MULTIPLIER–CONTINUATION ALGORITHMS FOR CONSTRAINED OPTIMIZATION

Bruce N. Lundberg*
Aubrey B. Poore** and Bing Yang**
Department of Mathematics
Colorado State University
Fort Collins, Colorado 80523

ABSTRACT

Several path following algorithms based on the combination of three smooth penalty functions, the quadratic penalty for equality constraints and the quadratic loss and log barrier for inequality constraints, their modern counterparts, augmented Lagrangian or multiplier methods, sequential quadratic programming, and predictor–corrector continuation are described. In the first phase of this methodology, one minimizes the unconstrained or linearly constrained penalty function or augmented Lagrangian. A homotopy path generated from the functions is then followed to optimality using efficient predictor–corrector continuation methods. The continuation steps are asymptotic to those taken by sequential quadratic programming which can be used in the final steps. Numerical test results show the method to be efficient, robust, and a competitive alternative to sequential quadratic programming.

*The work of the first author was partially supported by the National Aeronautics and Space Administration through NASA Grant # NGT–06–002–802. **The second and third authors were supported in part by the Air Force Office of Scientific Research through Grant # AFOSR–88–0059.

1. Introduction

Path following algorithms for the solution of constrained optimization have been revitalized in recent years, due in no small part to the claims and success of the Karmarkar algorithm in linear programming. The ones presented here can be viewed as a combination of various elements and techniques in nonlinear programming: three smooth penalty functions (quadratic penalty for equality constraints, quadratic loss and log barrier for inequality constraints), their more modern counterparts, augmented Lagrangian or multiplier methods, sequential quadratic programming (Newton's method), and predictor–corrector continuation methods for efficient path following. The objective in this work then is to describe this class of algorithms and to present numerical evidence of their efficiency, robustness, and potential.

One view (ref. 1) of these algorithms starts with the three aforementioned smooth penalty functions. One first performs an unconstrained or linearly constrained optimization of the penalty function. The minimizer then satisfies a set of first order necessary conditions (the gradient of the penalty function is zero when all constraints are incorporated into the penalty function) from which one can define an equivalent system of parameterized nonlinear equations. This system represents a perturbation of the Karush–Kuhn–Tucker first order necessary conditions, and the solution is followed to optimality using efficient predictor–corrector continuation methods. The simplest predictor–corrector steps are asymptotic to ' those taken in sequential quadratic programming, and thus the local convergence rates are the same as those of sequential quadratic programming. When shifts and weights are added to these three penalty functions and are adaptively chosen or updated during the optimization phase, one has the class of multiplier or augmented Lagrangian methods. Theoretically, one can expect a shorter path through the use of augmented Lagrangians (ref. 2, Theorem 12.2.1), which suggests that the use of these updates in the weights, scales, and shifts may be used to generate good paths to optimality.

A different perspective of these algorithms evolves from sequential quadratic programming (SQP) itself. These SQP methods perform exceptionally well in minimizing function evaluations, but may be slow since the combinatorial complexity of the inequality constraints is reflected in the subproblems at each step. Furthermore, they are currently restricted to small to medium size problems with promise for large scale applications (refs. 2 and 3). Augmented Lagrangian methods, on the other hand, are currently used effectively for large scale problems with structured sparsity arising, for example, from discretized partial differential equations. Also, augmented Lagrangians are often used as merit functions for globalizing sequential quadratic programming. However, the minimizer of the augmented Lagrangian at any given stage is not a solution of the original problem, and thus the homotopy between the minimizer of the augmented Lagrangian and sequential quadratic programming may be viewed as an intermediate globalization technique. We find this to be very efficient.

In the sections to follow, we briefly outline the methodology and present in section seven the results of our preliminary numerical testing.

