BIT 38:1 (1998), 127–150

A PENALTY CONTINUATION METHOD FOR THE ℓ_∞ SOLUTION OF OVERDETERMINED LINEAR SYSTEMS *

MUSTAFA C. PINAR[1] and SAMIR ELHEDHLI[2]

[1] *Department of Industrial Engineering, Bilkent University Ankara 06533, Turkey. email: mustafap@bilkent.edu.tr*

[2] *Graduate Program in Management, McGill University Montréal H3V1G3, Canada. email: elhedhls@management.mcgill.ca*

Abstract.

A new algorithm for the ℓ_∞ solution of overdetermined linear systems is given. The algorithm is based on the application of quadratic penalty functions to a primal linear programming formulation of the ℓ_∞ problem. The minimizers of the quadratic penalty function generate piecewise-linear non-interior paths to the set of ℓ_∞ solutions. It is shown that the entire set of ℓ_∞ solutions is obtained from the paths for sufficiently small values of a scalar parameter. This leads to a finite penalty/continuation algorithm for ℓ_∞ problems. The algorithm is implemented and extensively tested using random and function approximation problems. Comparisons with the Barrodale-Phillips simplex based algorithm and the more recent predictor-corrector primal-dual interior point algorithm are given. The results indicate that the new algorithm shows a promising performance on random (non-function approximation) problems.

AMS subject classification: 65K05, 65D10.

Key words: ℓ_∞ optimization, overdetermined linear systems, quadratic penalty functions, characterization.

1 Introduction.

The purpose of this paper is to give a new finite algorithm for the problem $[\ell_\infty]$

$$\min_{x\in R^n} \|Ax-b\|_\infty = \min_{x\in R^n} \max_{i=1..m} |a_i^T x - b_i|$$

where $A \in \mathcal{R}^{m\times n}$ is assumed to have rank n with no rows or columns identically zero, and $b \in \mathcal{R}^m$. It is well-known [19] that $[\ell_\infty]$ is equivalent to the following linear program

$$[\text{LINFLP}] \qquad \begin{array}{ll} \min & y \\ \text{s.t.} & Ax - ye \leq b \\ & Ax + ye \geq b, \end{array}$$

with the corresponding dual problem

* Received August 1996. Revised March 1997. Communicated by Kaj Madsen.

$$\text{[LINFLD]} \qquad \begin{array}{ll} \max & b^T(v-u) \\ \text{s.t.} & A^T(v-u)=0 \\ & e^T(u+v)=1 \\ & u,v \geq 0 \end{array}$$

where e is an m-vector with all components unity.

The problem has a wide range of applications, including time series, function approximation and data fitting analysis. As a result, efficient ways to solve it were the subject of many papers (see for example [18]). The first attempts seem to have been made by statisticians as the problem arises frequently in data fitting analysis. More efficient ways, however, were designed when it was realized that the problem is equivalent to a linear program, and, hence it can be solved via any linear programming method. The first numerically stable algorithm for the solution of the ℓ_∞ problem via the Stiefel exchange method was given by Bartels and Golub [1]. Barrodale and Phillips [2] designed a simplex method that exploits the special structure of the coefficient matrix. Alternatively, Bartels, Conn and Charalambous [3] used a direct nondifferentiable descent method. After Karmarkar's outstanding paper which started the area of interior-point methods, Ruzinsky and Olsen [17] used the same ideas to design a polynomial algorithm for the ℓ_∞ problem. Coleman and Li [6] used a formulation of the ℓ_∞ problem based upon the null space of A^T to propose a globally and quadratically convergent algorithm. Later, subsequent developments in the interior-point area led to the method of Zhang [19], where the predictor-corrector primal-dual interior point approach was adapted to the linear ℓ_∞ problems. Zhang's interior point algorithm also possesses local quadratic convergence properties under nondegeneracy assumptions.

The approach of the present paper is based upon recent ideas of Pınar [16] where a quadratic penalty function method was developed to solve a standard dual linear program with only inequality constraints. This was inspired by related work from Madsen and Nielsen [11] and Madsen, Nielsen and Pınar [13] where the ℓ_1 solution of overdetermined linear systems was studied and later applied to linear programming. However, the ideas of [11, 13] are based on a smoothing approximation of the ℓ_1 function, which is different from the subject of the present paper.

The new method consists of solving a quadratic penalty subproblem for smaller and smaller values of a scalar parameter. In theory, a solution to the original problem could be obtained from a solution to the unconstrained problem when the parameter tends to zero. However, the key to a stable and efficient algorithm is that the parameter tends to zero in a numerically well-defined manner. This is a consequence of the fact that there is a threshold value where optimal solutions to the original problem can be found from the solutions of the penalty problem by solving a linear system. This property is essential both for the efficiency and the numerical stability of the designed algorithm. The algorithm generates a sequence of non-interior iterates that satisfies primal feasibility only upon termination. The purpose of the present paper is to specialize the ideas of [16] to the ℓ_∞ problem. In particular, we describe the properties of the quadratic penalty function as applied to this problem. Although these results are obtained from the analysis of [16], mutatis mutandis, they are reiterated because (1) the ideas of [16] are fairly recent and not well-known, (2) we wish to make the paper self-contained. We redefine the penalty algorithm in the context of the linear ℓ_∞ problem and prove its finite convergence. The algorithm is implemented in a software system referred to as LINFSOL, extensively tested, and is compared with the Barrodale-Phillips simplex algorithm and the predictor-corrector interior point algorithm of Zhang.

Our algorithm is not the first penalty function algorithm to be proposed for ℓ_∞ problems. Joe and Bartels [9] and Bartels, Conn and Li [4] both used an exact nondifferentiable penalty function approach to solve the problem. Joe and Bartels use the dual formulation of the problem to apply the exact penalty function whereas Bartels, Conn and Li use a primal approach. Our ideas differ from the above in three important aspects:

1. We use a differentiable quadratic penalty function which was long forgotten due to potential numerical instabilities. To the contrary, we demonstrate the numerical stability and efficiency of this approach.

2. We utilize a finitely convergent Newton method to solve the penalty subproblems.

3. We exploit the piecewise linear dependence of the minimizers of the penalty function on the penalty parameter to devise a penalty continuation algorithm.

The rest of the paper is organized as follows. In Section 2 we will expose the fundamental properties of quadratic penalty functions applied to the ℓ_∞ problem. In Section 3 we will present the penalty continuation algorithm based on these properties. Section 4 is devoted to the finite convergence analysis. Finally, in Section 5 we give experimental results.

2 A quadratic penalty function approach.

Let us consider the following quadratic penalty function

$$F(x,y,t) = ty + \frac{1}{2}r_1^T(x,y)\Theta_1(x,y)r_1(x,y) + \frac{1}{2}r_2^T(x,y)\Theta_2(x,y)r_2(x,y),$$

where $r_1(x,y) = Ax - ye - b$ and $r_2(x,y) = Ax + ye - b$ and $\Theta_1(x,y)$ and $\Theta_2(x,y)$ are $m \times m$ diagonal matrices such that $\Theta_1 = diag(\theta_1)$, $\Theta_2 = diag(\theta_2)$ with

$$\theta_{1_{ii}}(x,y) = \begin{cases} 1 & \text{if } a_i^T x - y > b_i \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\theta_{2_{ii}}(x,y) = \begin{cases} 1 & \text{if } a_i^T x + y < b_i \\ 0 & \text{otherwise.} \end{cases}$$

We will be concerned with the unconstrained minimization problem:

[LINFCP]

$$\min_{x \in R^n, y \in R} F(x,y,t)$$

for decreasing values of t. It is well-known that the unconstrained minimization of $F(x,y,t)$ is well defined [5].

For ease of notation let z be the $n+1$ vector with $z_i = x_i$, $i = 1, \ldots, n$ and $z_{n+1} = y$, and denote by X the set of optimal vectors z to [LINFLP], and by M_t the set of minimizers of $F(x,y,t)$ for a fixed value of t. Let also $z_t = (x_t, y_t)$ denote a minimizer of $F(x,y,t)$.

