of each iteration. The first one is that the sequence of values of $\boldsymbol{x}$ is obtained from minimization calculations instead of trying to satisfy just the first order conditions for a constrained stationary point. Therefore we avoid the unwelcome possibility of convergence to a maximum instead of to a minimum point. Also we usually find that the set of vectors $\boldsymbol{x}$ from which the iterations of the Rosen–Kreuser algorithm converge successfully is much larger than the corresponding set for Newton's method. The other main advantage over Newton's method is that there is no need to calculate second derivatives or to introduce approximations to second derivative matrices that may have negative eigenvalues. We have in mind that Buckley's [6] algorithm is suitable for the linearly constrained minimization calculation. Therefore the Rosen–Kreuser method can avoid all the difficulties mentioned at the end of Section 2 that are due to indefinite second derivative matrices.

However there is one disadvantage of the Rosen–Krueser method that can be overcome easily. It is that, unlike Wilson's [28] extension of Newton's method, a quadratic function subject to linear inequality constraints is not always minimized in one iteration. For example, if $F(\boldsymbol{x})$ is the function of one variable $\frac{1}{2}x^2$ and the only constraint is the condition

$$x + 1 \geq 0, \tag{3.4}$$

then the objective function (3.3) is the expression

$$\tfrac{1}{2}x^2 - \lambda(x + 1), \tag{3.5}$$

so the iteration replaces x by $\max[\lambda, -1]$. Therefore the Rosen–Kreuser iteration requires just one iteration only if λ is set to zero initially, but this is done only when the initial value of x is feasible. Therefore sometimes the Rosen–Kreuser method uses two iterations to solve this problem. Since a function is minimized subject to linear constraints on each iteration, it is not unreasonable to expect only one iteration to be needed when all the constraint functions $c_i(\boldsymbol{x})$ are linear.

This feature is obtained by Robinson's [21] method which is very similar to the Rosen–Kreuser algorithm. In each iteration the main difference is that Robinson uses the objective function

$$F(\boldsymbol{x}) - \sum_i \lambda_i\{c_i(\boldsymbol{x}) - u_i(\boldsymbol{x})\}, \tag{3.6}$$

instead of the function (3.3), where $u_i(\boldsymbol{x})$ is the current linear approximation to the function $c_i(\boldsymbol{x})$. Because expression (3.6) is equal to $F(\boldsymbol{x})$ when all the constraints are linear, Robinson's method minimizes a general function subject to linear constraints in only one iteration. When the constraint functions are non-linear we retain the second derivative properties that we require, because the second derivatives of the functions (3.3) and (3.6) are the same. Therefore the use of expression (3.6) is recommended.

Robinson [21] analyses the convergence properties of his algorithm. A good way of understanding some of them is to consider the quadratic programming form of Newton's method and to compare the objective functions (2.11) and (3.6). The notation is confusing because in eq. (2.11) $\boldsymbol{x}$ is the starting point of the iteration and $\boldsymbol{\delta}$ is a displacement from $\boldsymbol{x}$. If we use this notation in expression (3.6) we obtain the objective function

$$Q_R(\boldsymbol{\delta}) = F(\boldsymbol{x} + \boldsymbol{\delta}) - \sum_i \lambda_i\{c_i(\boldsymbol{x} + \boldsymbol{\delta}) - u_i(\boldsymbol{x} + \boldsymbol{\delta})\}. \tag{3.7}$$

Assuming that the vectors $\boldsymbol{\lambda}$ are the same in Robinson's and Newton's method, it is now clear that the first and second derivatives of $Q(\boldsymbol{\delta})$ and $Q_R(\boldsymbol{\delta})$ are equal at $\boldsymbol{\delta} = 0$. Morover the linear constraints on $\boldsymbol{\delta}$ are the same in the two algorithms. It follows that, if the conditions of Section 1 hold, then the difference between the values of $\boldsymbol{\delta}$ obtained by the two methods is of order $\|\boldsymbol{\delta}\|^2$. Therefore Robinson's method also has a second order rate of convergence. In practice the linearly constrained minimization calculation of Robinson's method must be stopped at a finite time, usually when good accuracy seems to be obtained. Therefore some versions of his method become identical to Newton's method in the final stages of convergence.

If we take this last remark further, we may be able to develop a version of Robinson's method that does more work than Newton's method only on the iterations when more work is needed because Newton's method fails to provide a satisfactory value of $\boldsymbol{\delta}$. Is this a good general purpose method for solving constrained minimization problems? Although the ideas of this section have provided some useful improvements to Newton's method, there is still a severe disadvantage to be overcome in order to provide a general purpose algorithm.

The disadvantage is shown clearly when there is one variable and one nonlinear constraint that is satisfied for only one value of x. Then the objective function $F(\boldsymbol{x})$ is immaterial. All the iterative methods that have been described so far reduce to solving the nonlinear equation

$$c(x)=0 \tag{3.8}$$

by Newton's method without any safeguards. It is well known that divergence or oscillatory behaviour may occur. Therefore a general purpose algorithm should include some feature that forces the constraints to be satisfied when linear approximations to the constraints are not adequate. This feature is present in the augmented Lagrangian methods.

4. Augmented Lagrangian methods

At the end of the last section we noted a severe disadvantage of forcing linear approximations to the constraints to be satisfied. Therefore in this section we consider methods that solve an unconstrained minimization problem on each iteration. The simplest approach of this type is to add a penalty term to $F(\boldsymbol{x})$ when $\boldsymbol{x}$ is infeasible. For example, when the constraints are the equalities (2.2), we may apply an algorithm for unconstrained minimization to the function

$$F(\boldsymbol{x})+r\sum_i [c_i(\boldsymbol{x})]^2, \tag{4.1}$$

where r is a large scalar that depends on the iteration number. We let the calculated value of $\boldsymbol{x}$ be $\boldsymbol{x}(r)$. Usually $\boldsymbol{x}(r)$ tends to the required solution if r tends to infinity (see Ryan [25], for instance).

The augmented Lagrangian method can be viewed as an extension of the penalty function idea that avoids the need for r to be very large. To introduce it we ask why $\boldsymbol{x}(r)$ is usually not the required solution to the constrained problem, $\boldsymbol{x}^*$ say. The reason is that the gradient of the function (4.1) is zero at $\boldsymbol{x}(r)$ but the gradient of this function at $\boldsymbol{x}^*$ has the value $\nabla F(\boldsymbol{x}^*)$. Therefore, in the usual case when $\nabla F(\boldsymbol{x}^*)$ is non-zero, the points $\boldsymbol{x}(r)$ and $\boldsymbol{x}^*$ have to be different.