2. Background and General Results

For expendiency and convenience of presentation, we present in this section the first order necesssary and second order sufficient conditions for the mathematical programming problem

$$\text{(2.1)} \qquad \text{Min } \{ f(x) \mid h(x) = 0,\ g(x) \geq 0\}$$

where f: $\mathbb{R}^n \to \mathbb{R}^1$, h: $\mathbb{R}^n \to \mathbb{R}^q$ and g: $\mathbb{R}^n \to \mathbb{R}^p$ are assumed to be twice continuously differentiable in an open set Ω containing the feasible region $\mathcal{R} = \{ x \mid h(x) = 0, g(x) \geq 0 \}$. The Karush–Kuhn–Tucker first order necessary conditions state that in the presence of a constraint qualification, there exist multiplier vectors λ and μ such that

$$\text{(2.2)} \qquad F(x,\lambda,\mu) = \begin{bmatrix} \nabla\mathcal{L}(x,\lambda,\mu) \\ h(x) \\ Mg(x) \end{bmatrix} = 0$$
$$\mu \geq 0,\ g(x) \geq 0,$$

where $\mathcal{L} = f - h^T\lambda - g^T\mu$, $M = \text{diag}(\mu)$, $\mu = (\mu_1,\dots,\mu_p)^T$, $\lambda = (\lambda_1,\dots,\lambda_q)$.

THEOREM 2.1 (ref. 4). Let (x_o,λ_o,μ_o) be a solution of $F(x,\lambda,\mu) = 0$. Assume f, g and h are twice continuously differentiable in a neighborhood of x_o and define two index sets $\bar{A}$ and A and a corresponding tangent space $\bar{T}$ by

$$\bar{A} = \{i\colon 1 \leq i \leq p,\ g_i(x_o) = 0\}, \qquad A = \{i \in \bar{A}\colon \mu_i^o \neq 0\},$$

$$\bar{T} = \{y \in \mathbb{R}^n\colon D_x h(x_o)y = 0,\ \ D_x g_i(x_o)y = 0 \ \ (i \in \bar{A})\}.$$

Then a necessary and sufficient condition that $D_{(x,\lambda,\mu)}F(z_o,\lambda_o,\mu_o)$ be nonsingular is that each of the following three conditions hold:

(a) $\bar{A} = A$;

(b) $\{\{\nabla_x g_i(x_o)\}_{i\in\bar{A}} \cup \{\nabla_x h_j(x_o)\}_{j=1}^{q}\}$ *is a linearly independent collection of* $q + |\bar{A}|$ *vectors where* $|\bar{A}|$ *denotes the cardinality of* $\bar{A}$;

(c) *the Hessian of the Lagrangian* $\nabla_x^2\mathcal{L}$ *is nonsingular on the tangent space* $\bar{T}$ *at* (x_o,λ_o,μ_o).

Furthermore, if (b) remains valid, $\mu_o \geq 0$, $g(x_o) \geq 0$, *and (a) and (c) are replaced by*

(a') $\mu_i^o > 0$ *for all* $i \in \bar{A}$,

(c') the Hessian of the Lagrangian $\nabla_x^2\mathcal{L}$ *is positive definite on the tangent space* $\bar{T}$ *at* (x_0,λ_0,μ_0),

then $D_{(x,\lambda,\mu)}F(z_0,\lambda_0,\mu_0)$ *is nonsingular and* x_0 *is a local minimum of the nonlinear programming problem (2.1).*

We note that condition (a) is a complementarity condition, (b) is the linear independence constraint qualification, and (c) is a second-order condition.

3. Newton's Method for Nonlinear Programming

The objective of this section is to briefly review Newton's method (Newton–Lagrange or sequential quadratic programming) for the general nonlinear programming problem (2.1). The methods which we later present are asymptotic to Newton's method and, in fact, may be used in the last few iterations. The starting point for Newton's method is the first order necessary condition as stated in equation (2.2).