2.1 Properties of F and its minimizers.

In this section, we give a characterization of the set of minimizers of F for fixed $t > 0$. It is obvious that $F(x,y,t)$ is composed of a finite number of quadratic functions. In each domain $\mathcal{D} \subseteq R^{n+1}$ where $\theta_1(x,y), \theta_2(x,y)$ are constant F is equal to a specific quadratic function. These domains are separated by the union of hyperplanes

$$\mathcal{B} = \{(x,y) \in R^{n+1};\ \exists i : \ a_i^T x - y - b_i = 0 \ \vee \ a_i^T x + y - b_i = 0\}.$$

So, for a given pair (x,y), the corresponding binary vectors $\theta_1(x,y), \theta_2(x,y)$ are found, and F is represented by $\mathcal{Q}_\theta$ on the subset,

$$\mathcal{C}_\theta = cl\{(\hat{x},\hat{y}) \in R^{n+1}; \theta_1(\hat{x},\hat{y}) = \theta_1 \ \wedge \ \theta_2(\hat{x},\hat{y}) = \theta_2\},$$

where $\mathcal{Q}_\theta$ is defined as follows:

$$\begin{aligned} \mathcal{Q}_\theta(\hat{x},\hat{y},t) \ &= F(x,y,t) + F_x^T(\hat{x}-x) + F_y^T(\hat{y}-y) \\ &+\tfrac{1}{2}(\hat{x}-x)^T F_{xx}(\hat{x}-x) + \tfrac{1}{2}(\hat{y}-y)^T F_{yy}(\hat{y}-y) \\ &+\tfrac{1}{2}(\hat{x}-x)^T F_{xy}(\hat{y}-y) + \tfrac{1}{2}(\hat{y}-y)^T F_{yx}(\hat{x}-x) \end{aligned}$$

with

$$\begin{aligned} F_y &\equiv \frac{\partial F(x,y,t)}{\partial y} = t + e^T(\Theta_1 + \Theta_2)ey + e^T(\Theta_2 - \Theta_1)Ax + e^T(\Theta_1 - \Theta_2)b, \\ F_x &\equiv \frac{\partial F(x,y,t)}{\partial x} = A^T(\Theta_2 - \Theta_1)ey + A^T(\Theta_1 + \Theta_2)Ax - A^T(\Theta_1 + \Theta_2)b, \\ F_{xy} &\equiv \frac{\partial^2 F(x,y,t)}{\partial x \partial y} = A^T(\Theta_2 - \Theta_1)e, \\ F_{yx} &\equiv \frac{\partial^2 F(x,y,t)}{\partial y \partial x} = e^T(\Theta_2 - \Theta_1)A, \\ F_{xx} &\equiv \frac{\partial^2 F(x,y,t)}{\partial x^2} = A^T(\Theta_1 + \Theta_2)A, \\ F_{yy} &\equiv \frac{\partial^2 F(x,y,t)}{\partial y^2} = e^T(\Theta_1 + \Theta_2)e. \end{aligned}$$

We refer to the gradient as the $(n+1)$ vector $\begin{bmatrix} F_x \\ F_y \end{bmatrix}$, and the Hessian as the $(n+1) \times (n+1)$ symmetric positive (semi)definite matrix $P = \begin{bmatrix} F_{xx} & F_{xy} \\ F_{yx} & F_{yy} \end{bmatrix}$ that can be written as $\bar{A}^T\Theta\bar{A}$, where $\bar{A} = \begin{bmatrix} -A & e \\ A & e \end{bmatrix}$ and $\Theta = \begin{bmatrix} \Theta_1 & 0 \\ 0 & \Theta_2 \end{bmatrix}$. We also denote by $\mathcal{N}(P)$ or $\mathcal{N}_\theta$ the null space of P. Let $\bar{b} = \begin{bmatrix} -b \\ b \end{bmatrix}$. Then the necessary condition for a minimizer z_t of F can be written compactly as

$$-te_{n+1} + \bar{A}^T\Theta\bar{b} = Pz_t \tag{2.1}$$

where e_{n+1} denotes an $n+1$ dimensional vector with a 1 at position $n+1$ and zero elsewhere. In the algorithm of Section 3 we do not form the matrices $\bar{A}$, Θ and P, neither the vector $\bar{b}$. These quantities are introduced here to facilitate notation in the analysis.

Now, consider the dual problem to [LINFCP]:

$$[\text{PD}] \qquad \begin{array}{ll} \max & b^T(v-u) - \frac{t}{2}(v^Tv + u^Tu) \\ \text{s.t.} & A^T(v-u) = 0 \\ & e^T(u+v) = 1 \\ & u, v \geq 0. \end{array}$$

As a consequence of strict concavity the above problem has a unique optimal solution and it can be shown that any minimizer (x_t, y_t) of F is related to the unique optimal solution (u_t, v_t) by the identities:

$$(u_t)_i = \frac{1}{t}\max\{0, r_{1i}(x_t, y_t)\}$$

$$(v_t)_i = -\frac{1}{t}\min\{0, r_{2i}(x_t, y_t)\}$$

for $i = 1, \ldots, m$. This implies the following result.

LEMMA 2.1. *$\theta_1(x_t, y_t)$ and $\theta_2(x_t, y_t)$ are constant for $(x_t, y_t) \in M_t$. Furthermore $a_i^Tx_t - y_t - b_i$ is constant for $\theta_{1_i} = 1$ and $a_i^Tx_t + y_t - b_i$ is constant for $\theta_{2_i} = 1$ $\forall i = 1, \ldots, m$.*

Following the lemma, the notation $\theta_1(M_t)$, $\theta_2(M_t)$ is used instead of $\theta_1(x_t, y_t)$, $\theta_2(x_t, y_t)$ for $(x_t, y_t) \in M_t$. The function $F(x, y, t)$ is convex, and hence the solution set M_t is convex. Now, let $\mathcal{J}_\theta = \{i|\theta_{1_i} = 0 \wedge \theta_{2_i} = 0\}$ and $\mathcal{D}_\theta = \{z = (x, y) \in R^{n+1}|a_i^Tx - y \leq b_i \wedge a_i^Tx + y \geq b_i \wedge i \in \mathcal{J}_\theta\}$. The following characterization of M_t can be obtained from the previous development.

COROLLARY 2.2. *Let $z_t \in M_t$ and $\theta_1 = \theta_1(M_t), \theta_2 = \theta_2(M_t)$. Then,*

$$M_t \equiv (z_t + \mathcal{N}_\theta) \cap \mathcal{D}_\theta.$$

Obviously, M_t is a singleton if $\mathcal{N}_\theta$ consists of the null vector.

2.2 Behavior of the minimizers as a function of t.

The purpose of this section is to show how the solution set M_t of $F(x, y, t)$ approximates the solution set X of [LINFLP] as t tends to zero. We begin with a basic lemma. For convenience of notation let $Q = \Theta\bar{A} = \Theta\begin{bmatrix} -A & e \\ A & e \end{bmatrix}$. Clearly, $P = Q^TQ$.

LEMMA 2.3. *Let $z_t \in M_t$ and $\theta_1 = \theta_1(M_t), \theta_2 = \theta_2(M_t)$. Then, the following system is consistent:*

$$Pd_z = e_{n+1}. \tag{2.2}$$

The proof can be obtained by setting the gradient of F at z_t equal to zero and reducing (2.2) into the normal equations corresponding to an overdetermined system.

Let d_z be a solution to (2.2). It is straightforward to verify that $z_t + td_z$ is the least squares solution to the overdetermined system of linear equations

$$Qh_z = \bar{b}. \tag{2.3}$$

To see this, it suffices to insert $Pd_z = e_{n+1}$ in $te_{n+1} + Pz_t = Q^T\bar{b}$ to get $tPd_z + Pz_t = Q^T\bar{b}$. This implies that $Q^TQ(z_t + td_z) = Q^T\bar{b}$.