Now the first order conditions (1.2) are satisfied at $\boldsymbol{x}^*$, so it follows that the gradient of the function

$$F(\boldsymbol{x})-\sum_i \lambda_i^* c_i(\boldsymbol{x})+r\sum_i [c_i(\boldsymbol{x})]^2 \tag{4.2}$$

is zero at $\boldsymbol{x}=\boldsymbol{x}^*$ for all values of r. Therefore it seems to be better to replace the function (4.1) by the expression

$$\Phi(\boldsymbol{x},\boldsymbol{\lambda},r)=F(\boldsymbol{x})-\sum_i \lambda_i c_i(\boldsymbol{x})+r\sum_i [c_i(\boldsymbol{x})]^2, \tag{4.3}$$

where $\boldsymbol{\lambda}$ is an estimate of $\boldsymbol{\lambda}^*$. This is the "augmented Lagrangian function". We choose r to be positive.

The augmented Lagrangian method is iterative and each iteration requires values of $\boldsymbol{\lambda}$ and r. An algorithm for unconstrained minimization is applied to the function (4.3). We let $\boldsymbol{x}(\boldsymbol{\lambda},r)$ be the vector of variables that is obtained. Then $\boldsymbol{\lambda}$ and r are adjusted for the next iteration in a way that makes $\boldsymbol{x}(\boldsymbol{\lambda},r)$ converge to $\boldsymbol{x}^*$ as the iterations proceed. This is usually possible because the penalty method $\{\boldsymbol{\lambda}=0, r\to\infty\}$ is included in the scheme. However it is preferable to keep r finite in order to avoid ill-conditioned matrices.

Therefore we wish to know what values of $(\boldsymbol{\lambda},r)$ make $\boldsymbol{x}(\boldsymbol{\lambda},r)$ equal to $\boldsymbol{x}^*$. Since algorithms for unconstrained minimization find local minima, we investigate when $\boldsymbol{x}^*$ is a local minimum of the function (4.3). The first order conditions for an unconstrained minimum state that the equation

$$\nabla F(\boldsymbol{x}^*)-\sum_i \lambda_i \nabla c_i(\boldsymbol{x}^*)=0 \tag{4.4}$$

must hold and the second order conditions require the second derivative matrix

$$G(\boldsymbol{x}^*,\boldsymbol{\lambda})+2rN(\boldsymbol{x}^*)N(\boldsymbol{x}^*)^{\mathrm{T}} \tag{4.5}$$

to be positive definite or positive semi-definite, where $G(\boldsymbol{x},\boldsymbol{\lambda})$ and $N(\boldsymbol{x})$ have been defined already in eqs. (2.4) and (2.5). As in Section 1 we assume that $N(\boldsymbol{x}^*)$ has full rank. It follows from eqs. (1.2) and (4.4) that $\boldsymbol{\lambda}$ must equal $\boldsymbol{\lambda}^*$, and, because of the condition (1.3) when $\boldsymbol{y}$ satisfies expression (1.4), the matrix (4.5) is positive definite if r is greater than a certain constant $\bar{r}$. Sometimes $\bar{r}$ is zero. A more specific result is given by Bertsekas [1] which is that the difference between $\boldsymbol{x}(\boldsymbol{\lambda},r)$ and $\boldsymbol{x}^*$ for $r>\bar{r}$ is of order $\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^*\|/(r-\bar{r})$. Therefore in the augmented Lagrangian method we require to choose the sequence of $(\boldsymbol{\lambda},r)$ values so that the condition

$$\frac{\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^*\|}{(r-\bar{r})}\to 0 \tag{4.6}$$

is satisfied. Since we prefer to keep r finite we wish $\boldsymbol{\lambda}$ to tend to $\boldsymbol{\lambda}^*$. However we have seen already that this is not necessary in the methods that use linear approximations to the constraint functions.

Methods for adjusting $(\boldsymbol{\lambda},r)$ automatically so that $\boldsymbol{\lambda}$ tends to $\boldsymbol{\lambda}^*$ for finite r are described and compared by Fletcher [11]. The adjustment of $\boldsymbol{\lambda}$ when r is large enough can be presented very elegantly as a dual problem in the following way [22]. We keep r fixed and we let $\psi(\boldsymbol{\lambda})$ be the function

$$\psi(\boldsymbol{\lambda}) = \Phi\{\boldsymbol{x}(\boldsymbol{\lambda}, r), \boldsymbol{\lambda}, r\}, \tag{4.7}$$

which is the value of the augmented Lagrangian that is obtained by minimizing over the $\boldsymbol{x}$-variables. It follows that the inequality

$$\psi(\boldsymbol{\lambda}) \le \Phi\{\boldsymbol{x}^*, \boldsymbol{\lambda}, r\} = \Phi\{\boldsymbol{x}^*, \boldsymbol{\lambda}^*, r\} = \psi(\boldsymbol{\lambda}^*) \tag{4.8}$$

is obtained for all $\boldsymbol{\lambda}$, where the second part depends on the fact that the constraint functions are zero at $\boldsymbol{x}^*$. Thus the problem of adjusting $\boldsymbol{\lambda}$ is the problem of maximizing the function $\psi(\boldsymbol{\lambda})$.

This remark is useful because first and second derivatives of $\psi(\boldsymbol{\lambda})$ can be computed quite easily. The first derivative is the vector

$$\nabla\psi(\boldsymbol{\lambda}) = -\boldsymbol{c}\{\boldsymbol{x}(\boldsymbol{\lambda}, r)\} \tag{4.9}$$

and the second derivative matrix is the expression

$$\nabla^2\psi(\boldsymbol{\lambda}) = -N\{\boldsymbol{x}(\boldsymbol{\lambda}, r)\}^{\mathrm{T}}[G\{\boldsymbol{x}(\boldsymbol{\lambda}, r), \boldsymbol{\lambda}, r\}]^{-1}N\{\boldsymbol{x}(\boldsymbol{\lambda}, r)\}, \tag{4.10}$$

where $G(\boldsymbol{x}, \boldsymbol{\lambda}, r)$ is the second derivative matrix with respect to $\boldsymbol{x}$ of the augmented Lagrangian function. Note that, when the unconstrained minimization calculation of the iteration is done by a quasi-Newton method, then an estimate of $G\{\boldsymbol{x}(\boldsymbol{\lambda}, r), \boldsymbol{\lambda}, r\}$ is obtained.

