A Unified Gradient Flow Approach to Constrained Nonlinear Optimization Problems

S. Wang[†], X.Q. Yang[‡] and K.L. Teo[‡]
[†]School of Mathematics & Statistics
The University of Western Australia
35 Stirling Highway, Crawley, WA 6009, Australia
swang@maths.uwa.edu.au
[‡]Department of Applied Mathematics
The Hong Kong Polytechnic University
Kowloon, Hong Kong
mayangxq@polyu.edu.hk
mateokl@polyu.edu.hk

Abstract

This paper presents a unified gradient flow approach to nonlinear constrained optimization problems. This method is based on a continuous gradient flow reformulation of constrained optimization problems and on a two level time discretization of the gradient flow equation with a splitting parameter θ. The convergence of the scheme is analyzed and it is shown that the scheme becomes first order when $\theta \in [0, 1]$ and second order when $\theta = 1$ and the time discretization step length is sufficiently large. Numerical experiments for continuous, discrete and mixed discrete optimization problems were performed, and the numerical results show that the approach is effective for solving these problems.

Dedication: The authors dedicate this work to Lucien Polak for his many pioneer contributions to the field of computational optimization and optimal control.

1 Introduction

Constrained nonlinear programming problems appear in almost every subject in engineering, science, economics, finance and management. Many efficient methods have been developed for solving these problems, especially those with nonlinear inequality constraints. Excellent surveys of these methods can be found in, for example, [5, 7]. However, most of these methods fail to be effective when they are applied to problems with equality constraints without the use of other techniques such as penalty functions. Some specialized methods for solving equality constrained optimization problems have been developed: the

method proposed in [8], Tensor Methods in [4] and a preconditioned conjugate gradient approach in [1].

Recently, gradient flow type methods have been successfully used for solving this kind of problems by various authors (cf., for example, [2, 3, 9, 10]). In this approach an optimization problem is formulated as an ordinary differential equation (ODE) so that the solution of this ODE converges to an equilibrium point of the original problem as $t \to \infty$. This approach has the merit that once any of equality constraints is satisfied, the trajectory of the solution will stay on the manifold determined by that constraint. As such, several effective methods for ODEs can be used for the convergence analysis of this approach. Though this approach can be used for constrained nonlinear programming problems, especially for those with equality constraints, the effectiveness and efficiency of the method depend strongly on how to handle the following issues. One is whether the numerical solution of an ODE converges to an equilibrium point of the original problem within a given tolerance in a finite number of time steps. This is important as several existing schemes are based on time-stepping schemes which require very small time step sizes. Stability and robustness of numerical schemes for solving ODEs are also important as these guarantee that the errors in the discretized differential equation will not propagate as t goes to infinity. For example, the methods proposed in [2] and [3] are based on the forward Euler's time-stepping scheme which is known to be unstable for solving ODEs unless the step length in time is sufficiently small. Furthermore, how to handle points at which the problem is singular (in some sense) needs to be addressed. This kind of point is excluded from consideration in some existing work (eg. [2]).

In this paper, we propose a gradient flow approach which can be regarded as an improvement on that of [2]. This formulation has a gradient formula for the Lagrange multiplier that is simpler than that in [2]. We then propose a unified discretization scheme for the continuous gradient flow problem based on a two level implicit time discretization scheme with a splitting parameter θ. We will show that the solution from the discretized gradient flow equation converges to a local minimum of the original problem either linearly or quadratically depending on the choice of θ. We also discuss a technique to handle inequality constraints using slack variables and barrier functions. To demonstrate the usefulness of our method, three test problems are solved. The numerical results presented in this paper show that the method is effective for solving constrained nonlinear programming problems with either continuous variables or a mixture of continuous and discrete variables. The numerical results also show that the method can also handle concave programming problems with equality constraints in which optimal solutions are at corner points defined by the constraints.

2 Problems with equality constraints

Consider the following constrained nonlinear optimization problem

$$\begin{aligned} \text{minimize} \quad & f(x) && (1)\\ \text{subject to} \quad & g(x) = 0, && (2) \end{aligned}$$

where $x = (x_1, x_2, ..., x_n)^\top \in \mathbb{R}^n$, $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $g = (g_1, g_2, ..., g_m)^\top : \mathbb{R}^n \mapsto \mathbb{R}^m$. Here m and n are positive integers and $m \leq n$. This problem is referred to as **Problem 1**. Before further discussion we first make the following assumptions: The functions f and g are assumed to be three times continuously differentiable.

We define the Lagrangian of the problem by

$$L(x, u) = f(x) + u^\top g(x) \tag{3}$$

where $u = (u_1, u_2, ..., u_m)^\top \in \mathbb{R}^m$ is the Lagrangian multiplier. A pair of points (x^*, u^*) is said to be a *stationary point* of (3) if the following first order necessary optimality conditions hold:

$$L_x(x^*, u^*) = f_x(x^*) + g_x^\top(x^*)u^* = 0, \quad L_u(x^*, u^*) = g(x^*) = 0, \tag{4}$$

where $L_x(x, u) := \left(\frac{\partial L(x,u)}{\partial x_1}, ..., \frac{\partial L(x,u)}{\partial x_n}\right)^\top$, $L_u(x, u) := \left(\frac{\partial L(x,u)}{\partial u_1}, ..., \frac{\partial L(x,u)}{\partial u_m}\right)^\top$ and $f_x(x) := \left(\frac{\partial f(x)}{\partial x_1}, ..., \frac{\partial f(x)}{\partial x_n}\right)^\top$ denote the gradients of L and f, respectively, and

$$g_x(x) = \begin{pmatrix} \frac{\partial g_1(x)}{\partial x_1} & \frac{\partial g_1(x)}{\partial x_2} & \cdots & \frac{\partial g_1(x)}{\partial x_n} \\ \frac{\partial g_2(x)}{\partial x_1} & \frac{\partial g_2(x)}{\partial x_2} & \cdots & \frac{\partial g_2(x)}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial g_m(x)}{\partial x_1} & \frac{\partial g_m(x)}{\partial x_2} & \cdots & \frac{\partial g_m(x)}{\partial x_n} \end{pmatrix} \tag{5}$$

denotes the Jacobian of g. To fulfill these optimality conditions, we propose the following continuous gradient flow reformulation: for a given initial condition x_0, solve

$$\frac{dx(t)}{dt} = -L_x(x(t), u^0(x(t))) = -(f_x(x(t)) + g_x^\top(x(t))u^0(x(t))), \quad x(0) = x_0 \tag{6}$$

where the function $u^0(x)$ is chosen to satisfy

$$L_x(x, u^0(x)) = \tau g_x^\top(x)g(x), \tag{7}$$

in the least squares sense, where $\tau > 0$ is a parameter. The solution is denoted by $x(t, x_0)$. The choice of $u^0(x)$ given above is to guarantee that the the solution $(x(t), u^0(x(t)))$ from the above map converges locally to an equilibrium point satisfying (4). This will be demonstrated below.