LEMMA 2.4. *Let $z_t \in M_t$ and $\theta_1 = \theta_1(M_t), \theta_2 = \theta_2(M_t)$. If the overdetermined system (2.3) is consistent then*

$$\frac{1}{t}\Theta_1(Ax_t - y_te - b) = -\Theta_1(Ad_x - d_ye) \tag{2.4}$$

$$\frac{1}{t}\Theta_2(Ax_t + y_te - b) = -\Theta_2(Ad_x + d_ye) \tag{2.5}$$

for any solution $d_z =(d_x, d_y)$ to (2.2).

PROOF. We know that $z_t + td_z$ is the least squares solution to the overdetermined system of linear equations $Qh_z = \bar{b}$. If this system is consistent, $z_t + td_z$ solves $Qh_z = \bar{b}$. Therefore, we get

$$Q(z_t + td_z) = \bar{b} \Rightarrow \begin{bmatrix} -\Theta_1 A & \Theta_1 e \\ \Theta_2 A & \Theta_2 e \end{bmatrix} \begin{bmatrix} x_t + td_x \\ y_t + td_y \end{bmatrix} = \begin{bmatrix} -b \\ b \end{bmatrix}.$$

Premultiplying both sides by $\begin{bmatrix} \Theta_1 & \mathbf{0} \\ \mathbf{0} & \Theta_2 \end{bmatrix}$ and using the fact that $\Theta_k^2 = \Theta_k$, $k = 1, 2$, we get

$$\begin{bmatrix} -\Theta_1 A & \Theta_1 e \\ \Theta_2 A & \Theta_2 e \end{bmatrix} \begin{bmatrix} x_t + td_x \\ y_t + td_y \end{bmatrix} = \begin{bmatrix} -\Theta_1 b \\ \Theta_2 b \end{bmatrix}.$$

□

Next, let d_z solve (2.2) and assume that $\theta_1(x_t + \epsilon d_x, y_t + \epsilon d_y) = \theta_1$ and $\theta_2(x_t + \epsilon d_x, y_t + \epsilon d_y) = \theta_2$, i.e., $z_t + \epsilon d_z = (x_t + \epsilon d_x, y_t + \epsilon d_y) \in C_\theta$ for some $\epsilon > 0$. The linear nature of the problem implies that $z_t + \delta d_z = (x_t + \delta d_x, y_t + \delta d_y) \in C_\theta$ for $0 \leq \delta \leq \epsilon$. Therefore, using the necessary condition for a minimizer of F we get

$$\begin{aligned} -te_{n+1} + \bar{A}^T\Theta\bar{b} = Pz_t \quad &\Rightarrow \quad tPd_z + Pz_t = \bar{A}^T\Theta\bar{b} \\ &\Rightarrow \quad P(\delta d_z + z_t) = -(t - \delta)e_{n+1} + \bar{A}^T\Theta\bar{b}. \end{aligned}$$

Hence, $z_t + \delta d_z = (x_t + \delta d_x, y_t + \delta d_y)$ is a minimizer of $F(x, y, t - \delta)$. Using Corollary 2.2, we have the following result.

LEMMA 2.5. *Let $z_t \in M_t$ and $\theta_1 = \theta_1(M_t), \theta_2 = \theta_2(M_t)$. Let d_z solve* (2.2). *If $\theta_1(x_t + \epsilon d_x, y_t + \epsilon d_y) = \theta_1$ and $\theta_2(x_t + \epsilon d_x, y_t + \epsilon d_y) = \theta_2$ for $\epsilon > 0$ then $\theta_1(x_t + \delta d_x, y_t + \delta d_y) = \theta_1$ and $\theta_2(x_t + \delta d_x, y_t + \delta d_y) = \theta_2$ and*

$$M_{t-\delta} = (z_t + \delta d_z + \mathcal{N}_\theta) \cap \mathcal{D}_\theta,$$

for $0 \leq \delta \leq \epsilon$.

Although t is a continuous parameter, there is only a finite number of binary vectors θ_1 and θ_2. Furthermore, the previous lemma ensures that whenever there exists t_1, t_2 where $\theta_1(x_{t_1}, y_{t_1}) = \theta_1(x_{t_2}, y_{t_2})$ and $\theta_2(x_{t_1}, y_{t_1}) = \theta_2(x_{t_2}, y_{t_2})$ we have $\theta_1(x_t, y_t) = \theta_1(x_{t_1}, y_{t_1})$ and $\theta_2(x_t, y_t) = \theta_2(x_{t_1}, y_{t_1})$ for all $t \in [t_1, t_2]$. As a consequence, $\theta_1(M_t)$ and $\theta_2(M_t)$ are piecewise constant functions of t. Hence, as t tends to 0, θ_1 and θ_2 should remain constant in a neighborhood of 0. That is, there exists a positive value of t, say t_0, such that θ_1 and θ_2 remain constant for $0 < t \leq t_0$. Therefore we have the following result.

THEOREM 2.6. *There exists $t_0 > 0$ such that $\theta_1(M_t)$ and $\theta_2(M_t)$ are constants for $0 < t \leq t_0$. Furthermore, if $\theta_1(z_t + \delta d_z) = \theta_1(M_t)$ and $\theta_2(z_t + \delta d_z) = \theta_2(M_t)$ then*

$$M_{t-\delta} = (z_t + \delta d_z + \mathcal{N}_\theta) \cap \mathcal{D}_\theta,$$

for $0 \leq \delta < t \leq t_0$, where d_z solves (2.2).

Now, for $z_t \in M_t$ and $\theta_1 = \theta_1(M_t), \theta_2 = \theta_2(M_t)$, let us define

$$u_t = \frac{1}{t}\Theta_1 r_1(x_t, y_t), \quad \text{and} \quad v_t = -\frac{1}{t}\Theta_2 r_2(x_t, y_t). \tag{2.6}$$

Recalling the necessary optimality conditions for a minimizer, we have $t - e^T\Theta_1 r_1(x_t, y_t) + e^T\Theta_2 r_2(x_t, y_t) = 0$, which implies that $e^T(u_t + v_t) = 1$, and $A^T\Theta_1 r_1(x_t, y_t) + A^T\Theta_2 r_2(x_t, y_t) = 0$ implies that $A^T(v_t - u_t) = 0$. Furthermore, (u_t, v_t) has all components nonnegative. Therefore, (u_t, v_t) is feasible for [LINFLD]. Now, we give the following important corollary to the previous theorem.

COROLLARY 2.7. *Let $0 < t \leq t_0$ where t_0 is given in Theorem* 2.6 *and let $\theta_1 = \theta_1(M_t)$, $\theta_2 = \theta_2(M_t)$. Then*

$$\Theta_1 r_1(x_t + t\tilde{d}_x, y_t + t\tilde{d}_y) = 0, \tag{2.7}$$

$$\Theta_2 r_2(x_t + t\tilde{d}_x, y_t + t\tilde{d}_y) = 0, \tag{2.8}$$

where $\tilde{d}_z = \begin{bmatrix} \tilde{d}_x \\ \tilde{d}_y \end{bmatrix}$ *is any solution to* (2.2). *Furthermore,*

$$\frac{1}{t}\Theta_1 r_1(x_t, y_t) = -\Theta_1(A\tilde{d}_x - \tilde{d}_y e), \tag{2.9}$$

$$\frac{1}{t}\Theta_2 r_2(x_t, y_t) = -\Theta_2(A\tilde{d}_x + \tilde{d}_y e), \tag{2.10}$$

i.e, $\Theta_1 r_1(x_t, y_t)/t$ and $\Theta_2 r_2(x_t, y_t)/t$ are constants.