The quantities (4.9) and (4.10) suggest that to maximize $\psi(\boldsymbol{\lambda})$ the value of $\boldsymbol{\lambda}$ should be replaced by the vector

$$\begin{aligned}\boldsymbol{\lambda}^{\dagger} &= \boldsymbol{\lambda} - [\nabla^2\psi(\boldsymbol{\lambda})]^{-1}\nabla\psi(\boldsymbol{\lambda})\\ &= \boldsymbol{\lambda} - [N^{\mathrm{T}}G^{-1}N]^{-1}\boldsymbol{c}.\end{aligned} \tag{4.11}$$

Using this correction formula for $\boldsymbol{\lambda}$ works very well in practice and it provides a quadratic rate of convergence (see [11] for instance). An earlier technique that is often used replaces $\boldsymbol{\lambda}$ by the vector

$$\boldsymbol{\lambda} - \frac{1}{2r}\boldsymbol{c} \tag{4.12}$$

[17], which is a steepest ascent step towards the maximum of the function $\psi(\boldsymbol{\lambda})$. This correction procedure converges to $\boldsymbol{\lambda}^*$ for sufficiently large r at a linear rate which improves as r increases [20]. We note that it does not require the calculation of any derivatives.

There is a close connection between the Newton formula (2.6) and eq. (4.11), which we show because the main purpose of this paper is to learn from the relations between different algorithms. However we cannot make a comparison of the formulae as they stand, because G is the second derivative matrix of the Lagrangian in eq. (2.6) but it is the second derivative matrix of the augmented

Lagrangian in eq. (4.11). We make these matrices the same by replacing $F(\boldsymbol{x})$ in Section 2 by the function (4.1). Therefore we have to replace ∇F in expression (2.6) by the vector $\nabla F + 2rN\boldsymbol{c}$, which gives the formula

$$\boldsymbol{\lambda}^{\dagger} = (N^{\mathrm{T}}G^{-1}N)^{-1}\{N^{\mathrm{T}}G^{-1}(\nabla F + 2rN\boldsymbol{c}) - \boldsymbol{c}\} \tag{4.13}$$

for revising $\boldsymbol{\lambda}$. When $\boldsymbol{x}$ is calculated to minimize expression (4.3) the equation

$$\nabla F - N\boldsymbol{\lambda} + 2rN\boldsymbol{c} = 0 \tag{4.14}$$

is satisfied. Therefore formulae (4.11) and (4.13) are the same. However Tapia [27] points out that eq. (4.13) is preferable in practice, because it does not require the unconstrained minimization calculation to be completed.

The remarks made so far in this section address the case when all the constraints on the variables are equalities. The augmented Lagrangian method has been extended to take account of inequality constraints also. To introduce this work we employ an approach that is due to Rockafellar [23].

It depends on the remark that, if S is a part of the space of the variables that contains the required solution $\boldsymbol{x}^*$, if we define the function

$$\psi_S(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}\in S} \Phi(\boldsymbol{x}, \boldsymbol{\lambda}, r), \tag{4.15}$$

where $\Phi(\boldsymbol{x}, \boldsymbol{\lambda}, r)$ is still expression (4.3), and if there is a vector of parameters $\boldsymbol{\lambda}^*$ such that $\boldsymbol{x}^*$ is the vector $\boldsymbol{x} \in S$ that minimizes $\Phi(\boldsymbol{x}, \boldsymbol{\lambda}^*, r)$, then the method that gives expression (4.8) also provides the inequality

$$\psi_S(\boldsymbol{\lambda}) \le \psi_S(\boldsymbol{\lambda}^*). \tag{4.16}$$

Therefore the following procedure suggests itself. We use any inequality constraints to define S but any equality constraint functions occur in the definition of $\Phi(\boldsymbol{x}, \boldsymbol{\lambda}, r)$. We calculate the right hand side of expression (4.15) for each value of $(\boldsymbol{\lambda}, r)$. Then we adjust $\boldsymbol{\lambda}$ at the end of each iteration in a way that is intended to maximize $\psi_S(\boldsymbol{\lambda})$. However this method is difficult to apply if any of the inequality constraints are nonlinear. Therefore we look at the idea of introducing some slack variables.

If the inequality constraint

$$h_i(\boldsymbol{x}) \ge 0 \tag{4.17}$$

occurs, for example, we let z_i be a new variable and we replace the constraint by the conditions

$$h_i(\boldsymbol{x}) - z_i = 0, \qquad z_i \ge 0. \tag{4.18}$$

If this replacement is made for all the inequality constraints and if the method described at the end of the last paragraph is applied, then we find that, besides the parameters λ_i of the constraint functions $c_i(\boldsymbol{x})$, there are Lagrange parameters, μ_i say, of the constraint functions $h_i(\boldsymbol{x}) - z_i$. Also the space S is just the set of vectors $(\boldsymbol{x}, z)$ for which $z \geq 0$. We have to calculate the least value of the augmented Lagrangian function for $(\boldsymbol{x}, z) \in S$. Because the part of the augmented Lagrangian that depends on z_i is the expression

$$-\mu_i\{h_i(\boldsymbol{x}) - z_i\} + r\{h_i(\boldsymbol{x}) - z_i\}^2, \tag{4.19}$$

it follows that z_i has the value

$$z_i = \max[0, h_i(\boldsymbol{x}) - \mu_i/2r], \tag{4.20}$$

in which case the term (4.19) is the quantity

$$r[h_i(\boldsymbol{x}) - \mu_i/2r]_-^2 - \mu_i^2/4r, \tag{4.21}$$

where the notation $[\cdots]_-$ indicates the minimum of zero and the term inside the square brackets. Thus the slack variable z_i is eliminated from the calculation.

Therefore, when the constraints (2.2) and (2.13) are present, the augmented Lagrangian method is as follows. We let λ_i $(i = 1, 2, \ldots, m)$ and μ_i $(i = 1, 2, \ldots, m')$ be the parameters of the equality and inequality constraints respectively. At the beginning of each iteration a value of $(\boldsymbol{\lambda}, \boldsymbol{\mu}, r)$ is given. The vector of variables $\boldsymbol{x}$ that minimizes the augmented Lagrangian function

$$\begin{aligned}\Phi(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, r) = F(\boldsymbol{x}) &- \sum_i \lambda_i c_i(\boldsymbol{x}) + r\sum_i [c_i(\boldsymbol{x})]^2 \\ &+ r\sum_i [h_i(\boldsymbol{x}) - \mu_i/2r]_-^2 - \sum_i \mu_i^2/4r\end{aligned} \tag{4.22}$$

is calculated. We let $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ be this value of $\boldsymbol{x}$ and we let $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ be the corresponding value of $\Phi(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, r)$. The parameters $(\boldsymbol{\lambda}, \boldsymbol{\mu}, r)$ are adjusted at the end of each iteration so that, as the calculation proceeds, the function $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ converges to its maximum value. Then $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ converges to the required solution of the constrained minimization problem [23]. The dependence of $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ on r is not shown, because usually a satisfactory value of r is obtained near the beginning of the calculation and then r is held constant for the remaining iterations.