Now, from the definition of *pseudo-inverse* (cf., for example, [6], page 37) and that of $L(x,u)$, defined in (3), we see that (7) becomes:

$$f_x(x) + g_x^\top(x)u^0(x) = \tau g_x^\top(x)g(x) \tag{8}$$

yielding a least squares solution $u^0(x)$ given by

$$u^0(x) = (g_x^\top(x))^+(\tau g_x^\top(x)g(x) - f_x(x)), \tag{9}$$

where the superscript + denotes the pseudo-inverse.

We assume that the linearly independence constraint qualification (LICQ) holds, i.e.,

$$g_{1x}(x), g_{2x}(x), \cdots, g_{mx}(x)$$

are linearly independent for any feasible point x. Thus, the $m \times m$ Gram matrix

$$G(x) := g_x(x)g_x^\top(x) \tag{10}$$

is non-singular for any feasible point x. Multiplying both sides of (8) by $g_x(x)$ gives

$$g_x(x)\left(f_x(x) + g_x^\top(x)u^0(x)\right) = \tau g_x(x)g_x^\top(x)g(x) \tag{11}$$

Therefore, the solution $u^0(x)$ in (9) can be expressed as

$$u^0(x) = G^{-1}(x)g_x(x)(\tau g_x^\top(x)g(x) - f_x(x)) = \tau g(x) - G^{-1}(x)g_x(x)f_x(x) \tag{12}$$

so long as $G(x)$ is non-singular. In particular, (12) holds true when x is sufficiently close to an equilibrium point x^*. Note that, when $G(x)$ is nonsingular, (12) follows also from (9) because $(g_x^\top(x))^+ = G^{-1}(x)g_x(x)$. This choice of u^0 is different from that in [2]. As it will be seen later, this choice has the merit that it reduces the complexity in the evaluation of the gradient of u^0 with respect to x. With this choice for $u^0(x)$, system (6) becomes

$$\dot{x} = \phi(x) := -\left(f_x(x) + g_x^\top(x)\left(\tau g(x) - G^{-1}(x)g_x(x)f_x(x)\right)\right). \tag{13}$$

The following theorem provides the convergence result of the scheme (6) and (7).

Theorem 2.1 *Assume that (x^*, u^*) is a stationary point of satisfying (4). Suppose the following conditions are satisfied:*

1. *LICQ holds at x^*,*

2. *for any $\overline{x}$ satisfying $g_x(x^*)\overline{x} = 0$, there holds*

$$\overline{x}^\top L_{xx}(x^*, u^*)\overline{x} > 0.$$

Then, the pair $(x(t), u^0(x(t)))$ *tends to* (x^*, u^*) *as* t *goes to* ∞*, provided that the starting point* x_0 *is sufficiently close to the point* x^**.*

PROOF. We will use the Poincaré-Lyapunov theorem (cf., for example, [11], p.88) to show that x^* is an asymptotically stable point for the scheme (6) and (7). According to the Poincaré-Lyapunov theorem, x^* is an asymptotically stable point for $\dot{x} = \phi(x)$ if $\phi(x)$ is continuously differentiable and if the original of the linearized system

$$\dot{y} = \phi_x(x^*)y,$$

where $y := x - x^*$, is exponentially stable (i.e., all eigenvalues of $\phi_x(x^*)$ are strictly negative).

First, we note that at x^*, $(x^*, u^0(x^*))$ satisfy (4) because the right-hand side of (7) is identically zero. This implies that $u^* = u^0(x^*)$.

Applying Taylor's expansion around the point x^* to the right-hand side of (13) gives the following linearized equation

$$\begin{aligned} \frac{dx}{dt} &= \phi(x^*) + \phi_x(x^*)(x - x^*) \\ &= -\left\{L_x(x^*, u^*) + \left[L_{xx}(x^*, u^*) + g_x^\top(x^*)u_x^0(x^*)\right](x - x^*)\right\} \\ &= -\left[L_{xx}(x^*, u^*) + g_x^\top(x^*)u_x^0(x^*)\right](x - x^*) \end{aligned} \tag{14}$$

since $L_x(x^*, u^*) = 0$. We now consider the term $u_x^0(x^*)$. Differentiating both sides of (11) with respect to x, evaluating the result at x^* and using (4), we obtain

$$g_x(x^*)\left(L_{xx}(x^*, u^*) + g_x^\top(x^*)u_x^0(x^*)\right) = \tau g_x(x^*)g_x^\top(x^*)g_x(x^*)$$

or

$$g_x(x^*)L_{xx}(x^*, u^*) + G(x^*)u_x^0(x^*) = \tau G(x^*)g_x(x^*).$$

Solving this for $u_x^0(x^*)$, we have

$$u_x^0(x^*) = -G^{-1}(x^*)g_x(x^*)L_{xx}(x^*, u^*) + \tau g_x(x^*).$$

Replacing the term $u_x^0(x^*)$ in (14) by the right-hand side of the above expression and noticing that $dx^*/dt = 0$, we obtain

$$\frac{dy}{dt} = -[Q(x^*)L_{xx}(x^*, u^*) + \tau P(x^*)]y \tag{15}$$

where $P(x) = g_x^\top(x)g_x(x)$, $Q(x) = I - g_x^\top(x)G^{-1}(x)g_x(x)$ and I denotes the $n \times n$ identity matrix. Since $g_x(x)Q(x) = 0$, $P(x)$ and $Q(x)$ are orthogonal and $Q(x)$ is an operator which projects a vector to the null space of $g_x(x)$.

We now show that the coefficient matrix $Q(x^*)L_{xx}(x^*,u^*)+\tau P(x^*)$ in (15) is positive definite.

Let $\lambda_i \in \mathbb{R}$ be an eigenvalue of $Q(x^*)L_{xx}(x^*,u^*)+\tau P(x^*)$ and $z_i \in \mathbb{R}^n$ an eigenvector corresponding to λ_i. Then, λ_i and z_i satisfy

$$[Q(x^*)L_{xx}(x^*,u^*)+\tau P(x^*)]z_i = \lambda_i z_i. \tag{16}$$

There are two cases to be considered.

Case 1: $g_x(x^*)z_i = 0$ (and thus $P(x^*)z_i = 0$).