The motivation for Newton's method is to form a linear model suggested by Taylor's formula about the current point (x,λ,μ) and then solve this model for the linear increments $(\Delta x,\Delta\lambda,\Delta\mu)$. The difference between a nonlinear equations approach and that of constrained optimization is that one keeps a quadratic term in the complementarity condition:

$$(3.1)\quad F(x+\Delta x,\lambda+\Delta\lambda,\mu+\Delta\mu) = \begin{bmatrix} \nabla^2 L\Delta x - D_x h^T\Delta x - D_x g^T\Delta x + \nabla L \\ D_x h\Delta x + h \\ (M + \Delta M)(D_x g\Delta x + g) \end{bmatrix} + O(\Delta^2)$$

where $O(\Delta^2)$ refers to those increments $(\Delta x,\Delta\lambda,\Delta\mu)$ which are second order and higher. Note that the quadratic increment $\Delta M D_x g\Delta x$ has been retained, but the addition of such higher order terms will not affect the quadratic convergence of Newton's method. Setting this bracketed quantity to zero yields

$$\begin{aligned} &\nabla^2 L\Delta x - D_x h^T\Delta x - D_x g^T\Delta x = -\nabla L \\ &D_x h\Delta x + h = 0 \\ &(M + \Delta M)(D_x g\Delta x + g) = 0, \end{aligned}$$

which, if $\mu + \Delta\mu \geq 0$ is imposed, represent the first order optimality conditions for the quadratic programming subproblem

$$(3.2)\qquad \begin{aligned} \text{MIN}\quad & f + \nabla f^T\Delta x + (1/2)\Delta x^T\nabla^2 L\Delta x \\ \text{ST}\quad & D_x h\Delta x + h = 0 \\ & D_x g\Delta x + g \geq 0. \end{aligned}$$

Second order information for both the constraints and the objective function are built into the quadratic programming subproblem through the appearence of $\nabla^2 L$. The generic Newton's method with no safeguards is as follows:

ALGORITHM [NEWTON'S METHOD FOR NONLINEAR PROGRAMMING]

```
Initialize $x = x_o$, $\lambda = \lambda_o$, and $\mu = \mu_o$
For k = 0,1,..., until satisfied, do (a), (b), (c)
    (a) Compute $f = f(x_k)$, $h = h(x_k)$, $g = g(x_k)$, $\nabla f = \nabla f(x_k)$,
            $D_x h = D_x h(x_k)$, $D_x g = D_x g(x_k)$, $\nabla^2 L = \nabla^2 L(x_k, \lambda_k, \mu_k)$,
    (b) Solve (3.2) for the correction $\Delta x$ in $x_k$ and the multipliers
            $\mu_k + \Delta\mu \geq 0$ and $\lambda_k + \Delta\lambda$,
    (c) Update: $x_{k+1} = x_k + \Delta x$, $\lambda_{k+1} = \lambda_k + \Delta\lambda$, $\mu_{k+1} = \mu_k + \Delta\mu$.
```

The quadratic convergence of Newton's method for nonlinear equations is preserved for the nonlinear programming problem under conditions similar to those in nonlinear equations:

Theorem 3.1 [Convergence of Newton's Method]. *Let $\hat{x}$ be a local solution of the nonlinear programming problem, assume that f, g, and h are C^2 functions whose second derivatives are Lipschitz continuous in a neighborhood of $\hat{x}$, and suppose that the linear independence constraint qualification (condition (b) in Theorem 2.1) is satisfied. Then there exist multipliers $\hat{\lambda}$ and $\hat{\mu} \geq 0$ for which the Karush–Kuhn–Tucker conditions (2.2) are valid. If conditions (a') and (b') in Theorem 2.1 are satisfied, then there is a neighborhood N of $(\hat{x}, \hat{\lambda}, \hat{\mu})$ such that if (x_o, λ_o, μ_o) is in N, then the iterates (x_k, λ_k, μ_k) are defined, remain in N and converge quadratically to $(\hat{x}, \hat{\lambda}, \hat{\mu})$.*