The methods for adjusting $(\boldsymbol{\lambda}, \boldsymbol{\mu}, r)$ are similar to the ones that are used to adjust $(\boldsymbol{\lambda}, r)$ when only equality constraints are present [11]. The calculation of first and second derivatives of $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is quite straightforward. Corresponding to eq. (4.9) we find the values

$$\begin{aligned}&\frac{\mathrm{d}\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})}{\mathrm{d}\lambda_i} = -c_i\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})\}, \\ &\frac{\mathrm{d}\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})}{\mathrm{d}\mu_i} = \begin{cases}-h_i\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})\}, & i \in I, \\ -\mu_i/2r, & i \notin I,\end{cases}\end{aligned} \tag{4.23}$$

where I contains the integers i $(1 \leq i \leq m')$ for which the term $[h_i\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})\} - \mu_i/2r]$ is negative. In order to show the second derivatives we let $N(\boldsymbol{x}, \boldsymbol{\mu})$ be the matrix that is obtained by adding the columns $\{\nabla h_i(\boldsymbol{x});\, i \in I\}$ to expression (2.5).

Then the second derivatives of $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ with respect to the parameters λ_i $(i = 1, 2, \dots, m)$ and μ_i $(i \in I)$ are the elements of the matrix

$$-N\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu}), \boldsymbol{\mu}\}^{\mathrm{T}}[G\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu}), \boldsymbol{\lambda}, \boldsymbol{\mu}, r\}]^{-1}N\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu}), \boldsymbol{\mu}\}, \tag{4.24}$$

where $G(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, r)$ is the second derivative matrix with respect to $\boldsymbol{x}$ of the function (4.22). The remaining second derivatives of $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ are zero except for the diagonal elements

$$\frac{\mathrm{d}^2\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})}{\mathrm{d}\mu_i^2} = -\frac{1}{2r}, \quad i \notin I. \tag{4.25}$$

By using these expressions the formula corresponding to eq. (4.11) can be applied to revise $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and it has a quadratic rate of convergence.

Although it is interesting that the adjustment of $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ can be done successfully by seeking the maximum of the function $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ without any constraints on $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, it is sometimes more efficient to make use of the fact that the Kuhn–Tucker conditions state that each component of $\boldsymbol{\mu}$ must be non-negative at the solution. This point is shown by the example of Section 3, where the problem is to minimize the function $\frac{1}{2}x^2$ subject to the constraint (3.4). If $r = \frac{1}{2}$ for example, then the augmented Lagrangian function (4.22) is the expression

$$\tfrac{1}{2}x^2 + \tfrac{1}{2}[x + 1 - \mu]_-^2 - \tfrac{1}{2}\mu^2, \tag{4.26}$$

so $x(\mu)$ and $\psi(\mu)$ have the values

$$\begin{aligned} x(\mu) &= \tfrac{1}{2}[\mu - 1]_+, \\ \psi(\mu) &= \tfrac{1}{4}[\mu - 1]_+^2 - \tfrac{1}{2}\mu^2. \end{aligned} \tag{4.27}$$

It follows that the analogue of eq. (4.11) gives the value

$$\mu^\dagger = \begin{cases} 0, & \mu \le 1, \\ -1, & \mu > 1. \end{cases} \tag{4.28}$$

We will not use the incorrect value $\mu^\dagger = -1$ if we take notice of the Kuhn–Tucker conditions. Therefore, when the first and second derivatives of the previous paragraph give a quadratic approximation to $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$, it is appropriate to calculate $(\boldsymbol{\lambda}^\dagger, \boldsymbol{\mu}^\dagger)$ to maximize the quadratic approximation subject to the condition $\boldsymbol{\mu} \ge 0$, see [11, 14].

Even this technique ignores information that is sometimes useful. We have in mind that a change in $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ induces a change in $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ that can be predicted from the quantities that occur in expressions (4.23) and (4.24). The predicted change in $\boldsymbol{x}$ is helpful in deciding which inequality constraints are important, but it is not used in the methods that have been described already for revising $(\boldsymbol{\lambda}, \boldsymbol{\mu})$. In order to include this information we return to the primal problem, where $\boldsymbol{x}$ is the vector of variables, instead of working with the dual function $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$.

We set up a quadratic programming problem that is similar to the one that occurs on each iteration of Wilson's method [28]. We define the new values of

$(\boldsymbol{\lambda}, \boldsymbol{\mu})$ to be the Lagrange multipliers at the solution of this quadratic programming problem. Its objective function is expression (2.11), except that $G(\boldsymbol{x}, \boldsymbol{\lambda})$ is replaced by $G\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu}), \boldsymbol{\lambda}, \boldsymbol{\mu}, r\}$ and, to take account of the penalty terms of the augmented Lagrangian function, the vector ∇F is replaced by the vector

$$\nabla F(\boldsymbol{x}) + 2r\sum_{i=1}^{m} c_i(\boldsymbol{x})\nabla c_i(\boldsymbol{x}) + 2r\sum_{i\in I} h_i(\boldsymbol{x})\nabla h_i(\boldsymbol{x}), \tag{4.29}$$

calculated at $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$. The constraints on $\boldsymbol{\delta}$ are given in expressions (2.12) and (2.14) where again $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$. Thus new values of $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ are calculated. Besides taking account of the predicted change in $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$, we obtain the extra advantage, that is also present in eq. (4.13), of allowing for inaccuracies in the unconstrained minimization calculation that determines $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$.

By changing only the linear inequality constraints of this quadratic programming problem we can obtain the two methods that were described earlier for revising $(\boldsymbol{\lambda}, \boldsymbol{\mu})$. If the constraints (2.14) are replaced by the equations

$$\nabla h_i(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\delta} + h_i(\boldsymbol{x}) = 0, \quad i \in I, \tag{4.30}$$

where the set I is defined after expression (4.23), then the Lagrange multipliers at the solution of the quadratic programming problem are the values of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ $(i \in I)$ that are obtained by the method mentioned after eq. (4.25). Alternatively, if instead of eq. (4.30) we use the inequalities

$$\nabla h_i(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\delta} + h_i(\boldsymbol{x}) \geq 0, \quad i \in I, \tag{4.31}$$

then the Lagrange multipliers at the solution of this quadratic programming problem are the ones that are obtained by the method given after eq. (4.28) [14]. Both these methods save some work by ignoring the inequality constraints that are not in I and the corresponding values of μ_i are set to zero. Note that, if the conditions of Section 1 are satisfied, then all three methods for revising $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ give identical results when $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is within a certain neighbourhood of $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$. Note also that the solution $\boldsymbol{\delta}$ of the quadratic programming problem that defines $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is relevant to the unconstrained minimization calculation of the iteration that uses the new value of $(\boldsymbol{\lambda}, \boldsymbol{\mu})$. Specifically, if the second derivative matrix of the quadratic programming calculation is used as an approximation to the second derivative matrix of the new augmented Lagrangian function, then $\boldsymbol{\delta}$ is the initial predicted change to $\boldsymbol{x}$ that is needed to solve the unconstrained minimization calculation. It is usual to make this approximation when second derivatives are not calculated.