Multiplying (16) from the left by $z_i^\top$, we obtain

$$z_i^\top (I + g_x^\top(x^*)G^{-1}(x^*)g_x(x^*))L_{xx}(x^*,u^*)z_i = \lambda_i ||z_i||^2.$$

Since $z_i^\top g_x^\top(x^*) = 0$, it follows that

$$\lambda_i = z_i^\top L_{xx}(x^*,u^*)z_i/||z_i||^2 > 0.$$

by the second assumption of the theorem.

Case 2: $g_x(x^*)z_i \neq 0$.

Multiplying (16) from the left by $z_i^\top P(x^*)$ and using the fact that $P(x^*)Q(x^*) = 0$, we obtain

$$\tau z_i^\top P^2(x^*)z_i = \lambda_i z_i^\top P(x^*)z_i.$$

But $z_i^\top P(x^*)z_i = ||g_x z_i||^2$ and $z_i^\top P^2(x^*)z_i = (g_x(x^*)z_i)^\top G(x^*)(g_x(x^*)z_i)$. So, the above equation gives

$$\lambda_i = \tau(g_x(x^*)z_i)^\top G(x^*)(g_x(x^*)z_i)/||g_x(x^*)z_i||^2 > 0$$

because $G(x^*)$ is positive definite by the assumption that LICQ is satisfied at x^*.

Therefore, combining the two cases, we have that $\lambda_i > 0$ for all $i = 1, 2, ..., n$, i.e., $-(Q(x^*)L_{xx}(x^*,u^*)+\tau P(x^*))$ in (15) is negative definite at x^*. By the Poincaré-Lyapunov theory, $\lim_{t\to\infty} y(t) = 0$, or $x(t) \to x^*$ as $t \to \infty$. Finally, from the choice of $u^0(x)$ we see that, when $t \to \infty$, $(x^*, u^0(x^*))$ satisfies (4). This implies that $u^0(x^*) = u^*$. □

The following theorem shows that $f(x(t))$ is strictly decreasing along $g(x(t)) = 0$ for $t \geq t_0$, where $x(t)$ is the solution of system (6).

Theorem 2.2 *Let $(x(t), u^0(x(t)))$ be the solution of (6) and (9). For a fixed $t_0 \geq 0$, if $g(x(t_0)) = 0$, then $g(x(t)) = 0$ for all $t \geq t_0$. Furthermore, if, in addition, $L_x(x(t), u^0(x(t))) \neq 0$ for all $t > t_0$, then, $f(x(t))$ is strictly decreasing for all $t > t_0$.*

PROOF. Suppose $g(x(t_0)) = 0$ for some $t_0 \geq 0$. Let us consider the solution form of the system (13) in the region $t \geq t_0$. Multiplying both sides of (13) by $g_x(x)$ and using (10), we have

$$g_x(x)\frac{dx}{dt} = -\tau G(x)g(x), \quad \text{or,} \quad \frac{dg(x)}{dt} = -\tau G(x)g(x).$$

The solution $x(t)$ of this system can be implicitly expressed as

$$g(x(t)) = \exp\left(-\tau \int_{t_0}^{t} G(x(s))ds\right)\xi, \quad t \geq t_0,$$

where ξ denotes a vector dependent of t_0, but independent of t. Applying $g(x(t_0)) = 0$ to this form gives $\xi = 0$. Therefore, $g(x(t)) = 0$ for all $t \geq t_0$.

We now prove the second part of the theorem. Differentiating $f(x(t))$ with respect to t and using (6) and (11) we have

$$\begin{aligned}
& \frac{df(x(t))}{dt} \\
= \; & f_x^\top(x(t))\frac{dx(t)}{dt} \\
= \; & -f_x^\top(x(t))[f_x(x(t)) + g_x^\top(x(t))u^0(x(t))] \\
= \; & -[f_x(x(t)) + g_x^\top(x(t))u^0(x(t))]^\top[f_x(x(t)) + g_x^\top(x(t))u^0(x(t))] \\
& \quad +(u^0(x(t)))^\top g_x(x(t))[f_x(x(t)) + g_x(x(t))^\top u^0(x(t))] \\
= \; & -||L_x(x(t), u^0(x(t)))||^2 + \tau((u^0(x(t)))^\top g_x(x(t)))(g_x^\top(x(t))g(x(t))) \\
= \; & -||L_x(x(t), u^0(x(t)))||^2 + \tau(u^0(x(t)))^\top G g(x(t)).
\end{aligned}$$

Therefore, if $L_x(x(t), u^0(x(t))) \neq 0$ and $g(x(t)) = 0$ when $t \geq t_0$, then $\frac{df(x(t))}{dt} < 0, t \geq t_0$. Thus $f(x(t))$ is strictly decreasing with respect to t when $t \geq t_0$. □

Remark 2.1: We comment that the conditions in Theorem 2.1 are sufficient. As will be seen in Section 5, the method works well event some of these conditions are not satisfied. In particular, the case that LICQ is not satisfied at a finite number of points can be handled by a technique at the algorithm design level, as presented in the next section. Also, the numerical results in Section 4 show that the method converges globally.

3 The algorithm

The formulation in the previous section is a continuous one. We now discuss the discretization of this formulation.

Let $0 = t_0 < t_1 < \cdots < t_k < \cdots$ be a series of time points and put $h_k = t_{k+1} - t_k$. We discretize (6) by the following implicit time-stepping algorithm:

$$\frac{x_{k+1} - x_k}{h_k} = -\big[(1-\theta)L_x(x_k, u^0(x_k)) + \theta L_x(x_{k+1}, u^0(x_{k+1}))\big],$$

where $\theta \in [0, 1]$ is a parameter and $u^0(x)$ is given in (9) or (12). From the above we have

$$x_{k+1} = x_k - h_k\big[(1-\theta)L_x(x_k, u^0(x_k)) + \theta L_x(x_{k+1}, u^0(x_{k+1}))\big]. \tag{17}$$