PROOF. Let $z_{t-\delta} \in M_{t-\delta}$ for $0 \leq \delta < t$. By Theorem 2.6, there exists d_z that solves (2.2) such that $z_{t-\delta} = z_t + \delta d_z$. Hence, there exists d_z^* that solves (2.2) such that $z_t + \delta d_z^* \in M_{t-\delta}$ for all $0 \leq \delta < t$. Now, it is well-known (see e.g., [8]) that

$$\lim_{t \to 0} \frac{1}{2} r_1^T(x_t, y_t)\Theta_1 r_1(x_t, y_t) + \frac{1}{2} r_2^T(x_t, y_t)\Theta_2 r_2(x_t, y_t) = 0.$$

Therefore, we get

$$\begin{aligned} \Theta_1 r_1(x_t + td_x^*, y_t + td_y^*) &= 0, \\ \Theta_2 r_2(x_t + td_x^*, y_t + td_y^*) &= 0. \end{aligned}$$

Any solution $\tilde{d}_z$ of (2.2) can be expressed as $\tilde{d}_z = d_z^* + \eta_z$ where $\eta_z = \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} \in \mathcal{N}(P)$. However, we have

$$P\eta_z = 0 \Rightarrow Q\eta_z = 0 \Rightarrow \begin{bmatrix} -\Theta_1 A & \Theta_1 e \\ \Theta_2 A & \Theta_2 e \end{bmatrix} \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} = 0.$$

Inserting this in the above two equations, we get equalities (2.7) and (2.8). Equalities (2.9) and (2.10) follow from Lemma 2.4 since (2.7) and (2.8) imply that (2.3) is consistent. □

The corollary indicates that the residuals corresponding to the violated inequalities approach their optimal values in a well-conditioned manner as t decreases to zero. As we shall see in Section 3, the behaviour of violated residuals is the driving force of the penalty continuation method. Now, we are in a position to give a new characterization of the ℓ_∞ solutions.

THEOREM 2.8. *Let $0 < t \leq t_0$, where t_0 is as given in Theorem 2.6 and let $\theta_1 = \theta_1(M_t)$, $\theta_2 = \theta_2(M_t)$. Let $z_t \in M_t$ and d_z solve* (2.2)*.Then*

$$X \equiv M_0,$$

where

$$M_0 = (z_t + td_z + \mathcal{N}_\theta) \cap \mathcal{D}_\theta,$$

and

$$u^* = \frac{1}{t}\Theta_1 r_1(x_t, y_t) \quad ; \quad v^* = -\frac{1}{t}\Theta_2 r_2(x_t, y_t)$$

solve [LINFLD].

PROOF. Let $z_0 \in M_0$. Then there exists a solution d_z^0 to (2.2) such that $z_0 = z_t + td_z^0$. By Corollary 2.7, we have $\Theta_1 r_1(x_0, y_0) = 0$ and $\Theta_2 r_2(x_0, y_0) = 0$. Now, using the fact that (u^*, v^*) is dual feasible (i.e., $e^T(u^* + v^*) = 1$ and $A^T(v^* - u^*) = 0$) we get:

$$\begin{aligned} y_0 &= y_0 + 0x_0 \\ &= y_0^T e^T(u^* + v^*) + x_0^T A^T(v^* - u^*) \\ &= (-x_0^T A^T + e^T y_0^T + b^T)u^* + (x_0^T A^T + e^T y_0^T - b^T)v^* + b^T(v^* - u^*) \\ &= -\tfrac{1}{t} r_1^T(x_0, y_0)\Theta_1 r_1(x_t, y_t) - \tfrac{1}{t} r_2^T(x_0, y_0)\Theta_2 r_2(x_t, y_t) + b^T(v^* - u^*) \\ &= b^T(v^* - u^*). \end{aligned}$$

This shows that z_0 and (u^*, v^*) are optimal in their respective problems. Since this holds for any $z_0 \in M_0$, $M_0 \subseteq X$.

If M_0 is a singleton, then the proof is complete. Assume otherwise and let $z = (x, y) \in X$. By feasibility, we have $Ax - ye \leq b$ and $Ax + ye \geq b$. Furthermore, (x, y) is complementary to (u^*, v^*) which implies that $\Theta_1 r_1(x, y) = 0$ and $\Theta_2 r_2(x, y) = 0$. This further implies

$$P\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -A^T\Theta_1 & A^T\Theta_2 \\ \Theta_1 e & \Theta_2 e \end{bmatrix} \begin{bmatrix} -b \\ b \end{bmatrix}.$$

Now, using the above and the necessary condition (2.1) for a minimizer of F (z_t minimizes $F(x, y, t)$) we have

$$\begin{aligned} &P(z - z_t) \\ &= te_{n+1} - \begin{bmatrix} -A^T\Theta_1 & A^T\Theta_2 \\ \Theta_1 e & \Theta_2 e \end{bmatrix} \begin{bmatrix} -b \\ b \end{bmatrix} + \begin{bmatrix} -A^T\Theta_1 & A^T\Theta_2 \\ \Theta_1 e & \Theta_2 e \end{bmatrix} \begin{bmatrix} -b \\ b \end{bmatrix} = te_{n+1} \end{aligned}$$

Thus $(z - z_t)/t$ is a solution to (2.2). We have shown that $z \in z_t + td_z + \mathcal{N}_\theta$. Now observing that $z \in \mathcal{D}_\theta$ by virtue of feasibility the proof is completed. □

Clearly, X is a singleton if $\mathcal{N}_\theta$ consists of the null vector.

2.3 Extended binary vectors.

In this section, we give some further results that are useful in the design of the algorithm and its finite convergence analysis. We define the following "extended binary vector",

$$\bar{\theta}_{1_i}(x, y) = \begin{cases} 1 & \text{if } a_i^T x - y \geq b_i \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \bar{\theta}_{2_i}(x, y) = \begin{cases} 1 & \text{if } a_i^T x + y \leq b_i \\ 0 & \text{otherwise.} \end{cases}$$

We also define the "active set" of indices

$$\mathcal{A}(x, y) = \{i | \bar{\theta}_{1_i}(x, y) = 1\} \cup \{i | \bar{\theta}_{2_i}(x, y) = 1\}.$$

Extended binary vectors differ from the binary vectors used thus far only for those points that belong to $\mathcal{B}$. However, this difference leads to an enlargement of the set of violated inequalities, and brings the matrix P closer to having full rank. This enlargement is instrumental in the finiteness proof of the modified Newton used to solve the penalty subproblems. This is discussed in Section 3.1.

Now, we analyze the behaviour of the set of extended binary vectors associated with the set of minimizers of $F(x, y, t)$ in the range $(0, t_0]$ where t_0 is as defined in Theorem 2.6. This is important in establishing the finite termination property of the penalty algorithm defined in Section 3.

Denote by $\mathcal{S}(M_t)$ the set of all distinct extended binary pairs of vectors corresponding to the elements of M_t. In other words, for any $(x_t, y_t) \in M_t$, $(\bar{\theta}_1(x_t, y_t), \bar{\theta}_2(x_t, y_t)) \in \mathcal{S}(M_t)$. We have the following result which is a consequence of linearity.

LEMMA 2.9. *If* $\mathcal{S}(M_{t_1}) = \mathcal{S}(M_{t_2})$ *where* $0 < t_2 < t_1$ *then* $\mathcal{S}(M_{t_1}) = \mathcal{S}(M_{t_2}) = \mathcal{S}(M_t)$ *for* $t_2 \leq t \leq t_1$.

Now, as $\theta_1(M_t)$ and $\theta_2(M_t)$ remain constant in $(0, t_0]$, and the number of extended binary vectors is finite, the previous lemma ensures that $\mathcal{S}(M_t)$ cannot keep changing infinitely as t approaches 0. Hence, we have the following theorem.

THEOREM 2.10. *There exists* $\bar{t}$ *such that* $\mathcal{S}(M_t)$ *is constant for* $t \in (0, \bar{t})$ *where* $0 < \bar{t} \leq t_0$.