The remarks at the beginning of the last paragraph suggest that the introduction of the dual function $\psi(\boldsymbol{\lambda}, \boldsymbol{\mu})$ does not help very much. This point of view may be true when the algorithm is described, but the dual function is very helpful to analysing the properties of the augmented Lagrangian method. For instance it provides an excellent way of proving convergence when the inequality constraints that are not in I are ignored by the method that revises $(\boldsymbol{\lambda}, \boldsymbol{\mu})$.

In addition to the development of the augmented Lagrangian method which has been surveyed, much research has been done on related questions that are important in practice. For instance the accuracy that is needed in the unconstrained minimization calculations is studied by Bertsekas [2] and Han [14]. It seems to be appropriate to finish the minimization calculation when the norm of the gradient of the augmented Lagrangian function is less than a multiple of some measure of the infeasibility of $\boldsymbol{x}$. Thus the accuracy of the current Lagrange parameter estimates is allowed for automatically and convergence can be retained even in methods that presume that $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is calculated exactly when they adjust $(\boldsymbol{\lambda}, \boldsymbol{\mu})$.

Another important consideration is the use of quasi-Newton methods for the unconstrained minimization calculations. They build up approximations to second derivatives of the augmented Lagrangian function from the changes in the first derivatives that occur along the search directions that are used in the unconstrained minimization. Thus superlinear convergence is obtained even though there may be large errors in the second derivative estimates along vectors that are not used as search directions (see Dennis and Moré [8], for instance). Because the matrix $G\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu}), \boldsymbol{\lambda}, \boldsymbol{\mu}, r\}$ occurs in the methods for revising $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, the question arises whether it is suitable to use the second derivative approximation from the unconstrained calculation in place of $G\{\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu}), \boldsymbol{\lambda}, \boldsymbol{\mu}, r\}$. Some remarks of Buys [7] suggest that this should not be done, but subsequently Han [14] and Tapia [27] proved the nicest result we could hope for. If the second derivative approximations from the unconstrained calculations are used in the procedures for revising $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ that have been described in this section, then for many quasi-Newton methods the sequence of values of $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ converges to $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ at a superlinear rate.

Procedures for revising r are also very important in practice but, because they may not be relevant to future work, we do not discuss them here. Some good automatic methods are described by Fletcher [11], that allow each constraint function to have its own weighting factor in order to compensate for any differences in scale. Because r is made larger when this is necessary to force convergence from poor starting approximations and because r usually remains at a moderate value, Fletcher provides some excellent general purpose algorithms for constrained minimization calculations. However he points out some disadvantages of the augmented Lagrangian method. The remainder of this section discusses the disadvantages in order to identify the main problems that are addressed by the work of Section 5.

The most obvious disadvantage is the need to minimize the augmented Lagrangian function on each iteration. There is the possibility of wasting effort by seeking a solution that is more accurate than necessary or of accepting an estimate of $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ that is not accurate enough. In some calculations we find that about half of the total number of function and gradient evaluations are made by the first iteration when the value of $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ may be given the default value of zero. In this case the sophistication of the augmented Lagrangian method is not used for much of the time. Moreover, in the final stages of the calculation the Newton method of Section 2 often converges equally well or even better so again the sophistication of the augmented Lagrangian method is not needed. Therefore, since we have to replace the unconstrained minimization calculation by a finite procedure, it may be better to regard the basic augmented Lagrangian method as an ideal algorithm that requires several additions in practice.

Even though the convergence of $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ to $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ is usually achieved quite easily, we question the use of a procedure that relies on accurate values of $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ in order to satisfy the constraints on the variables. It seems to be more sensible and more direct to make use of the values and gradients of the constraint functions in order to satisfy the constraints.

This last remark is particularly relevant when some of the constraints are linear for they are easy to satisfy by direct calculation. However, in general the vectors $\boldsymbol{x}(\boldsymbol{\lambda}, \boldsymbol{\mu})$ obtained by the augmented Lagrangian method do not satisfy the linear constraints. It is sometimes helpful to reduce the size of the calculation by using linear constraints to eliminate variables, but such economies are not present in the basic augmented Lagrangian method.

When inequality constraints occur, the second derivatives with respect to $\boldsymbol{x}$ of the augmented Lagrangian function (4.22) have discontinuities. For example, suppose that the constraint

$$h_i(\boldsymbol{x}) \geq 0 \tag{4.32}$$

was predicted to be inactive when $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ was chosen, but during the unconstrained minimization calculation of the iteration we find that $h_i(\boldsymbol{x})$ becomes negative. In this case μ_i is zero so, at a value of $\boldsymbol{x}$ where $h_i(\boldsymbol{x})$ is zero, the function (4.22) has the second derivative discontinuity

$$2r\nabla h_i(\boldsymbol{x})\nabla h_i(\boldsymbol{x})^{\mathrm{T}}. \tag{4.33}$$

This remark is interesting because it shows that the size of the discontinuity is proportional to r. We have seen that the presence of r is a feature of the augmented Lagrangian method that does not occur in the algorithms described in Sections 2 and 3. However, in practice the second derivative discontinuities can usually be disregarded.

None of these disadvantages are present in a method that is described in the next section, but the proposed method has some difficulties that may be avoided by further research.

5. An extension of Newton's method

We have seen that the methods that do a minimization calculation on each iteration become very similar to Newton's method when the vector of variables approaches the required solution. We have also seen that each minimization calculation must be replaced by a finite procedure. Therefore we take the view in this section that the work of Section 2 provides the basis of a good general algorithm and we use our knowledge of the techniques of Sections 3 and 4 to seek suitable extensions to Newton's method in order to force convergence from poor starting approximations. It is assumed throughout this section that approximations to second derivatives are made, which is an advantage because it is helpful to make the approximation have properties that may not be obtained by the true second derivative matrix. In particular we recall that many of the best algorithms for unconstrained calculations force second derivative approximations to be positive definite.