This discretization becomes the explicit forward Euler's scheme when $\theta = 0$ and the full implicit backward Euler's scheme when $\theta = 1$. Note that (17) is nonlinear in x_{k+1} if $\theta \neq 0$. For any k, we let $u_k = u^0(x_k)$. Using a Taylor's expansion we have

$$L_x(x_{k+1}, u_{k+1}) = L_x(x_k, u_k) + H(x_k, u_k)\delta x_k + \phi(\delta x_k) \tag{18}$$

where $\delta x_k = x_{k+1} - x_k$, $H(x_k, u_k) = L_{xx}(x_k, u_k) + L_{xu^0}(x_k, u_k)u_x(x_k)$ and ϕ is the remainder satisfying

$$||\phi(\delta x_k)|| = \mathcal{O}(||\delta x_k||^2). \tag{19}$$

From (6) we see that

$$H(x_k, u_k) = L_{xx}(x_k, u_k) + L_{xu^0}(x_k, u_k)u^0_x(x_k) = f_{xx}(x_k) + J(x_k, u^0(x_k)) + g_x(x_k)^\top u^0_x(x_k) \tag{20}$$

where f_{xx} denotes the Hessian of f and $J(x, u^0(y)) := r_x(x, u^0(y))$ with $r(x, u^0(y)) := g_x^\top(x)u^0(y)$ (i.e. the Jacobian of $g_x(x)^\top u^0(y)$ with respect to x with g_x defined in (5)). All the terms on the left side of (20) are $n \times n$ matrices. Substituting (18) into (17) we have

$$\begin{aligned} x_{k+1} &= x_k - h_k\big[(1-\theta)L_x(x_k, u_k) + \theta(L_x(x_k, u_k) + H(x_k, u_k)\delta x_k + \phi(\delta x_k))\big] \\ &= [I + h_k\theta H(x_k, u_k)]x_k - h_k\theta H(x_k, u_k)x_{k+1} - h_k L_x(x_k, u_k) - h_k\theta\phi(\delta x_k), \end{aligned} \tag{21}$$

where $u_k := u^0(x_k)$ with u^0 defined by (9) or (12). To obtain an algorithm, omit the unknown higher order term $\phi(\delta x_k)$ in (21) and solve the resultant equation for x_{k+1} to obtain:

$$\begin{aligned} x_{k+1} &= \big[I + \theta h_k H(x_k, u_k)\big]^{-1}\big[(I + \theta h_k H(x_k, u_k)x_k - h_k L_x(x_k, u_k)\big] \\ &= x_k - h_k\big[I + \theta h_k H(x_k, u_k)\big]^{-1} L_x(x_k, u_k) \end{aligned} \tag{22}$$

for any $\theta \in [0, 1]$, where $u_k = u(x_k)$ given by (9) or (12). Starting from an initial guess x_0, (22) defines a series $\{x_k\}_{k=1}^\infty$. The convergence of this series is given in the following theorem.

Theorem 3.1 *Let $\{x_k\}_1^\infty$ be the sequence defined by (22) and (x^*, u^*) be a solution to (4) such that LICQ holds. If the initial guess x_0 is sufficiently close x^*, then we have*

1. *x_k converges linearly to x^* if $\theta \in [0, 1]$ and $h_k > 0$ is sufficiently small,*
2. *x_k converges to x^* quadratically if $\theta = 1$ and $h_k \to \infty$.*

PROOF. For brevity we only prove the case that the time step lengths are independent of k, i.e. $h_k = h$ for all k. We now prove the two cases separately.

Case 1.

Rewriting (22) as

$$[I + \theta h H(x_k, u_k)](x_{k+1} - x_k) = -hL_x(x_k, u_k),$$

we have that

$$x_{k+1} = x_k - h\big[L_x(x_k, u_k) + \theta H(x_k, u_k)(x_{k+1} - x_k)\big]$$

Let $u^* = u^0(x^*)$ and $e_k = x_k - x^*$. Subtracting x^* from both sides of the the above equality, noticing that $L(x^*, u^*) = 0$ and $\delta x_k = e_{k+1} - e_k$, and using the mean value theorem we have

$$\begin{aligned} e_{k+1} &= e_k - h\left[L_x(x_k, u_k) - L(x^*, u^*) + \theta H(x_k, u_k)(e_{k+1} - e_k)\right] \\ &= e_k - h\left[H(\xi_k, u^0(\xi_k))e_k + \theta H(x_k, u_k)(e_{k+1} - e_k)\right] \\ &= \left[I + h\theta H(x_k, u_k) - hH(\xi_k, u^0(\xi_k))\right] e_k - \theta h H(x_k, u_k)e_{k+1}, \end{aligned}$$

where ξ_k denotes a point in the line segment connecting x_k and x^*. Solving this for e_{k+1}, we get

$$\begin{aligned} e_{k+1} &= \big[I + \theta h H(x_k, u_k)\big]^{-1}\big[I + h\theta H(x_k, u_k) - hH(\xi_k, u^0(\xi_k))\big]e_k \\ &= \Big\{I - h\big[I + \theta h H(x_k, u_k)\big]^{-1} H(\xi_k, u^0(\xi_k))\Big\} e_k \end{aligned}$$

Taking the norm on both sides of this equality we obtain

$$||e_{k+1}|| \leq \rho(x_k, \xi_k, \theta, h)||e_k||, \tag{23}$$

where

$$\rho(x_k, \xi_k, \theta, h) = ||I - h\big(I + \theta h H(x_k, u_k)\big)^{-1} H(\xi_k, u^0(\xi_k))||. \tag{24}$$

¿From the estimate (23) we see that if $\rho(x^*, u^*, \theta, h) < 1$, the first order terms in (23) show that e_k converges to zero linearly. To show this, we first prove that the Hessian $H(x^*, u^*)$ is positive definite. Differentiating both sides of (8) (or equivalently (7)) with respect to x gives

$$H(x, u^0(x)) \equiv L_{xx}(x, u^0(x)) + L_{xu^0}(x, u^0(x))u'(x) = \tau((g_x^\top(x))_x g(x) + g_x(x)g_x^\top(x).$$

By Theorem 2.1, when $(x, u^0(x)) \to (x^*, u^*)$, the above reduces to

$$H(x^*, u^*) = \tau G(x^*),$$

and thus $H(x^*, u^*)$ is positive definite because $G(x^*)$ is positive definite for any x^* satisfying the LICQ condition.

Now, when x_k is sufficiently close to x^*, from (24) it is seen that if $\theta \in [0, 1]$ and h is sufficiently small,

$$\rho(x_k, \xi_k, \theta, h) \leq \left(1 - \frac{h\lambda_{\min}^k}{1 + \theta h \lambda_{\max}^k}\right) < 1,$$

where $\lambda_{\min}^k$ and $\lambda_{\max}^k$ denote the minimum and maximum eigenvalues of $H(x_k, u_k)$, respectively. Therefore, (23) implies that

$$\lim_{k\to\infty} e_k = 0$$

linearly. This proves that x_k converges to x^* linearly.