As with Newton's method for nonlinear equations, there are the questions of linear algebra, updating techniques for quasi–Newton steps, and globalization methods. These questions are discussed, for example, in the review paper of Stoer (ref. 3), the book of Fletcher (ref. 2), and the references therein. Two commonly used merit functions for globalization are the L_1 penalty function

$$P = \nu f + (1/r)\ \Sigma\ |h_i(x)| + (1/r)\ \Sigma\ g_j(x)^-$$

and the augmented Lagrangian

$$L_a(x,\lambda,\mu,r) = f + (1/2r)\ \Sigma\ (h_i(x) - r\hat{\lambda}_i)^2 + (1/2r)\ \Sigma\ [(g_j(x) - r\hat{\mu}_j)^-\ ]^2$$

where $g_j^-(x) = \text{Min}\ \{ g_j(x), 0 \}$, r is the penalty parameter, and $\hat{\lambda}$ and $\hat{\mu}$ are the shifts or approximate multipliers. The use of the L_1 penalty function as a merit function for sequential quadratic programming has proved to be highly successful for small to medium problems. Augmented Lagrangians on the other hand have as their domain of application large scale problems such as those that appear in discretized partial differential equations; however, the function evaluation count on small to medium problems is not nearly as favorable as that of sequential quadratic programming (ref. 5). On the other hand, sequential quadratic programming may be slow in comparison to augmented Lagrangian methods, primarily due to the combinatorial complexity of the inequality constraints which persists in the quadratic subprograms. In this work we combine both approaches by using a few steps of augmented Lagrangians followed by a homotopy phase, and then sequential quadratic programming.

4. Homotopy Methods for Constrained Optimization

The idea of a homotopy method is to embed a difficult problem into a parameterized set of problems such that at one parameter value the problem is "easy" to solve and at another value one recovers the "difficult" problem. One then continues the solution of this parameterized system from the easy problem to the desired one. For nonlinear equations $F(x) = 0$, two commonly used homotopies are $G(x,t) = tF(x) + (1-t)(F(x) - F(a))$ (the global Newton homotopy) and $G(t,x) = tF(x) + (1-t)(x-a)$ (the regularizing homotopy). Indeed, at $t = 0$ these are easy to solve, while at $t = 1$ one recovers the original problem. These homotopy methods tend to be quite robust, but currently are not as efficient as the use of merit functions with a modified or quasi-Newton method. The homotopy methods discussed here are generally very efficient, but the the easy problem is not as easy as the above ones for nonlinear equations in that the "easy" problem requires the solution of an unconstrained or linearly constrained optimization problem.

To illustrate the idea and for later numerical comparisons, we first consider the mixed quadratic penalty-log barrier function

$$P(x,r) = f + (1/2r)\ \Sigma\ h_i^2(x) - r\ \Sigma\ \ln(g_j(x)) \tag{4.1}$$

or in the more general form

$$\begin{aligned} P(x,\nu,\hat{\lambda},\mu,\sigma,\omega,\delta,r) = \nu f &+ (1/2r)\ \Sigma\ \sigma_i(\gamma_i h_i(x) - r\sigma_i^{-1}\hat{\lambda}_i)^2 \\ &- r\ \Sigma\ \omega_j\ \ln(g_j(x) + r\delta_j) \end{aligned} \tag{4.2}$$

wherein weights σ_i and ω_j, scales ν and γ_i, and shifts $\hat{\lambda}_i$ and δ_i have been introduced. The homotopy generated from this penalty function depends on the system parameters, which can be adaptively chosen during the optimization phase.

To explain how one can derive a homotopy, we consider the simpler form (4.1). At $r = r_0$, a minimizer, x_0, of this penalty function satisfies

$$\nabla P = \nabla f + \Sigma\, \nabla h_i (h_i/r) - \Sigma\, (r/g_j) \nabla g_j = 0,$$

which along with the definitions $\lambda_i = -h_i/r$ and $\mu_j = r/g_j$ yield an equivalent system of parameterized nonlinear equations

$$\begin{aligned} &\nabla L = 0, && L = f - h^T\lambda - g^T\mu \\ &h + r\lambda = 0, && \\ &Mg - re = 0, && M = \mathrm{diag}(\mu),\ e = (1,\ldots,1)^T. \end{aligned} \tag{4.3}$$

A solution to this parameterized system at $r = r_0$ is given by $\lambda = -h(x_0)/r$ and $\mu = r/g(x_0)$, componentwise. Furthermore, these equations represent the first order necessary conditions at $r = 0$ since $\mu(r) > 0$ for $r > 0$. Once the optimization phase is complete, continuation techniques can be used to track the solution to optimality at $r = 0$. Further discussion of this homotopy can be found in the work of Poore and Al–Hassan (ref. 1).