For $t \in (0, \bar{t})$ let $(x_t, y_t) \in M_t$ with $\bar{\theta}_1 = \bar{\theta}_1(x_t, y_t)$ and $\bar{\theta}_2 = \bar{\theta}_2(x_t, y_t)$. Consider the system:

$$\bar{P} d_z = e_{n+1} \tag{2.11}$$

where

$$\bar{P} = \begin{bmatrix} A^T\bar{\Theta}_1 A + A^T\bar{\Theta}_2 A & -A^T\bar{\Theta}_1 e + A^T\bar{\Theta}_2 e \\ -e^T\bar{\Theta}_1 A + e^T\bar{\Theta}_2 A & e^T\bar{\Theta}_1 e + e^T\bar{\Theta}_2 e \end{bmatrix}.$$

This is a consistent system of linear equations as shown in Lemma 2.3. By Theorem 2.10 there exists $(x_t, y_t) \in M_t$ such that $\bar{\theta}_1(x_t, y_t) = \bar{\theta}_1$ and $\bar{\theta}_2(x_t, y_t) = \bar{\theta}_2$ for all $t \in (0, \bar{t})$. This implies that there exists d_z that solves (2.11) such that $z_t + \delta d_z \in M_{t-\delta}$ for all $\delta \in [0, t)$. Now, as t approaches 0, it is well-known (e.g., [8]) that both $\bar{\Theta}_1 r_1(x_t, y_t)$ and $\bar{\Theta}_2 r_2(x_t, y_t)$ tend to zero. Therefore, we have

$$\bar{\Theta}_1 r_1(x_t + td_x, y_t + td_y) = 0,$$

$$\bar{\Theta}_2 r_2(x_t + td_x, y_t + td_y) = 0.$$

By continuity of r_1 and r_2, we have that $r_{1i}(x_t + td_x, y_t + td_y) < 0$ for all indices i such that $\bar{\theta}_{1i} = 0$ and $r_{2i}(x_t + td_x, y_t + td_y) > 0$ for all indices i such that $\bar{\theta}_{2i} = 0$. This implies that $(x_t + td_x, y_t + td_y)$ is a feasible point for [LINFLP]. Let $u^* = \frac{1}{t}\bar{\Theta}_1 r_1(x_t, y_t)$ and $v^* = -\frac{1}{t}\bar{\Theta}_2 r_2(x_t, y_t)$. Since (u^*, v^*) is a feasible solution of [LINFLD] and $(x_t + td_x, y_t + td_y)$ is complementary to (u^*, v^*) and is feasible in [LINFLP], it follows that $(x_t + td_x, y_t + td_y)$ and (u^*, v^*) solve [LINFLP] and [LINFLD], respectively. Clearly, if the solution to (2.11) is unique, $d_z^* = (d_x^*, d_y^*)$ say, then $(x_t + td_x^*, y_t + td_y^*)$ solves [LINFLP]. Since d_z can be replaced by $d_z + \eta_z = (d_x + \eta_x, d_y + \eta_y)$ where $\eta_z \in \mathcal{N}(\bar{P})$, it follows that

$$\bar{\Theta}_1 r_1(x_t + td_x, y_t + td_y) = 0,$$

$$\bar{\Theta}_2 r_2(x_t + td_x, y_t + td_y) = 0$$

for any solution d_z of (2.11). Thus we have the following theorem.

THEOREM 2.11. *Let* $t \in (0, \bar{t})$ *and* $(x_t, y_t) \in M_t$ *with* $\bar{\theta}_1 = \bar{\theta}_1(M_t)$ *and* $\bar{\theta}_2 = \bar{\theta}_2(M_t)$. *Also, let* $u^* = \frac{1}{t}\bar{\Theta}_1 r_1(x_t, y_t)$ *and* $v^* = -\frac{1}{t}\bar{\Theta}_2 r_2(x_t, y_t)$. *Then,*

$$\bar{\Theta}_1 r_1(x_t + td_x, y_t + td_y) = 0, \tag{2.12}$$

$$\bar{\Theta}_2 r_2(x_t + td_x, y_t + td_y) = 0 \tag{2.13}$$

for any solution $d_z = (d_x, d_y)$ *to* (2.2). *Furthermore, if* d_z *is unique or* $r_1(x_t + td_x, y_t + td_y) \leq 0$ *and* $r_2(x_t + td_x, y_t + td_y) \geq 0$ *then* $(x_t + td_x, y_t + td_y)$ *solves* [LINFLP].

3 The penalty algorithm.

Based on the analysis given in the previous sections, a penalty algorithm for ℓ_∞ problems is designed. The algorithmic ideas are mainly motivated by the results of Theorem 2.11 and Lemma 2.5. For t sufficiently small ($t \in (0, \bar{t})$), the point $z_t + td_z$ and the dual pair (u^*, v^*) as defined in Theorem 2.11 form a partially complementary and partially feasible pair regardless of the choice of d_z (i.e., they satisfy (2.12)–(2.13)). Therefore, if $z_t + td_z$ is feasible, $(z_t + td_z, (u^*, v^*))$ is a primal-dual optimal pair. If the partial complementarity and feasibility conditions as stated in Theorem 2.11 hold but $z_t + td_z$ is not feasible, we move to the smallest positive breakpoint along d_z. If these conditions do not hold, this is an indication that we are far from the solution. In this case, inspired by Lemma 2.5 one can find the largest δ for which $\bar{\theta}_1(x_t + \delta d_x, y_t + \delta d_y) = \bar{\theta}_1(x_t, y_t)$ and $\bar{\theta}_2(x_t + \delta d_x, y_t + \delta d_y) = \bar{\theta}_2(x_t, y_t)$. This property provides a sound criterion to reduce t. More precisely, we propose the following algorithm.

```
Choose $t$ and compute a minimizer $z_t$ of $F$
while not STOP
    reduce $t$
    compute a minimizer $z_t$ of $F$
end while.
```

Here, STOP is a function that returns TRUE if conditions (2.12)–(2.13) hold and primal feasibility is achieved (dual feasibility is always maintained at a minimizer of z_t of F). To complete the description we need an algorithm to compute a minimizer of F, and a procedure to reduce t.

3.1 Computing a minimizer of F.

The algorithm for computing a minimizer z_t of F is adapted from robust linear regression using Huber functions [10]. It is a standard Newton iteration with a simple line search to solve the nonlinear system of equations $F_x(x, y, t) = 0$ and $F_y(x, y, t) = 0$. The idea is to inspect the regions of R^{n+1} to locate the region where the local quadratic Q_θ contains its own minimizer. At a given iterate, the Newton step is computed using the quadratic expansion of F. If a unit step in this direction yields a point in the same region, then the global minimizer has been found. That is to say that the quadratic representation of F which contains the global minimizer has been found. Otherwise, the algorithm proceeds with a line search.

A search direction h_z is computed by minimizing the quadratic $Q_{\bar{\theta}}$ where $\bar{\theta} = \bar{\theta}(x, y)$ and $z = (x, y)$ is the current iterate. This yields the linear system

$$\bar{P}h_z = -te_{n+1} - \bar{P}z + \bar{A}^T \bar{\Theta} \bar{b}. \tag{3.1}$$

Denote the right-hand side vector in equation (3.1) by g, so that we have $\bar{P}h_z = g$. If $\bar{P}$ has full rank then h_z is the unique solution to (3.1). Otherwise, if the system (3.1) is consistent, a minimum norm solution is computed and used as a search direction. If the system is inconsistent, the projection of g onto $\mathcal{N}(\bar{P})$

is computed to serve as a descent direction. These choices are motivated and justified in [10]. After a finite number of iterations (this can be shown using the analysis in [10]) we have $z + h_z \in \mathcal{C}_{\bar{\theta}}$. Therefore, $z + h_z$ is a minimizer of F. The modified Newton algorithm is summarized below.

```
repeat
    $\bar{\theta}_1 = \bar{\theta}_1(x,y)$; $\bar{\theta}_2 = \bar{\theta}_2(x,y)$
    if (3.1) is consistent then
        find $h_z$ from (3.1)
        if $z + h_z \in \mathcal{C}_{\bar{\theta}}$ then
            $z \leftarrow z + h_z$
            stop = true
        else
            $z \leftarrow z + \alpha h_z$ (line search)
        endif
    else
        compute $h_z$ = null space projection of $g$
        $z \leftarrow z + \alpha h_z$ (line search)
    endif
until stop.
```

3.2 Reducing t.

Let z_t be a minimizer of $F(x,y,t)$ for some $t > 0$ and let $\bar{\theta}_1 = \bar{\theta}_1(x_t, y_t)$ and $\bar{\theta}_2 = \bar{\theta}_2(x_t, y_t)$. Furthermore, let d_z be a solution to (2.11). Two cases arise:

Case 1 In this case,

$$\bar{\Theta}_1 r_1(x_t + td_x, y_t + td_y) = 0,$$

and

$$\bar{\Theta}_2 r_2(x_t + td_x, y_t + td_y) = 0$$

but $(x_t + td_x, y_t + td_y)$ is infeasible in [LINFLP], i.e., there exists j such that either $r_{1_j}(x_t + td_x, y_t + td_y) > 0$ or $r_{2_j}(x_t + td_x, y_t + td_y) < 0$. Let $\psi_1 \equiv \{\alpha_k, k = 1, 2, \ldots, m_1\}$ and $\psi_2 \equiv \{\beta_k, k = 1, 2, \ldots, m_2\}$ be the sets of positive kink points where the components of $r_1(x_t + td_x, y_t + td_y)$ and $r_2(x_t + td_x, y_t + td_y)$ change sign, respectively. We choose $\alpha^* = \min(\min_k \alpha_k, \min_k \beta_k)$, and $t_{next} = (1 - \alpha^*)t$. We let

$$x_{t_{next}} \equiv x_t + \alpha^* td_x; \quad y_{t_{next}} \equiv y_t + \alpha^* td_y.$$

Then, $(x_{t_{next}}, y_{t_{next}})$ is used as the starting point of the modified Newton algorithm with the reduced value of t.

Case 2

$$\bar{\Theta}_1 r_1(x_t + td_x, y_t + td_y) \neq 0,$$

and/or

$$\bar{\Theta}_2 r_2(x_t + td_x, y_t + td_y) \neq 0.$$

In this case we reduce t as follows. Let $\Delta((1-\epsilon)t)$ denote the number of changes from $\mathcal{A}(z_t)$ to $\mathcal{A}(z_t+\epsilon t d_z)$. Then, bisection is used to find $\bar{\epsilon}$ such that $\Delta((1-\bar{\epsilon})t) \approx \frac{1}{2}\Delta(t)$. Then,

$$t_{next} = (1-\bar{\epsilon})t, \qquad z_{t_{next}} \equiv z_t + \bar{\epsilon} t d_z.$$

As in Case 1, $(x_{t_{next}}, y_{t_{next}})$ is used as the starting point of the modified Newton algorithm with the reduced value t_{next} of t.

4 Finite convergence.

In this section, the algorithm is shown to converge finitely. In the analysis an iteration of the algorithm means either a modified Newton iteration or an execution of the t-reduction procedure.

Now, let $(x,y) \in M_t$ and $u = \frac{1}{t}\bar{\Theta}_1 r_1(x,y)$ and $v = -\frac{1}{t}\bar{\Theta}_2 r_2(x,y)$. From Theorem 2.11, conditions (2.12)–(2.13) hold at $(x+td_x, y+td_y)$ where (d_x, d_y) is any solution to (2.11). Following the reduction procedure, either $A(x+td_x) - (y+td_y)e \leq b$ and $A(x+td_x)+(y+td_y)e \geq b$ and thus $z^+ = z+td_z$ is a solution to [LINFLP] and the algorithm stops, or $\mathcal{A}(z) \subset \mathcal{A}(z^+)$. The latter condition follows directly from the choice of α^* in Case 1 of the reduction procedure. In addition to this, it guarantees that $z + \alpha^* t d_z \in C_{\bar{\theta}}$. Therefore, using the definition of the gradient and the way d_z is calculated, we have $z + \alpha^* t d_z \in M_{(1-\alpha^*)t}$. Therefore, we have proved the following lemma.

LEMMA 4.1. *Assume $t \in (0, \bar{t})$. Let $z = (x,y) \in M_t$ with $\bar{\theta}_1 = \bar{\theta}_1(x,y)$ and $\bar{\theta}_2 = \bar{\theta}_2(x,y)$. Let d_z solve (2.2), and $z^+ = (x^+, y^+)$ be generated by one iteration of the algorithm. Then either*

$$z^+ \equiv z + td_z \in X$$

and the algorithm stops, or

$$z^+ \equiv z + \alpha^* t d_z \in M_{t^+},$$

$$t^+ = (1-\alpha^*)t$$

where α^* is as defined in Case 1 of the reduction procedure, and $\mathcal{A}(z^+)$ is an extension of $\mathcal{A}(z)$.

Let $z \in M_t$ for some $t > 0$. Unless the stopping criteria are met and the algorithm stops with a primal-dual optimal pair, t is reduced by at least a factor of κ where $\kappa \in (0\,,\,1)$ as discussed in the reduction procedure. Since the modified Newton iteration is a finite process, t will reach the range $(0,\,\bar{t})$ where $\bar{t}$ is as defined in Theorem 2.10 in a finite number of iterations. Now assume $t \in (0,\,\bar{t})$. From Lemma 4.1 either the algorithm terminates or the active set $\mathcal{A}$ is expanded. Repeating this argument, in a finite number of iterations the matrix $\bar{P}$ will finally have rank $n+1$ since A has rank n. When $\bar{P}$ has full rank the solution d_z to the system (2.2) is unique, and $z^+ = z + td_z$ solves [LINFLP] by Theorem 2.11. Therefore we have proved the following theorem.

THEOREM 4.2. *The algorithm terminates in a finite number of iterations with a primal-dual optimal pair.*

5 Implementation and testing.

In this section, we provide implementation details and numerical experience with the algorithm of Section 3. The first aim is to test the viability of the approach, the behaviour of the algorithm and the numerical accuracy of the method. Second, we wish to test the potential of the algorithm as a competitor to the well-known algorithms for the ℓ_∞ problem. This involves comparisons with the Barrodale-Phillips implementation of the simplex algorithm [2], and the interior point algorithm of Zhang [19]. The Barrodale-Phillips implementation of the simplex method is accepted to be one of the best codes available for the linear ℓ_∞ problem according to Bartels, Conn and Li [4].

5.1 Implementation.

We refer to the penalty continuation code as LINFSOL[1]. The implementation was done in Fortran 77 on a SUN SPARC 4 workstation. The major work in the Newton algorithm is dominated by the requirement to solve the least squares systems of the form $\bar{A}^T\bar{\Theta}\bar{A}\,x = g$, more precisely the system (3.1). To check optimality and reduce t, system (2.11) is solved. Both linear systems have dimensions $(n+1)\times(n+1)$. The AAFAC package based on the work of Nielsen [15] is used to solve these linear systems. In AAFAC the solution is obtained via an LDL^T factorization of the matrix $P_k = \bar{A}^T\bar{\Theta}\bar{A}$. Let us recall that $\bar{\Theta}_{ii} = 1$ for the columns of A corresponding to indices in the active set $\mathcal{A}$. Based on this observation, D and L are computed directly from the active columns of $\bar{A}$, i.e., without squaring the condition number as would be the case if P_k was first computed. The efficiency of the penalty algorithm hinges on the following observation. Normally, only a few entries of the diagonal matrix $\bar{\Theta}$ change between two consecutive iterations of the modified Newton method. Therefore, the factorization of P_k can be obtained by relatively few up- and down-dates of the factorization of P_{k-1}. Occasionally, a refactorization is performed. This consists in the successive updating $LDL^T \leftarrow LDL^T + a_j a_j^T$ for all j in the active set (starting with $L = I, D = 0$). It is considered only when some columns of $\bar{A}$ leave the active set, i.e., when downdating is involved. If many columns are involved in the change from P_{k-1} to P_k it may be cheaper to refactorize P_k. Otherwise, a refactorization is used when a downdating results in a rank decrease and there is an indication that rounding errors have marked influence.

When the system (2.11) is solved after the modified Newton algorithm, the LDL^T factors are available from the previous Newton step. The details of these numerical linear algebraic tasks can be found in [15].

For efficiency in work and storage, we keep only one copy of A and b. For the computation of residuals r_1 and r_2, the matrix-vector product Ax is formed only once and used for both vectors. A similar practice is adopted for the update (or, downdate) of the factorization.

The stopping criteria are based on checking conditions (2.12)–(2.13) and satisfying primal feasibility. The feasibility tolerance used is computed as $m\|b\|_\infty\epsilon_M$,

[1] LINFSOL is available for distribution as a standard Fortran 77 subroutine. It can be obtained from the authors on request.

where ϵ_M denotes the computer's unit roundoff.