For the general constrained minimization calculation, where $F(\boldsymbol{x})$ is the objective function and the constraints (2.2) and (2.13) are given, the algorithm that we recommend is due to Han [15, 16] and is similar to Wilson's method. It has the following form. At the beginning of each iteration we have a vector of variables $\boldsymbol{x}$ and a positive definite matrix B, which can be regarded as an approximation to the second derivative matrix of the Lagrangian function

$$F(\boldsymbol{x})-\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{c}(\boldsymbol{x})-\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{x}). \tag{5.1}$$

We calculate the vector $\boldsymbol{\delta}$ that minimizes the quadratic objective function

$$Q(\boldsymbol{\delta})=\boldsymbol{\delta}^{\mathrm{T}}\nabla F(\boldsymbol{x})+\tfrac{1}{2}\boldsymbol{\delta}^{\mathrm{T}}B\boldsymbol{\delta} \tag{5.2}$$

subject to the linear constraints

$$\nabla c_i(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\delta}+c_i(\boldsymbol{x})=0, \quad i=1,2,\dots,m, \tag{5.3}$$

and

$$\nabla h_i(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\delta}+h_i(\boldsymbol{x})\geq 0, \quad i=1,2,\dots,m'. \tag{5.4}$$

We use $\boldsymbol{\delta}$ as a search direction in the space of the variables and therefore let the new value of $\boldsymbol{x}$ be the vector

$$\boldsymbol{x}^{\dagger}=\boldsymbol{x}+\alpha\boldsymbol{\delta}, \tag{5.5}$$

where α is a step-length. We let $(\boldsymbol{\lambda},\boldsymbol{\mu})$ be the Lagrange parameters at the solution of the quadratic programming problem that defines $\boldsymbol{\delta}$. The matrix B is revised by using the difference $\{\boldsymbol{g}(\boldsymbol{x}^{\dagger},\boldsymbol{\lambda},\boldsymbol{\mu})-\boldsymbol{g}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})\}$ in a way that maintains positive definiteness, where $\boldsymbol{g}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{\mu})$ is the gradient with respect to $\boldsymbol{x}$ of the function (5.1). Then a new iteration is begun. There are many open questions in this description that are answered in the remainder of this section. We note that Biggs [4] has proposed a similar algorithm.

As in quasi-Newton methods for unconstrained minimization calculations (see [8], for instance), we expect the algorithm to use the step-length $\alpha=1$ on every iteration in the final stages of the calculation. Therefore Han [16] studies this case. He proves some local convergence theorems that depend on the conditions that $\boldsymbol{x}$ is sufficiently close to $\boldsymbol{x}^*$ and B is sufficiently close to the true second derivative matrix of the Lagrangian function at the solution, G^* say.

However, because there is no need for G^* to be positive definite when constraints are active, we may not be able to satisfy the last condition of Han's theorem. Therefore is it sensible to force B to be positive definite? We seem to have a choice of three strategies. We may do what has been described, or we may change the objective function by adding in a penalty term in order that G^* becomes positive definite, or we may relax the conditions on B. If we add in a penalty term then the difficulties of choosing its size and of second derivative discontinuities due to inequality constraints occur. If we let B have negative eigenvalues then the quadratic programming problem becomes difficult to solve unless we have positive definiteness in the set of values of $\boldsymbol{\delta}$ that satisfy the linear constraints (5.3) and (5.4). It would be awkward to maintain this condition when some negative eigenvalues occur in B, because the directions of the constraint normals change from iteration to iteration. Therefore it is easiest to follow the recommended procedure, keeping B positive definite even when G^* is not.

Convergence theorems have not yet been proved for this procedure, but we are hopeful that the actual local convergence properties are good when $\alpha = 1$ for the following reasons. When $\boldsymbol{x}$ is near the solution then the linear approximations to the constraints will usually cause the constraints to be satisfied quite well by the calculated values of $\boldsymbol{x}$. It follows that the main purpose of the matrix B is to control the choice of variables within the set of feasible points. Within this set the curvature of the augmented Lagrangian function is positive. Therefore the condition that B be positive definite allows the important part of G^* to be approximated accurately, even when G^* has some negative eigenvalues. Further, when $\boldsymbol{x}$ and $\boldsymbol{x}^\dagger$ are close to satisfying the constraints, then the difference in gradients $\{\boldsymbol{g}(\boldsymbol{x}^\dagger, \boldsymbol{\lambda}, \boldsymbol{\mu}) - \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})\}$ usually provides suitable information about G^* that is consistent with a positive definite approximation. A numerical example that is reported later supports these statements.

The method that we recommend for revising the matrix B is based on the BFGS formula for unconstrained minimization calculations. If a change of $\boldsymbol{\delta}$ in the variables is found to give the change

$$\boldsymbol{\gamma} = \boldsymbol{g}(\boldsymbol{x} + \boldsymbol{\delta}) - \boldsymbol{g}(\boldsymbol{x}) \tag{5.6}$$

in the gradient of the objective function and if the condition

$$\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\gamma} > 0 \tag{5.7}$$

is satisfied, then the BFGS formula replaces the second derivative approximation B by the matrix

$$B^\dagger = B - \frac{B\boldsymbol{\delta}\boldsymbol{\delta}^{\mathrm{T}}B}{\boldsymbol{\delta}^{\mathrm{T}}B\boldsymbol{\delta}} + \frac{\boldsymbol{\gamma}\boldsymbol{\gamma}^{\mathrm{T}}}{\boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\gamma}}, \tag{5.8}$$

which maintains positive definiteness [5]. However, in our calculation it may happen that condition (5.7) cannot be satisfied due to the negative curvature of the Lagrangian function. Therefore we replace $\boldsymbol{\gamma}$ in eq. (5.8) by the vector

$$\boldsymbol{\eta} = \theta\boldsymbol{\gamma} + (1-\theta)B\boldsymbol{\delta}, \tag{5.9}$$

where $\boldsymbol{\gamma}$ is still the difference in gradients (5.6) and where θ is the parameter

$$\theta = \begin{cases} 1, & \boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\gamma} \geq 0.2\boldsymbol{\delta}^{\mathrm{T}}B\boldsymbol{\delta}, \\ \dfrac{0.8\boldsymbol{\delta}^{\mathrm{T}}B\boldsymbol{\delta}}{\boldsymbol{\delta}^{\mathrm{T}}B\boldsymbol{\delta} - \boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\gamma}}, & \boldsymbol{\delta}^{\mathrm{T}}\boldsymbol{\gamma} < 0.2\boldsymbol{\delta}^{\mathrm{T}}B\boldsymbol{\delta}. \end{cases} \tag{5.10}$$

Thus positive definiteness is maintained, the value $\theta = 1$ occurs often in practice which provides the usual BFGS formula, and in all other cases a part of the required correction to the second derivative approximation is made in a way that prevents the determinant of $B^{\dagger}$ from being less than 0.2 of the determinant of B. The factor 0.2 is suggested because it seems to be of a suitable size, but it should be noted that little numerical experimentation has been done with the technique described in this paragraph.

An algorithm is now specified except for the choice of α in eq. (5.5). A complication in choosing the step-length is that usually the value of α that is best from the point of view of satisfying the constraints does not provide the least value of $F(\boldsymbol{x})$. Therefore we require a single objective function that obtains a suitable balance between these two aims and which can be used to guide the choice of α. If we find such a function, if it takes its minimum value at the required solution $\boldsymbol{x}^*$ and if the directions $\boldsymbol{\delta}$ are descent directions, then we have a means of forcing convergence from poor starting approximations.