Case 2.
Rewrite (22) with $\theta = 1$ and $h_k = h$ as

$$\frac{x_{k+1} - x_k}{h} = -\left[L_x(x_k, u_k) + H(x_k, u_k)\delta x_k\right].$$

When $h \to \infty$, the above equation reduces to

$$H(x_k, u_k)\delta x_k + L_x(x_k, u_k) = 0.$$

This coincides with the equation resulting from the application of the classic Newton's method to $L_x(x, u^0(x)) = 0$. Therefore, using the well known theoretical result for Newton's method we have that the x_k converges quadratically to x^* if x_0 is sufficiently close to x^* (cf., for example, [7]). This completes the proof. □

An interesting feature of Theorem 3.1 is that the method allows sufficiently large h when the splitting parameter $\theta = 1$. In this case, x_k converges quadratically to a local minimum.

Based on (22) and Theorem 3.1 we propose the following discrete Gradient Flow algorithm.

Algorithm GF:

1. Choose $\epsilon > 0$ and a $\theta \in [0, 1]$, a sequence of time step sizes $\{h_k\}$ and an initial guess $x_0 \in \mathbb{R}^n$. Let $k = 0$.

2. Compute

$$\begin{aligned} u_k &= (g_x^\top(x_k))^+(\tau g_x^\top(x_k)g(x_k) - f_x(x_k)), && (25)\\ x_{k+1} &= x_k - h_k\big[I + \theta h_k H(x_k, u_k)\big]^{-1} L_x(x_k, u_k) \\ &\quad + \exp\big[-t_k(||f_x(x_k)||^2 + ||g_x(x_k)||^2)\big]||g(x_k)||^2 E && (26)\end{aligned}$$

   where $E = (1, 1, ..., 1)^\top \in \mathbb{R}^n$.

3. If $||L_x||_2 + ||g||_2 \le \varepsilon$, stop, otherwise let $k = k + 1$ and return to Step 2.

Remark 3.1. When $G(x_k)$ is non-singular, instead of using (25), we can use (12) to evaluate u_k. In this case, both (25) and (12) yield the same solution u_k. However, in the

case that $G(x_k)$ is singularly at a point, (25) prevents the algorithm from breaking down at the point by finding a least squares solution. Theorem 2.1 guarantees that $||g(x_k)||_2$ remains bounded even $G(x_k)$ is singular.

Remark 3.2. The last term in (26) is to prevent the solution sequence from being trapped at isolated points where $f_x(x) = 0$ and $g_x(x) = 0$, but $g(x) \neq 0$. Such a point is called a singular point for the problem. If this singular situation happens at x_k for some k, (26) will generate a new starting point x_{k+1}. Obviously, this artificial term goes to zero exponentially if one of $||f_x||$ and $||g_x||$ is not zero.

When $\theta \neq 0$, Algorithm GF needs the evaluation of u_x appeared in (20). This can be done by differentiating (7). We will present a detailed discussion on a more general situation in the next section which contains this as a special case.

4 Inequality constraint considerations

The above approach is for problems of (1) and (2) in which only equality constraints are involved. In this section we shall discuss the handling of inequality constraints. This is based on the state-space transformation technique proposed in [2].

Consider nonlinear programming problems of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^{n_1}} \quad & f_1(x) && (27)\\ \text{subject to} \quad & g_i(x) = 0, \quad i = 1, 2, ..., m_1 && (28)\\ & p_i(x) \leq 0, \quad i = 1, 2, ..., m_2 && (29) \end{aligned}$$

where n_1 is a positive integer, m_1, m_2 are non-negative integers, $f_1 : \mathbb{R}^{n_1} \mapsto \mathbb{R}$, $g = (g_1, ..., g_{m_1})^\top : \mathbb{R}^{n_1} \mapsto \mathbb{R}^{m_1}$ and $p = (p_1, ..., p_{m_2})^\top : \mathbb{R}^{n_1} \mapsto \mathbb{R}^{m_2}$. This problem is referred to as **Problem 2**.

Introducing slack variables $(s_1, s_2, ..., s_{m_2}) \in \mathbb{R}^{m_2}$, the inequality constraints (29) can be re-written as

$$g_{m_1+i}(x, s) := p_i(x) + s_i = 0 \qquad i = 1, 2, ..., m_2$$

with $s := (s_1, ..., s_{m_2})^\top \geq 0$. Using this, (27)–(29) can be rewritten as

$$\begin{aligned} \min_{(x,s) \in \mathbb{R}^n} \quad & f_1(x), && (30)\\ \text{subject to} \quad & g(x, s) = 0, && (31)\\ & s \geq 0, && (32) \end{aligned}$$

where $g = (g_1, g_2, ..., g_m)^\top : \mathbb{R}^n \mapsto \mathbb{R}^m$, $m = m_1 + m_2$ and $n = n_1 + m_2$. Let $y = \begin{pmatrix} x \\ s \end{pmatrix} \in \mathbb{R}^n$. Note that the projection of the feasible set defined by (31) and (32) onto $\mathbb{R}^{n_1}$ should be equivalent to the feasible set for Problem 2.

Now, let D be an $n \times n$ diagonal matrix defined by

$$D(y) = \text{diag}(1, 1, ..., 1, s_1, s_2, ..., s_{m_2}), \tag{33}$$

and define

$$L(y, u) = f(y) + u^\top g(y), \quad \text{where } f(y) = f_1(x)$$
$$\tilde{L}(y, u) := D^\alpha(y) L_x(y, u) = D^\alpha(y)\big[f_y(y) + g_y^\top(y)u\big]$$

for $u \in \mathbb{R}^m$. Using the $\tilde{L}(y, u)$ defined above, we propose the following gradient flow formula with barrier

$$\frac{dy}{dt} = -\tilde{L}(y, u^0(y)) \tag{34}$$

with α a positive real number and $u^0(y)$ given by

$$\tilde{L}(y, u^0(y)) = D^\alpha(y)(f_y(y) + g_y^\top(y)u^0(y)) = \tau g_y^\top(y)g(y). \tag{35}$$

Note that in this formulation the diagonal matrix $D^\alpha(y)$ is a barrier to prevent the solution y to move across the hyperplane that one of the components of y equals zero. Obviously, this is exactly in the same form as that of (6) and (7) (with x and $L_x(x, u^0(x))$ replaced with y and $\tilde{L}(y, u^0(y))$. Therefore, by Theorem 2.1, the solution $y(t)$ converges to a local minimum of (27)-(29) and $g(y(t)) \to 0$ as $t \to \infty$. Let

$$\mathbb{R}^n_* = \{(x_1, ..., x_n) \in \mathbb{R}^n : x_i \geq 0, i = n_1 + 1, ..., n\}.$$

Then the above continuous form has the property that if an initial condition $y_0 \in \mathbb{R}^*$, the solution $y(t)$ of (34) also satisfies $y \in \mathbb{R}^n_*$. This is obvious since, for any $i = n_1 + 1, ..., n$, the origin 0 is an equilibrium point for the ith equation of (34). Therefore, the continuous solution can never go across the barrier $x_i = 0$ for $i = n_1 + 1, ..., n$. However, although the solution to the continuous problem satisfies (32), the series $\{y_k\}$ from the discretized form (17) may violate this constraint when h_k is large. Thus, a control on the time step sizes has to be introduced to avoid this overshooting. This is given in the following algorithm of Gradient Flow with Barrier.