As regards competing algorithms, we have used the original implementation of the Barrodale-Phillips algorithm available in the NAG Subroutine Library [14]. Since Fortran implementations of the interior point algorithms for the ℓ_∞ problem were not available, we have also developed an efficient Fortran 77 implementation of the predictor-corrector primal-dual algorithm of Zhang based on reference [19] and his Matlab code. We have made extensive use of BLAS routines [7]. In [19] Zhang uses $x^0 = 0$ as the starting point for his algorithm. We have adopted the starting point $x^0 = (A^TA)^{-1}A^Tb$ as this leads to a twofold improvement in the number of iterations of the interior point method in some cases.

Unless othewise stated, all runs were performed on a SUN SPARC 1000E workstation.

5.2 Test problems.

Two kinds of test problems are used. The first type is randomly generated problems obtained from a problem generator. For this purpose, an ℓ_∞ problem generator is designed, which provides problems with a previously known solution. The idea is based on linear programming theory. For given dimensions m and n, appropriate vectors A, x, y, u, v and b are suitably chosen to satisfy the Karush-Kuhn-Tucker optimality conditions for the ℓ_∞ problem. The entries of the matrix A, the vector x and the scalar y are chosen from a uniform distribution. The entries of u and v, however, are chosen to satisfy dual feasibility, and b is selected so that complementary slackness holds. The generator can provide nondegenerate, primal degenerate and dual degenerate problems. Problems with exactly $n+1$ residuals where the maximum is acheived are nondegenerate. Primal degeneracy is forced through the choice of an additional number of *pdeg* equations to be satisfied as equality at the optimal solution so that at optimality there are $n+1+pdeg$ equations among the $2m$ that are binding. Dual degeneracy, however, is achieved when *ddeg* additional dual variables are chosen to be zero so that in total there are $n+1-ddeg$ nonzero optimal dual multipliers.

The second type of test problems is function approximation problems in the ℓ_∞ norm. The problem is to estimate a certain function $f(x)$ by a polynomial of degree $n-1$ on a set of m evenly spaced points over an interval $[\xi_1, \xi_2]$ of length ξ. The estimation is done through the determination of the coefficients of the polynomial. Explicitly, we consider the problem of computing the n unknowns $x_1, \ldots, x_n$ in the system

$$f(\mu) = \sum_{j=1}^{n} x_j \mu^{j-1}, \quad \mu = 0, \frac{\xi_2 - \xi_1}{m}, \ldots, \xi_2 - \xi_1.$$

Obviously, $b_i = f(i\xi/m)$, for $i = 1, \ldots, m$ and $a_{ij} = (i\xi/m)^{j-1}$ for $i = 1, \ldots, m$, and $j = 1, \ldots, n$. These problems tend to be increasingly ill-conditioned as m and n are increased.

5.3 *Initialization strategies.*

An important determinant of performance of the penalty algorithm is the initialization strategy. This consists in the choice of suitable initial values x^0, y^0 and t^0 of x, y and t. After considerable preliminary experimentation with several alternatives, we have set aside the following strategies as the most promising candidates:

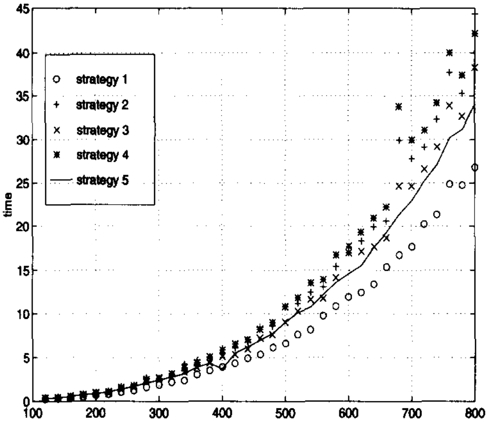

Figure 5.1: Performance of initialization strategies for random nondegenerate problems ($m/n = 4$).

STRATEGY 1

$$x^0 = 2(A^TA)^{-1}A^Tb, \quad y^0 = MINRES(m/2, Ax^0 - b), \qquad t^0 = 0.01 \times n \times y^0$$

STRATEGY 2

$$x^0 = (A^TA)^{-1}A^Tb, \quad y^0 = MINRES(m/2, Ax^0 - b), \qquad t^0 = 0.1 \times n \times y^0$$

STRATEGY 3

$$x^0 = (A^TA)^{-1}A^Tb, \quad y^0 = MINRES(m/2, Ax^0 - b), \qquad t^0 = 0.01 \times n \times y^0$$

STRATEGY 4

$$x^0 = (A^TA)^{-1}A^Tb, \quad y^0 = MINRES(m/4, Ax^0 - b), \qquad t^0 = 0.1 \times n \times y^0$$

STRATEGY 5

$$x^0 = 2(A^TA)^{-1}A^Tb, \quad y^0 = MINRES(m/2, Ax^0 - b), \qquad t^0 = 0.1 \times n \times y^0$$

where $MINRES(k, r)$ returns the kth smallest entry of r in absolute value. In Figure 5.1, we illustrate the performance of LINFSOL under these five competing initialization strategies using randomly generated nondegenerate test problems with the ratio $m/n = 4$. The CPU time is given in seconds.

We observe that all five strategies perform similarly with a slight dominance of Strategy 1. As this test gave similar results for primal and dual degenerate problems we adopted Strategy 1 as our default initialization strategy for the randomly generated problems.

On the other hand, the performance of the above initialization strategies on the function approximation problems were somewhat inferior compared to the case of randomly generated problems. This is essentially due to the following observation. Although the optimal solution in these problems is nondegenerate the largest nonzero residuals at the solution can be as small as 10^{-8}. We refer to this property as near degeneracy. The performance of LINFSOL is affected negatively by this property as supported by the following theorem.

THEOREM 5.1. *Let* (x^*, y^*) *be an optimal point for* [LINFLP]. *Let*

$$\tilde{\theta}_{1_i}(x^*, y^*) = \begin{cases} 1 & \text{if } a_i^T x^* - y^* = b_i \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\tilde{\theta}_{2_i}(x^*, y^*) = \begin{cases} 1 & \text{if } a_i^T x^* + y^* = b_i \\ 0 & \text{otherwise.} \end{cases}$$

Define

$$\tilde{P} = \begin{bmatrix} A^T\tilde{\Theta}_1 A + A^T\tilde{\Theta}_2 A & -A^T\tilde{\Theta}_1 e + A^T\tilde{\Theta}_2 e \\ -e^T\tilde{\Theta}_1 A + e^T\tilde{\Theta}_2 A & e^T\tilde{\Theta}_1 e + e^T\tilde{\Theta}_2 e \end{bmatrix}.$$

Let $d_z = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$ *be any solution to* $\tilde{P}d = e_{n+1}$ *and assume that*

$$\tilde{\Theta}_1(-Ad_x + d_y e) < 0, \quad \text{and} \quad \tilde{\Theta}_2(Ad_x + d_y e) > 0. \tag{5.1}$$

Then, for $0 < t \le t_0 < \delta$, $\theta_1(M_t) = \tilde{\theta}_1$ *and* $\theta_2(M_t) = \tilde{\theta}_2$ *where* $\delta = \min\{\delta_1, \delta_2\}$ *with* $\delta_1 = \max_{r_1(x^*,y^*)<0} r_1(x^*, y^*)$ *and* $\delta_2 = \min_{r_2(x^*,y^*)>0} r_2(x^*, y^*)$.