Another homotopy can be based on the quadratic penalty–loss function

$$P(x,r) = f + (1/2r)\, \Sigma\, h_i^2(x) + (1/2r)\, \Sigma\, [g_j^-(x)]^2 \tag{4.4}$$

or more generally, the augmented Lagrangian function,

$$\begin{aligned} L_a(x,\nu,\hat{\lambda},\hat{\mu},\sigma,\phi,\delta,\beta,r) = \nu f &+ (1/2r)\, \Sigma\, \sigma_i(\gamma_i h_i(x) - r\sigma_i^{-1}\hat{\lambda}_i)^2 \\ &+ (1/2r)\, \Sigma\, \phi_j[(\beta_j g_j(x) - r\phi_j^{-1}\hat{\mu}_j)^-\,]^2 \end{aligned} \tag{4.5}$$

where $g_j^-(x) = \mathrm{Min}\ \{\ g_j(x),\ 0\ \}$, weights ν, σ_i and ϕ_j, shifts $\hat{\lambda}_i$ and $\hat{\mu}_j$, and scales γ_i and β_j have been introduced. These parameters can again be adaptively chosen during the optimization phase. At a minimizer of the augmented Lagrangian L_a

$$\begin{aligned} \nabla L_a = \nu\nabla f &+ (1/r)\, \Sigma\, \nabla(\gamma_i h_i)(\sigma_i\gamma_i h_i(x) - r\hat{\lambda}_i) \\ &+ (1/r)\, \Sigma\, \nabla(\beta_j g_j)[(\phi_j\beta_j g_j(x) - r\hat{\mu}_j)^-\,] = 0. \end{aligned} \tag{4.6}$$

The definitions $\lambda = -(S\Gamma h - r\hat{\lambda})/r$ and $\mu = -(\phi Bg - r\hat{\mu})^-/r$ along with this equation yield the equivalent system

$$(4.7)\qquad \begin{array}{ll} \nabla L = 0, & L = \nu f - (\Gamma h)^T\lambda - (Bg)^T\mu \\ \Gamma h + S^{-1}r(\lambda - \hat{\lambda}) = 0 & S = \mathrm{diag}(\sigma) \\ \beta_j g_j(x) + \phi_j^{-1}r(\mu_j - \hat{\mu}_j) = 0 & \text{if } \beta_j g_j(x) < \phi_j^{-1}r\hat{\mu}_j \\ \mu_j = 0 & \text{if } \beta_j g_j(x) \geq \phi_j^{-1}r\hat{\mu}_j \end{array}$$

where $S = \mathrm{diag}\ (\sigma)$ and $\Phi = \mathrm{diag}\ (\phi)$ represent the weights and $\Gamma = \mathrm{diag}\ (\gamma)$ and $B = \mathrm{diag}\ (\beta)$ the scales. The usual updates for the multipliers $\hat{\lambda}$ and $\hat{\mu}$ (refs. 2 and 3) can be used in the optimization phase. Note that the use of different scales, weights, and updated multiplier approximations all change the homotopy path, and thus may be used to generate "good" paths. Furthermore, the ill-conditioning present in the penalty method is no longer present in the homotopies generated from these penalty functions. A final important modification to these homotopies is the normalization of the mulitpliers (ref. 1) to prevent multipliers tending to infinity, which happens generically when the linear independence constraint qualification is violated.