Algorithm GFB

1. Choose a $\theta \in [0, 1]$, a tolerance ε, a set of time step sizes $\{h_k\}$ and an initial guess $y_0 \in \mathbb{R}^n_*$. Let $k = 0$.

2. Compute

$$u_k := u^0(x_k) = (g_y^\top(y_k))^+(\tau D^{-\alpha}(y_k) g_y^\top(y_k) g(y_k) - f_y(y_k)). \tag{36}$$

3. Compute

$$\begin{aligned} y_{k+1} &= y_k - h_k\big[I + \theta h_k H(y_k, u_k)\big]^{-1}\tilde{L}(y_k) \\ &\quad + \exp\big[-t_k(||f_y(y_k)||^2 + ||g_y(y_k)||^2)\big]||g(y_k)||^2 E \end{aligned} \tag{37}$$

where $E = (1, 1, ..., 1)^\top \in \mathbb{R}^n$, $\tilde{L}(y, u) = D^\alpha(y)L_y(y, u)$ and

$$H(y_k, u_k) = \frac{d}{dy}\tilde{L}(y, u^0(y))\Big|_{(y_k, u_k)}.$$

4. If $||\tilde{L}(y_k, u^0(y_k))||_2 + ||g(y_k)||_2 \leq \varepsilon$, stop. Otherwise, continue.

5. If $y_{k+1} \in \mathbb{R}^n_*$, then $k = k + 1$ and GOTO Step 2. Otherwise, $h_k = h_k/2$ and GOTO Step 3.

Remark 4.1: Algorithm GFB is just a basic algorithm to perform the barrier gradient flow method. More refinements can be added. For example, in Step 5, h_k is halved if $y_{k+1} \in \mathbb{R}^n_*$. This may course the problem that h_k becomes very small. Therefore, if $y_{k+1} \in \mathbb{R}^n_*$, we may increase h_k gradually to avoid the situation to happen.

Remark 4.2: The iterative process in Step 5 of Algorithm GFB will be terminated in a finite number of iterations if the iterate y_k is in the interior of $\mathbb{R}^n_*$. This is because when the distance between the boundary of the feasible region and y_k is positive, there exists a fixed $h_k = h_{k-1}/2^n$ (and so a fixed n) such that the y_{k+1} given by (37) is inside the region. Note that it is possible that y_k is very close to the (finite) boundary of $\mathbb{R}^n_*$ so that a large number of iterations is needed in Step 5 of Algorithm GFB. In this case, the sequence generated by the algorithm converges to a generalized equilibrium point satisfying $\tilde{L}(y, u^0(y)) = 0$, though $L_y(y, u^0(y))$ may not be zero.

The evaluation of term $H(y_k, u_k)$ in (37) involves that of $u^0_y(x_k)$. We now discuss this. Differentiating both sides of (35) with respect to y gives

$$\alpha D^{\alpha-1}(y)D_y(y)(f_x(y) + g_x^\top(y)u^0(y)) + D^\alpha(y)\big[f_{yy}(y) + J_{g_y^\top u^0}(y) + g_y^\top(y)u^0_y(y)\big] = \tau J_{g_y^\top g}(y),$$

where $J_{g_y^\top u^0}$ denotes the Jacobian of $g_y^\top(y)u^0$ with u^0 being regarded as a constant and $J_{g_y^\top g}$ the Jacobian of $g_y^\top(y)g(y)$. (Note the left-hand side of this equality also gives a representation for $H(y, u^0(y))$ defined in Step 2 of Algorithm GFB.) ¿From (33) we have that $D_y(y) = \mathrm{diag}(0, ..., 0, 1, ..., 1)$. Solving the above equation for $g_y^\top(u^0_y)^\top$ we have

$$g_y^\top u^0_y = D^{-\alpha}\big[\tau J_{g_x^\top g} - \alpha D^{\alpha-1}D_y(f_x + g_x^\top u^0)\big] - (f_{yy} + J_{g_x^\top u^0}).$$

In this expression we have dropped the independent variable y. Taking pseudo-inverse in the least squares sense we obtain

$$u^0_y = (g_y^\top)^+\Big\{D^{-\alpha}\big[\tau J_{g_y^\top g} - \alpha D^{\alpha-1}D_y(f_y + g_y^\top u^0)\big] - (f_{yy} + J_{g_y^\top u^0})\Big\}. \tag{38}$$

In the case that G is non-singular, (38) can be expressed as

$$u_y^0 = G^{-1} g_y \Big\{ D^{-\alpha} \big[\tau J_{g_y^\top g} - \alpha D^{\alpha-1} D_y (f_y + g_y^\top u^0) \big] - (f_{yy} + J_{g_y^\top u^0}) \Big\}. \tag{39}$$

We comment that these formulas for u_y are much simpler than that in [2]. The latter is based on the expression $g_y D^\alpha L_y = \tau g$ instead of (35). A special case of (38) or (39) is to set $D = I_{n\times n}$ and $D_y = 0$. In this case, (38) and (39) reduce, respectively, to

$$\begin{aligned} u_y^0 &= (g_y^\top)^+ \Big[\tau J_{g_y^\top g} - (f_{yy} + J_{g_y^\top u^0}) \Big], \\ u_y^0 &= G^{-1} g_y \Big[\tau J_{g_y^\top g} - (f_{yy} + J_{g_y^\top u^0}) \Big]. \end{aligned}$$

These provide formulas for u_y^0 used in (26).

5 Numerical experiments

To demonstrate the usefulness of Algorithms GF and GFB, numerical experiments were performed. The algorithms are coded using the MATLAB programming language. In all the examples solved below, the Jacobians and Hessians were computed symbolically using the MATLAB symbolic tool box and both of the parameters θ and τ in Algorithms GF and GFB are chosen to be 1. All the numerical computations were performed in MATLAB long format, and the stopping criterion is chosen to be

$$E = ||L_x(x,u)||_2 + ||g(x)||_2 \le \varepsilon \tag{40}$$

where $\varepsilon > 0$ is the tolerance.