Han [15] proves that the function

$$\Phi(\boldsymbol{x}, r) = F(\boldsymbol{x}) + r\sum_{i=1}^{m} |c_i(\boldsymbol{x})| + r\sum_{i=1}^{m'} |[h_i(\boldsymbol{x})]_-| \tag{5.11}$$

has all of these properties provided that the constant r is sufficiently large. The value of r must not be less than any of the estimates of Lagrange parameters that are given by the quadratic programming problem that defines $\boldsymbol{\delta}$ on each iteration. He proves that, if α is chosen to minimize the function of one variable

$$\phi_0(\alpha) = \Phi(\boldsymbol{x} + \alpha\boldsymbol{\delta}, r) \tag{5.12}$$

on each iteration and if the eigenvalues of the matrices B are bounded above and are bounded away from zero, then either the algorithm terminates at a Kuhn–Tucker point because $\boldsymbol{\delta} = \boldsymbol{0}$ or the points of accumulation of the calculated sequence of values of $\boldsymbol{x}$ are Kuhn–Tucker points. Therefore we seem to have a suitable algorithm if the conditions on B are satisfied.

However I tried some numerical experiments with the objective function (5.11) and occasionally obtained very poor results. They occurred when a large value of r had to be chosen because of the early iterations, but a much smaller value would have been adequate for the remainder of the calculation. In this case the penalty for failing to satisfy a constraint is too severe. Therefore, when the constraints are curved, each iteration makes only a small change to $\boldsymbol{x}$. This inefficiency can be avoided by using a "temporary objective function", which remains fixed during each line search but that can be adjusted during the calculation.

A temporary objective function that occurs naturally is the Lagrangian function (5.1), which depends on the iteration because $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are obtained from the quadratic programming problem that defines $\boldsymbol{\delta}$. Already we have used the gradient of this function in our method for revising B. Now we consider whether it is suitable to choose the value of α to minimize the function

$$\phi_1(\alpha) = F(\boldsymbol{x} + \alpha\boldsymbol{\delta}) - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{c}(\boldsymbol{x} + \alpha\boldsymbol{\delta}) - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{x} + \alpha\boldsymbol{\delta}) \tag{5.13}$$

instead of the function (5.12).

It is suitable only if $\phi_1(\alpha)$ decreases initially when α is increased from zero. We prove that this condition is obtained by using the function

$$\phi_2(\alpha) = Q(\alpha\boldsymbol{\delta}) - \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{u}(\boldsymbol{x} + \alpha\boldsymbol{\delta}) - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{v}(\boldsymbol{x} + \alpha\boldsymbol{\delta}), \tag{5.14}$$

where Q is defined by eq. (5.2) and where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the linear approximations to the constraint functions $\boldsymbol{c}$ and $\boldsymbol{h}$ that are made by the iteration. The function $\phi_2(\alpha)$ is quadratic and it follows from the definition of $\boldsymbol{\delta}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ that it is stationary at $\alpha = 1$. Because the matrix B is positive definite this stationary point is a minimum. Therefore the derivative $\phi_2'(0)$ is negative. Because we can show that $\phi_2'(0)$ and $\phi_1'(0)$ are equal, it follows that $\phi_1(\alpha)$ does decrease when α is small, which is the required result.

Another consideration when minimizing $\phi_1(\alpha)$ is that, although $\phi_2(\alpha)$ is a convex function, there is no need for $\phi_1(\alpha)$ to be convex or even bounded below. Therefore the condition

$$0 < \alpha \leq 1 \tag{5.15}$$

is imposed. We suggest that the value of α be calculated by an approximate line search method to minimize the function (5.13) subject to condition (5.15). The method proposed by Fletcher [9] is suitable. The description of the algorithm that is recommended for constrained minimization calculations is now complete.

This algorithm was applied to the following numerical example. Minimize the function

$$F(\boldsymbol{x}) = \exp(x_1x_2x_3x_4x_5), \tag{5.16}$$

subject to the constraints

$$\begin{aligned} &x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,\\ &x_2x_3 - 5x_4x_5 = 0,\\ &x_1^3 + x_2^3 + 1 = 0, \end{aligned} \tag{5.17}$$

starting at the point $(-2, 2, 2, -1, -1)$ with B set to the unit matrix. Every iteration uses the step-length $\alpha = 1$. The calculated values of $\boldsymbol{x}$ are given in Table

Table 1
A calculation where G^* is positive definite

Iteration	x_1	x_2	x_3	x_4	x_5
0	−2.0000	2.0000	2.0000	−1.0000	−1.0000
1	−1.7742	1.6909	1.7472	−0.7876	−0.7876
2	−1.7609	1.6471	1.7436	−0.7584	−0.7584
3*	−1.7530	1.6370	1.7594	−0.7590	−0.7590
4	−1.7193	1.5984	1.8243	−0.7640	−0.7640
5	−1.7171	1.5957	1.8273	−0.7636	−0.7636
∞	−1.7171	1.5957	1.8272	−0.7636	−0.7636

1. The asterisk denotes that the value of θ in eq. (5.10) is less than one on the third iteration.

The same constrained problem has been solved by Fletcher [11] using several versions of the augmented Lagrangian method. He requires about 36 function and gradient evaluations to achieve the accuracy that is obtained by our method in only 6 function and gradient evaluations. Therefore our algorithm is much faster on some problems.

This example does not test the hypothesis, made earlier in this section, that a superlinear rate of convergence can still be achieved when the second derivative matrix of the Lagrangian function, namely G^*, has negative eigenvalues. Therefore $F(\boldsymbol{x})$ was changed to the function

$$F(\boldsymbol{x}) = \exp(x_1x_2x_3x_4x_5) - \tfrac{1}{2}(x_1^3 + x_2^3 + 1)^2 \tag{5.18}$$

and the algorithm was run again from the same initial conditions. Now the matrix G^* has a negative eigenvalue of about -136 and four positive eigenvalues which are all less than 2. The problem has the same solution as before because the change in $F(\boldsymbol{x})$ is a function of one of the equality constraints. Therefore we are mainly interested in the effect of the large negative eigenvalue on the rate of convergence. The calculated values of $\boldsymbol{x}$ are given in Table 2. Again every iteration chooses a step-length of one. Now there are more asterisks

Table 2
A calculation where G^* has a large negative eigenvalue

Iteration	x_1	x_2	x_3	x_4	x_5
0	−2.0000	2.0000	2.0000	−1.0000	−1.0000
1*	−1.7742	1.6909	1.7472	−0.7876	−0.7876
2*	−1.7635	1.6500	1.7384	−0.7579	−0.7579
3*	−1.7582	1.6430	1.7493	−0.7582	−0.7582
4	−1.7494	1.6329	1.7663	−0.7595	−0.7595
5	−1.7172	1.5959	1.8282	−0.7642	−0.7642
6	−1.7171	1.5956	1.8274	−0.7637	−0.7637
7	−1.7171	1.5957	1.8272	−0.7636	−0.7636

which indicate that $\theta < 1$ occurs more often. It is very promising that the drastic change in the objective function causes only two extra evaluations of the function and gradient.