5. Continuation Methods

The system of parameterized equations posed in the previous section can be written as $G(z,r) = 0$ where r is the homotopy parameter and is arranged so that it goes from $r_0 > 0$ to 0. The primary objective of this section is to briefly describe the methodology of traversing the path from $r_0 = r_0$ to optimality at $r = 0$. The idea is to generate a sequence of points $\{(z_i,r_i)\}_{i=o}^{n}$ with $r_0 = r_0$ and $r_n = 0$. To get from (z_i,r_i) to (z_{i+1},r_{i+1}), one first predicts a new point (z_{i+1}^p,r_{i+1}^p) near the curve and then corrects back to the curve to obtain the desired (z_{i+1},r_{i+1}). Prediction is based on extrapolation of current and previous information about the solution. The extrapolation via polynomial interpolation of the solution values has been used for some time, but extrapolation of the tangents to the curve as is used here appears to be numerically more robust and efficient. A brief explanation of this methodology is given in the remainder of this section, but a more comprehensive explanation can be found in the works of Keller (ref. 6) and Shampine and Gordon (ref. 7).

A formal differentiation of $G(z,r) = 0$ with respect to a third variable s yields Davidenko differential equation

$$D_zG\frac{dz}{ds} + D_r\frac{dr}{ds} = 0,$$

where s can be chosen to be arclength by adding the normalization

$$\left\|\frac{dz}{ds}\right\|^2 + \left|\frac{dr}{ds}\right|^2 - 1 = 0.$$

Once an orientation, ie $\frac{dr}{ds}$ is positive or negative, is known and if D_zG is nonsingular, then one can write this system as the differential equation

$$\frac{dw}{ds} = f(w).$$

Thus given a point (w_k,s_k) on the curve, the diffential equation can be integrated to obtain

$$w(s_k + \Delta s) = w_k + \int_{s_k}^{s_k + \Delta s} f(w)\, ds$$

where f(w) denotes the tangent to the curve at the point w(s). If a polynomial $P_{k,m}$ of degree at most m is used to interpolate f at (w_{k-j},s_{k-j}) for j = o,...,m, the predicted solution is taken to be

$$w_{k+1}^p = w_k + \Delta s\, D(\Delta s), \quad D(\Delta s) = \frac{1}{\Delta s} \int_{s_k}^{s_k + \Delta s} P_{k,m}(s)\, ds.$$

Given an error tolerance, once can vary the order of this formula to achieve the largest stepsize, Δs, possible. This method varies from the standard Adams–Bashforth technique in that the stepsize and order are varied at each step.

Once a predicted point is obtained the correction back to the curve can be obtained in several ways. Two popular ones are the vertical correction, wherein the system of equations G(z,r) = 0 is solved with r fixed at the predicted value, and the correction in a hyperplane orthogonal to the predictor direction. In this latter method one solves the augmented system of equations

$$\begin{aligned} G(w) &= 0 \\ N(w) &= 0 \end{aligned}$$

where $N(w) = D^T(w - w_{k+1}^P)$ represents the plane orthogonal to the predictor direction and passing the predicted point w_{k+1}^p and w = (z,r).

6. Relation to Sequential Quadratic Programming

The predictor–corrector method of the previous section gives steps toward the optimal solution and is, in fact, asymptotic to those obtained by sequential quadratic programming. The continuation phase may thus be viewed as a method for globalizing Newton's method. To explain the connection between the continuation phase with these two homotopies and sequential quadratic programming, we confine our attention in this section to the the homotopy generated by the quadratic penalty–log barrier function, ie.

$$\begin{array}{ll} \nabla L = 0, & L = \nu f - h^T\lambda - g^T\mu \\ h + r\lambda = 0 & \\ Mg - re = 0 & M = \mathrm{diag}(\mu). \end{array}$$