Test 1. The first example is chosen to be

$$\begin{aligned} \min_{x\in\mathbb{R}^5} \quad & f(x) = \sum_{i=1}^{5} (x_i + x_i^2 + x_i^3) \\ \text{subject to} \quad & \sum_{i=1}^{5} (x_i - 1)^2 = 1. \end{aligned}$$

This problem was solved for various values of h and the following 7 initial conditions

$$\begin{aligned} x_0^1 &= (0.9999999, 1.0000001, 1, 1, 1) \\ x_0^2 &= (-1, -1, -1, -1, -1) \\ x_0^3 &= (-5, -5, -5, -5, -5) \\ x_0^4 &= (-300, -200, -50, -100, -500) \\ x_0^5 &= (-1000, -2000, -1000, -100, -500) \\ x_0^6 &= (-3000, -3000, -5000, -2000, -5000) \\ x_0^7 &= (0, 0, 0, 0, 10^{12}) \end{aligned}$$

	$h = 10^{-1}$	$h = 1$	$h = 10$	$h = 10^2$	$h = 10^3$
x_0^1	59 (339)	14 (248)	9	8	7
x_0^2	61 (190)	18 (8)	12	11	10
x_0^3	64 (201)	21 (10)	15	14	13
x_0^4	80 (631)	45 (384)	36	32	31
x_0^5	81 (759)	38 (63)	32	31	30
x_0^6	91 (945)	51 (498)	53	47	52
x_0^7	127 (463)	84 (307)	78	77	76

Table 1: Results of Test 1 for different initial guesses and h.

The tolerance ε is chosen to be $= 10^{-8}$ in (40). The iteration counts for the different values of h and initial guesses are listed in Table 1. From this table we see that the rate of convergence of the method is linear when h is small and quadratic when h is large. To compare our method with that in [3], we also solved the problem by using the Barrier Newton Gradient Flow method proposed from [2] with $h = 0.1$ and $h = 1$. The iteration counts are also listed in Table 1 in brackets. Theoretically, the rate convergence of the algorithm in [2] is linear when $h = 0.1$ and quadratic when $h = 1$. However, from the table we see that the number of iterations of the latter depends strongly on the initial conditions. In contrast to this, the performance of our method is very uniform with respect to all the initial conditions. This demonstrates that the present method is much more robust than that of [2]. To show this further, we plot in Figure 1 the errors E from our method with $h = 1000$ and Evtushenko & Zhadan's method with $h = 1$ for the initial guess x_0^7. In this case, both of the methods converge quadratically by theory. From this figure it is seen that the error from our method is monotonically decreasing except for the first iteration. However, the error from Evtushenko & Zhadan's method goes down quickly to a certain magnitude, but it oscillates around this value for about 300 iterations before converges to zero. This phenomenon happens for all of the four initial guesses x_0^1, x_0^4, x_0^6 and x_0^7. Obviously, our method is more stable than that of Evtushenko and Zhadan [2]. This is because the discretization in the latter is based the forward Euler's time-stepping method which is known to be unstable unless the mesh size is sufficiently small, while our discretization is based on an implicit time-stepping method which is absolutely stable.

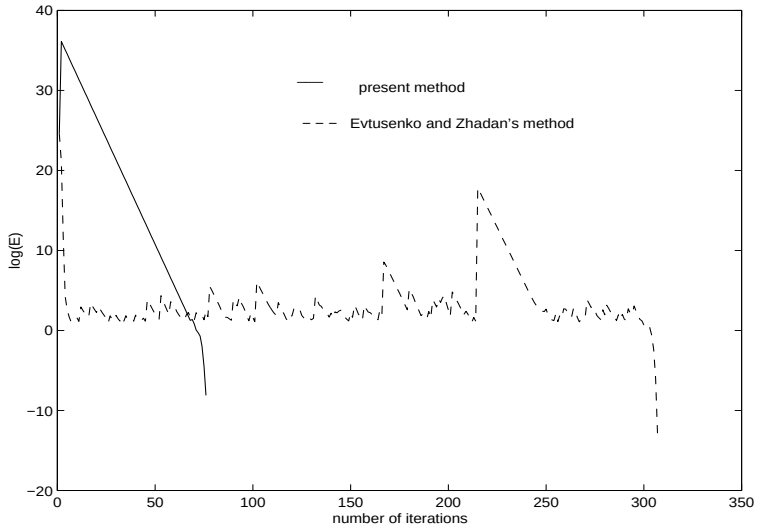

Figure 1: Computed errors from the present and Evtushenko & Zhadan's methods.

Test 2. Consider the following constrained concave programming problem

$$\begin{aligned} \min_{x\in\mathbb{R}^5} \quad & f(x) = -\sum_{i=1}^{5} x_i^2, \\ \text{subject to} \quad & \sum_{i=1}^{5} x_i \le 1, \\ & x_1 \le 0.8, \\ & x_i \ge 0, \quad i = 1, 2, ..., 5. \end{aligned}$$

This problem has inequality constraints. Thus, we introduce two slack variables x_6 and x_7 to convert the problem into the following one

$$\begin{aligned} \min_{x\in\mathbb{R}^7} \quad & f(x) = -\sum_{i=1}^{5} x_i^2, \\ \text{subject to} \quad & \sum_{i=1}^{5} x_i - 1 + x_6 = 0, \\ & x_1 - 0.8 + x_7 = 0, \\ & x_i \ge 0, \quad i = 1, 2, ..., 7. \end{aligned}$$

The transformed problem has 7 decision variables. To solve this problem we use Algorithm GFB in Section 4 because of the bound constraints. Note that the problem is a concave quadratic programming problem and a minimum of this problem is at one of the 'corners' of the feasible region defined by the linear constraints. This makes the problem hard to solve by a gradient-based method as the corner points are not smooth. Therefore, the

optimality condition $\tilde{L}_y(y,u)=0$ may not be satisfied at these points. However, we may find approximations to the minimum by relaxing the stopping criterion (40), i.e., we set the tolerance ε to a moderate value. Based on this, the following four sets of values of τ, h_0, ε and the initial guess x_0 were used.

$$\begin{aligned}
&\tau=1, \quad h_0=10, \quad \varepsilon=10^{-4}, \quad x_0=(10,5,1,2,5,1,2);\\
&\tau=0.1, \quad h_0=1, \quad \varepsilon=10^{-4}, \quad x_0=(1,1,1,1,1,1,1);\\
&\tau=1, \quad h_0=1, \quad \varepsilon=5\times10^{-3}, \quad x_0=(0.1,0.1,0.1,0.1,0.1,0.1,0.1);\\
&\tau=1, \quad h_0=10, \quad \varepsilon=10^{-3}, \quad x_0=(1,5,10,2,1,4,7).
\end{aligned}$$

Corresponding to these choices we obtained the following four local minima.

$$\begin{aligned}
&x_{\text{opt}}=(0.0000,0.5000,0.0000,0.0000,0.5000,0.0000,0.8000), \quad f_{\text{opt}}=-0.5000\\
&x_{\text{opt}}=(0.0000,0.2500,0.2500,0.2500,0.2500,0.0000,0.8000), \quad f_{\text{opt}}=-0.2500\\
&x_{\text{opt}}=(0.8000,0.0470,0.0470,0.0470,0.0470,0.0123,0.0000), \quad f_{\text{opt}}=-0.6488\\
&x_{\text{opt}}=(0.0000,0.0002,0.9998,0.0000,0.0000,0.0000,0.7999), \quad f_{\text{opt}}=-0.9995
\end{aligned}$$

Obviously, the last in the above is a global minimum of the problem, while the rest are local minima.