Another good feature of the proposed algorithm, which has not been mentioned before, is that it is invariant under changes of scale of the constraints. This means that, if we multiply the constraint functions by arbitrary positive constants, then, in exact arithmetic, the calculated sequence of values of $\boldsymbol{x}$ is the same as before, Of course these changes of scale should not make any difference, but it is usually awkward to achieve this nice property when penalty terms are added to the objective function to help satisfy the constraints. Moreover we prefer algorithms to be independent of the scale of the components of $\boldsymbol{x}$, because many calculations involve a variety of units that may have very different magnitudes. Therefore we wish to avoid any dependence on vector and matrix norms and instead we require our scalars to be invariant under linear transformations of the variables. This property is achieved also, except that, in common with most variable metric algorithms for unconstrained optimization, the initial choice of the matrix B may introduce a dependence on scale.

We are also aware of two deficiencies in the recommended algorithm. One is that we have not allowed for the possibility, which can occur due to the nonlinearity of the constraints, that there may be no feasible solution to one of the quadratic programming problems we have to solve, when there is a feasible solution $\boldsymbol{x}^*$ to the main calculation. The other deficiency follows from the remark that, by letting the number of equality constraints be the number of variables, we see that our algorithm ought to be able to solve systems of nonlinear equations in a way that is scale invariant. However, methods of this type that converge reliably from poor starting approximations have not been developed yet. Therefore we expect to encounter some serious difficulties when the rank of the matrix N, defined by eq. (2.5), is less than m.

Because of these remarks and because of the uncertainty of the convergence properties, due mainly to the use of a "temporary objective function", research on the algorithm will continue. However the given discussion and numerical results suggest that the method as it stands will solve many constrained minimization problems more efficiently than other methods.

References

[1] D.P. Bertsekas, "Convergence rate of penalty and multiplier methods", *Proceedings of the 1973 IEEE Conference on Decision and Control* (1973) 260–264.

[2] D.P. Bertsekas, "Combined primal-dual and penalty methods for constrained minimization", *SIAM Journal on Control* 13 (1975) 521–544.

[3] D.P. Bertsekas, "Multiplier methods: a survey", *Automatica* 12 (1976) 133–145.

[4] M.C. Biggs, "Constrained minimization using recursive quadratic programming", in: L.C.W.

Dixon and G.P. Szegö, eds., *Towards global optimization* (North-Holland, Amsterdam, 1975) pp. 341–349.

[5] C.G. Broyden, "The convergence of a class of double-rank minimization algorithms 2. The new algorithm", *Journal of the Institute of Mathematics and its Applications* 6 (1970) 222–231.

[6] A.G. Buckley, "An alternate implementation of Goldfarb's minimization algorithm", *Mathematical Programming* 8 (1975) 207–231.

[7] J.D. Buys, "Dual algorithms for constrained optimization", Ph.D. thesis, University of Leiden (Bronder-Offset, Rotterdam, 1972).

[8] J.E. Dennis and J.J. Moré, "Quasi-Newton methods, motivation and theory", *SIAM Review* 19 (1977) 46–89.

[9] R. Fletcher, "A new approach to variable metric algorithms", *The Computer Journal* 13 (1970) 317–322.

[10] R. Fletcher, "Methods related to Lagrangian functions", in: P.E. Gill and W. Murray, eds., *Numerical methods for constrained optimization* (Academic Press, London, 1974) pp. 219–239.

[11] R. Fletcher, "An ideal penalty function for constrained optimization", *Journal of the Institute of Mathematics and its Applications* 15 (1975) 319–342.

[12] R. Fletcher, "The quest for a natural metric", presented at the ninth international symposium on mathematical programming, (Budapest, 1976).

[13] U.M. Garcia-Palomares and O.L. Mangasarian, "Superlinearly convergent quasi-Newton algorithms for nonlinearly constrained optimization problems", *Mathematical Programming* 11 (1976) 1–13.

[14] S-P. Han, "Penalty Lagrangian methods in a quasi-Newton approach", Report TR 75-252, Computer Science, Cornell University (Ithaca, 1975).

[15] S-P. Han, "A globally convergent method for nonlinear programming", Report TR 75-257, Computer Science, Cornell University (Ithaca, 1975).

[16] S-P. Han, "Superlinearly convergent variable metric algorithms for general nonlinear programming problems", *Mathematical Programming* 11 (1976) 263–282.

[17] M.R. Hestenes, "Multiplier and gradient methods", *Journal of Optimization Theory and its Applications* 4 (1969) 303–320.

[18] G.P. McCormick, "Second order convergence using a modified Armijo step-size rule for function minimization", presented at the ninth international symposium on mathematical programming (Budapest, 1976).

[19] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables* (Academic Press, New York, 1970).

[20] M.J.D. Powell, "A method for nonlinear constraints in minimization problems", in: R. Fletcher, ed., *Optimization* (Academic Press, London, 1969) pp. 283–298.

[21] S.M. Robinson, "A quadratically convergent algorithm for general nonlinear programming problems", *Mathematical Programming* 3 (1972) 145–156.

[22] R.T. Rockafellar, "New applications of duality in convex programming", presented at the seventh international symposium on mathematical programming (The Hague, 1970).

[23] R.T. Rockafellar, "A dual approach to solving nonlinear programming problems by unconstrained optimization", *Mathematical Programming* 5 (1973) 354–373.

[24] J.B. Rosen and J. Kreuser, "A gradient projection algorithm for nonlinear constraints", in: F.A. Lootsma, ed., *Numerical methods for nonlinear optimization* (Academic Press, London, 1972) pp. 297–300.

[25] D.M. Ryan, "Penalty and barrier functions", in: P.E. Gill and W. Murray, eds., *Numerical methods for constrained optimization* (Academic Press, London, 1974) pp. 175–190.

[26] R.W.H. Sargent, "Reduced-gradient and projection methods for nonlinear programming", in: P.E. Gill and W. Murray, eds., *Numerical methods for constrained optimization* (Academic Press, London, 1974) pp. 149–174.

[27] R.A. Tapia, "Diagonalized multiplier methods and quasi-Newton methods for constrained optimization", manuscript (Rice University, Houston, 1976).

[28] R.B. Wilson, "A simplical method for convex programming", Ph.D. thesis, Harvard University (Cambridge, MA, 1963).