The result is that a Newton step as defined from the solution of (3.2) is asymptotic to a vertical correction plus an Euler prediction as r tends to 0. More precisely, if $\Delta = (\Delta x, \Delta\lambda, \Delta\mu)$ denotes the Newton step as defined by the quadratic programming subproblems (3.2), $\Delta_1 = (\Delta x_1, \Delta\lambda_1, \Delta\mu_1)$, the Euler predictor or tangent to the curve with $\Delta r = -r$, and $\Delta_2 = (\Delta x_2, \Delta\lambda_2, \Delta\mu_2)$ the vertical correction with $\Delta r = 0$, then

$$\Delta = \Delta_1 + \Delta_2 + O(r\|\Delta\lambda_2\|, \|\Delta\|^2) \text{ as } r \to 0 \text{ and } \Delta \to 0.$$

A similar result applies for the quadratic penalty–loss function with the slight modification

$$\Delta = \Delta_1 + \Delta_2 + O(r\|\Delta\lambda_2,\Delta\mu_2\|) \text{ as } r \to 0.$$

This suggests that as soon as the predicted value reaches $r = 0$, one could just as effectively switch to sequential quadratic programming without a globalizer.

7. Numerical Examples

In this section we consider two approaches to constrained optimization based on the quadratic penalty–log barrier function and the augmented Lagrangian or shifted quadratic penalty–quadratic loss function. For the former we have previously compared the numerics and briefly summarize some of their properties (ref. 1). For the augmented Lagrangian approach we present some of our recent testing on some nontrivial test problems (ref. 5) to demonstrate the robustness, efficiency, and potential for the methodology.

For the quadratic penalty–log barrier function we first use the loss function $(g^-)^T g^-$ to generate a point $\hat{x}$ at which $g(\hat{x}) > 0$ or is at least close to feasible region $\{ x : g(x) \geq 0 \}$ and then define a δ so that $g(\hat{x}) + \delta > 0$. Then we use a quasi–Newton with a BFGS update to minimize the penalty function P function $P(x,r) = f(x) + h^T(x)h(x)/(2r) - r\Sigma \ln(g_i(x) + r\delta_i/r^0)$ at some value of the penalty parameter, say r_0, at which the problem is reasonably well conditioned. A quadratic–cubic line search and a Armijo stopping criterion (ref. 8), modified to maintain feasibility $(g(x) + \delta > 0)$ has been used to globalize the quasi–Newton method. Once the minimization problem is solved, continuation techniques are used to track the solution to optimality at $r = 0$. The initial value of $r_0 = .1$ has been used in the numerical experiments reported in the table below under the heading PENCON, but scaling has not been used. Additional information can be found in (ref. 1).

For the quadratic penalty–loss function, scaling and adaptive choices of the weights and scales have been used. For this penalty function, one does not need an initial feasible point for the inequality constraints. Again the BFGS update has been used, but the line search has been modeled after that of Fletcher (ref. 2).

To get some estimation of the relative performance of this algorithm, we have solved several test problems from the book by W. Hock and K. Schittkowski (ref. 5) and give a comparative summary of the number of function evaluations in the table below. The function evaluation for the codes other than PENCON and LOSSCON are taken from (ref. 5). Consistent with those function evaluation counts, we count the evaluation of a p dimensional vector as p function evaluations; however, we do not count upper and lower bounds on variable since they are handled directly in the code and gradient evaluations of linear constraints are counted only once. The approximation of the Hessian of the Lagrangian in the continuation phase is based on finite differences (ref. 8).

CODE	AUTHOR	METHOD
VF02AD	Powell	Quadratic Approximation
OPRQP	Bartholomew–Biggs	Quadratic Approximation
GRGA	Abadie	Generalized Reduced Gradient
VF01A	Fletcher	Multiplier
FUNMIN	Kraft	Multiplier
FMIN	Kraft, Lootsma	Penalty
PENCON	Al–Hassan, Poore	Penalty–Continuation
LOSSCON	Lundberg, Poore, Yang	Loss Function–Continuation

CODE:	VF02AD	OPRQP	GRGA	VF01A	FUNMIN	FMIN	PENCON	LOSSCON
PROB. NO.								
5	16	16	86	32	38	200	26	16
10	48	126	678	280	554	687	88	63
12	48	132	277	300	492	306	130	68
13	180	300	192	565	928	4,178	269	209
14	36	126	108	192	726	838	107	91
15	30	165	508	377	496	2,464		113
19	78	1,785	314	838	3,339	661	423	304
20	160	200	102	383	728	4,094	282	104
29	52	206	646	421	482	414	155	162
118	**	1,800	3,857	**	**	**	2,804	