Test 3. Consider the following mixed discrete optimization problem

$$\begin{aligned}
\min \quad & f(x)=(x_1-2)^2+(x_2-2)^2+(x_3-2.8)^2\\
\text{subject to} \quad & x_1,x_2\geq 0,\\
& x_3\in\{1,2,3,4,5,6,7\}.
\end{aligned}$$

To solve this problem, we use the transformation technique proposed in [12] to transform it into the following continuous problem

$$\begin{aligned}
\min \quad & f(x)=(x_1-2)^2+(x_2-2)^2+(\sum_{j=1}^{7} jy_j-2.8)^2+\beta(\sum_{j=1}^{7} y_j-1)^2\\
& +\gamma\sum_{j=1}^{7}(j^2y_j-jy_j^2)\\
\text{subject to} \quad & \sum_{j=1}^{7} y_j-1=0,\\
& \sum_{j=1}^{7}(j^2y_j-jy_j^2)=0,\\
& x_i,y_j\geq 0, \quad i=1,2,\ j=1,2,...,7,
\end{aligned}$$

where β and γ are two non-negative constants. In this formulation the last two terms are penalties. When the equality constraints are satisfied, these penalty terms are zero. It is also proved in [12] that when the equality and bound constraints on y are satisfied, one y_j is unity and the rest are zeros. Thus, $x_3 = \sum_{j=1}^{7} j y_j$ guarantees that x_3 takes one of the values in the discrete set $\{1, 2, ..., 7\}$. Note the solutions to this problem are also corner points of the constraints which are not smooth. To solve this problem using Algorithm GFB, we choose $\alpha = 1 = \beta$ and $\varepsilon = 10^{-2}$. The following three different sets of τ, h_0 and x_0 were used:

$$\begin{aligned}
&\tau = 0.1, \quad h_0 = 0.001, \quad x_0 = (1, 0.5, 1, 1, 1, 1, 1, 1, 1);\\
&\tau = 1, \quad h_0 = 0.1, \quad x_0 = (5, 0.1, 2, 1, 2, 3, 2, 1, 1);\\
&\tau = 1, \quad h_0 = 0.1, \quad x_0 = (5, 5, 7, 7, 7, 7, 7, 7, 7).
\end{aligned}$$

Corresponding to these starting points, the following solutions were obtained

$$\begin{aligned}
x^1_{\text{opt}} &= (2.0000, 2.0000, 0.0000, 0.0000, 0.9980, 0.0022, 0.0000, 0.0000, 0.0000);\\
x^2_{\text{opt}} &= (2.0000, 2.0000, 0.0000, 0.0001, 0.9999, 0.0000, 0.0000, 0.0000, 0.0000);\\
x^3_{\text{opt}} &= (2.0000, 2.0000, 0.0000, 0.0016, 0.9731, 0.0280, 0.0008, 0.0000, 0.0000).
\end{aligned}$$

All these solutions suggest that the solution to the original problem is $x_1 = x_2 = 2$ and $x_3 = 3$.

6 Conclusion

In this paper we first proposed a gradient flow formulation of a nonlinear equality constrained optimization problem. We then presented a discretization technique with a splitting parameter for the gradient flow equation. It has been shown that the solution to the discretized problem converges to a local minimum of the original problem either linearly or quadratically, depending on the choice of the splitting parameter. Techniques for handling inequality constraints were also discussed. Numerical experiments on both convex and non-convex problems with and without discrete decision variables were performed and the numerical results show that the method is efficient and useful.

Acknowledgement

The authors would like to thank the two referees, Professors Bingsheng He and David Mayne for their patience and valuable comments and suggestions. They have pointed out several errors in the earlier versions of the paper. Professor Mayne has also made some

useful suggestions to strengthen some results of the paper. This is far beyond the duty of a referee. We are deeply indebted to both of them who agreed to disclose their names to us upon our request.

References

[1] T.F. Coleman and A. Verma, "A preconditioned conjugate gradient approach to linear equality constrained minimization". *Comput. Optim. Appl.*, **20**, 61–72 (2001).

[2] Y.G. Evtushenko and V.G. Zhadan, "Stable barrier-projection and barrier Newton methods in nonlinear programming", *Optimization Methods and Software*, **3**, 237–256 (1994).

[3] Y.G. Evtushenko and V.G. Zhadan, "Stable barrier-projection and barrier Newton methods in linear programming", *Computational Optimization and Applications*, **3**, 289–303 (1994).

[4] D. Feng and R.B. Schnabel, "Tensor Methods for Equality Constrained Optimization", *SIAM Journal on Optimization*, **6**, 653-673 (1996).

[5] R. Fletcher, *Practical Methods of Optimization, 2nd Ed.*, Wiley, Chichester (1987).

[6] C.L. Lawson and R.J. Hanson, *Solving Least Squares Problems*, SIAM, Philadelphia (1995).

[7] D. Luenberger, *Linear and nonlinear programming, 2nd Ed.*, Addison-Wesley, Reading, Mass (1984).

[8] D.Q. Mayne and E. Polak, "Feasible directions algorithms for optimization problems with equality and inequality constraints", *Mathematical Programming*, **11**, 67-80 (1976).

[9] R.J. Orsi, R.E. Mahony and J.B. Moore, "A dynamical systems analysis of semidefinite programming with application to quadratic optimization with pure equality constraints", *App. Math. Optm.*,**40**, 191–210 (1999).

[10] G.V. Smirnov, "Convergence of barrier-projection methods of optimization via vector Lyapunov functions", *Optimization Methods and Software*, **3**, 153-162, 1994.

[11] F. Verhulst, *Nonlinear Differential Equations and Dynamical Systems*, Springer–Verlag, Berlin (1990).

[12] S. Wang, K.L. Teo and H.W.J. Lee, "A new approach to nonlinear mixed discrete programming problems", *Eng. Opt.*, **30**, 249–262 (1998).