FUNCTION EVALUATION COUNT **indicates failure

Comparisons from this table illustrate that these penalty–continuation algorithms currently come in second and occasionally first and third; however, the methodology is quite robust, primarily because of the robustness of penalty paths leading to optimality and the robustness of the continuation methodology. The answers given by PENCON and LOSSCON have approximately twelve digits of accuracy on a 14+ digit CDC machine, whereas the remaining answers listed above are computed on a

10 digit machine (ref. 5) and those for VF02AD are generally to low accuracy. Furthermore, the quadratic penalty–loss function tends to perform a little better than the quadratic penalty – log barrier function with the current implementations. Scaling yields a significant improvement in some problems by reducing the number of function evaluations required in the unconstrained optimization phase and can reduce the number of steps taken in the continuation phase. The use of multiplier updates can also significantly reduce the length of the homotopy path but currently requires more function evaluations due to the reoptimization required after an update. These updates, however, may be useful for the large scale problems.

8. Summary and Conclusion

One of the objectives in this work has been to examine homotopy methods generated by three smooth penalty functions, the quadratic penalty for equality constraints and the log barrier and quadratic loss function for inequality constraints, and their modern counterparts, augmented Lagrangian or multiplier methods. We have shown that these methods are asymptotic to sequential quadratic programming and are competitive with these methods. However, considerable work will be required on large scale problems to assess their full potential.

The robustness and efficiency of this class of algorithms based on these smooth penalty functions, augmented Lagrangians, the derived homotopy, and predictor–corrector continuation techniques have been illustrated in Section 7. The methodology shows considerable promise and potential for solving constrained optimization problems; however, as with any method a word of caution is appropriate. One can construct simple examples illustrating the following situations for penalty paths: given $\epsilon > 0$ a penalty path may exist only for the penalty parameter $r \geq \epsilon$ or only for $r \leq \epsilon$, may not exist at all, may diverge, or may exist for $r > 0$ but the limit point at $r = 0$ is not a local minimum of the original problem. (When this last situation occurs, the Karush–Kuhn–Tucker equations have a singularity.) In spite of these examples penalty path following is a mathematically robust method of solving constrained optimization problems as is illustrated by the examples in Hock and Schittkowski (ref. 5), and these homotopy methods appear to make them efficient and a competitive alternative to the early stages of sequential quadratic programming.

REFERENCES

1. Poore, A.B., and Q. Al–Hassan, "The expanded Lagrangian system for constrained optimization problems", SIAM J. Control and Optimization, 2 (1988), 417–427.

2. Fletcher, R., Practical Methods of Optimization, Second Edition, John Wiley & Sons, Ltd., New York, 1987.

3. Stoer, J., "Principles of Sequential Quadratic Programming for Solving Nonlinear Programs", in: K. Schittkowski, ed., Computational Mathematical Programming, NATO ASI Series, Vol. F15, Springer–Verlag, Heidelberg, 1985.

4. Poore, A. B. and C. A. Tiahrt, "Bifurcation problems in nonlinear parametric programming", Mathematical Programming 39 (1987) 189–205.

5. Hock, W. and Schittkowski, K., Test Examples for Nonlinear Programming Code, Springer–Verlag, New York, 1981.

6. Keller, H. B., "Numerical solution of bifurcation and nonlinear eigenvalue problems", in P. Rabinowitz, ed., Applications of Bifurcation Theory, Academic Press, New York, pp. 359–384, 1977.

7. Shampine, L. F. and Gordon, M. K., Computer Solution of Ordinary Differential Equations: The Initial Value Problem, W. H. Freeman and Co., San Francisco, 1975.

8. Dennis, J. E. and Schnabel, R. B., Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice–Hall, Englewood Cliffs, NJ, 1983.