The proof of this theorem can be obtained, mutatis mutandis, from the proof of Theorem 7 in [12]. A close inspection of conditions (5.1) reveals that they are equivalent to a nondegeneracy assumption which is satisfied by the function approximation problems. The theorem shows that one should expect to decrease the parameter t to the level of the smallest optimal nonzero residual value before termination occurs. This implies that LINFSOL has to reduce t several times, resulting in a large number of iterations. This also makes the design of an effective starting strategy difficult for this class of problems. Bartels, Conn and Li [4] showed that the function approximation problems in the ℓ_∞ norm are characterized by a *sign alternating property* which states that at an optimal solution there are $n+1$ equations whose residuals $|b_i - a_i^T x|$ correspond to the maximum residual $||Ax - b||_\infty$ with error terms $(b_i - a_i^T x)$ alternating in sign as the counter moves from 1 to $n+1$. They propose an alternative starting point for their primal nondifferentiable penalty code based on Chebychev theory. In an

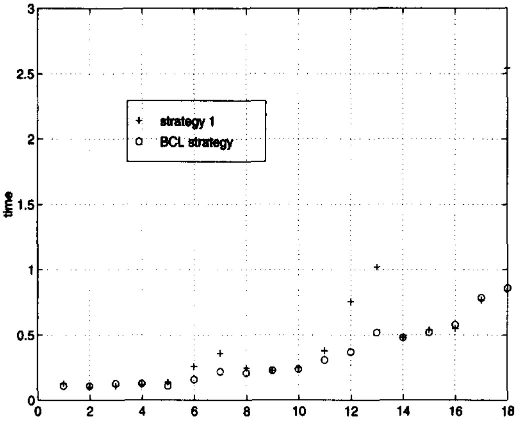

Figure 5.2: Performance of initialization strategies for function approximation problems (problem number $m = 50, 100, 200$; $n = 2, 3, 4, 5, 6, 7$).

effort to improve the performance of LINFSOL on these problems we have experimented with several strategies based on the Bartels-Conn-Li recommendations. We have finally settled for the following strategy. Let $z_i = \xi_1 + \frac{\xi_2 - \xi_1}{n}(i-1)$ for $i = 1, \ldots, n+1$. Consider the $(n+1) \times (n+1)$ linear system of equations

$$f(z_i) - \sum_{j=1}^{n} \gamma_j z_i^{j-1} = (-1)^i \phi, \quad i = 1, \ldots, n+1,$$

in the $n+1$ unknowns $\gamma_1, \ldots, \gamma_n$ and ϕ. We solve this system and use the solution as the initial point for LINFSOL. We choose $t^0 = 10^{-n+1}$ following Theorem 5.1. This point results in a significant improvement in some problems. This is illustrated in Figure 5.2 where we compare the Bartels-Conn-Li strategy (BCL) with Strategy 1 described above using the exponential function over the interval $[0, 1]$ (i.e., $\xi_1 = 0$ and $\xi_2 = 1$). We have used 18 problems where m assumes the values 50, 100 and 200, and n assumes the values $2, 3, 4, 5, 6, 7$ for each value of m in this order. This test was done on a SUN SPARC 4 Workstation.

5.4 Performance and comparison with the competing methods.

In this section we report the results of our comparisons with the Barrodale-Phillips code and the interior point algorithm of Zhang. We refer to the Barrodale-Phillips code as BP and to the predictor-corrector interior point code as PCIP. To present our results we use eight plots where we report the average run time and iteration results for ten problem instances for given m and n. Our purpose is to display continuous behaviour as the problem size is increased. For the non-degenerate problems the first two plot reports run time and iteration results for problems where the ratio m/n is kept constant at two and m varies from 60 to

800 in steps of 20. To illustrate behaviour at a higher m/n ratio the next two plots report results where the ratio m/n is kept constant at four and m varies from 120 to 800 in steps of 20. For the primal and dual degenerate problems we give four plots (run time and iteration results) where we keep the ratio $m/n = 2$ and vary m from 60 to 800 in steps of 20. In the charts where iteration numbers are graphed BP iterations have been divided by 10 to make the plots easier to read. Furthermore, we also report the number of refactorizations in LINFSOL, in connection with the computations of factors L and D. All run time results are in CPU seconds exclusive of input and output. All three codes report results accurate to machine precision. We remark that LINFSOL may decrease the value of the parameter t to 10^{-4} for some of the random problems.

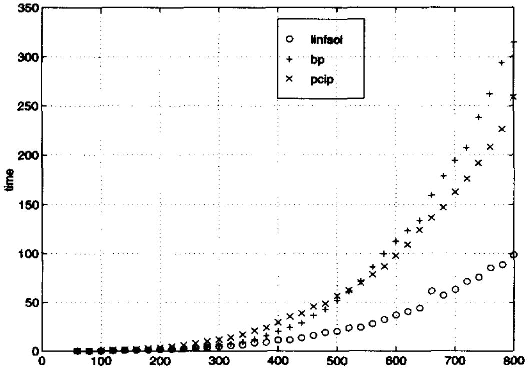

Figure 5.3: Run time comparison for nondegenerate problems $(m/n = 2)$.

We begin with the randomly generated nondegenerate problems with the ratio $m/n = 2$ in Figures 5.3 and 5.4. Here we observe that in general LINFSOL is increasingly faster than BP by a factor of three. It is also faster than PCIP by approximately a factor of two and a half. The number of iterations of LINFSOL grows very slowly while that of PCIP remains almost constant around 10. We observe a steady growth in the number of iterations of BP. LINFSOL performs between three and fourteen t-reductions on these problems in average. The average number of refactorizations remains around three.

In Figures 5.5 and 5.6 we present the same information for nondegenerate problems with the ratio $m/n = 4$.

We observe that LINFSOL outperforms both BP and PCIP by a factor of three and two and a half, respectively, as the problem size is increased. The number of t-reductions vary between five and sixteen. The average number of refactorizations in LINFSOL is between three and four. PCIP uses between nine and eleven iterations on these problems.

In Figures 5.7, 5.8, 5.9 and 5.10 we report results on primal and dual degenerate problems, respectively. Here, the ratio m/n is kept constant at two. The degree of primal degeneracy is *pdeg* $= \lfloor \frac{m-n}{2} \rfloor$ while we use *ddeg* $= \lfloor \frac{n}{2} \rfloor$ for dual

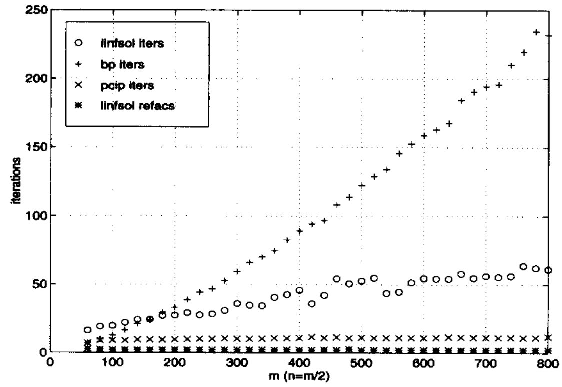

Figure 5.4: Iteration comparison for nondegenerate problems $(m/n = 2)$.

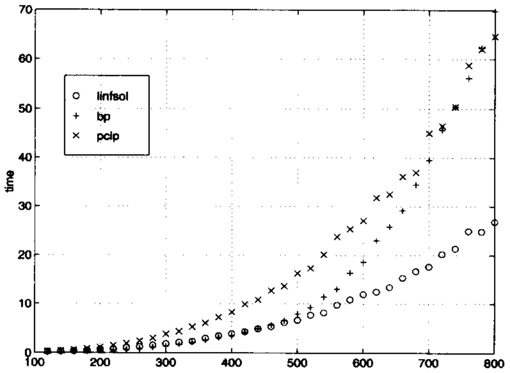

Figure 5.5: Run time comparison for nondegenerate problems $(m/n = 4)$.

degenerate problems.

We notice that LINFSOL performs better on primal degenerate problems compared to dual degenerate problems. This may be due to the fact that dual degenerate problems have non-unique primal optimal solutions. In our experience we have found this factor to weaken the performance of the penalty algorithm. It is worth mentioning here that the performance of BP is also adversely affected by dual degeneracy. For both classes we observe that LINFSOL becomes increasingly faster than BP. More precisely, for primal degenerate problems LINFSOL sustains a speed-up of three and a half over BP while it reaches a speed-up of three in the dual degenerate case. For primal degenerate problems LINFSOL is twice as fast as PCIP. On dual degenerate problems, PCIP appears to do better than LINFSOL reaching a speed-up factor of one and a half. This is reflected in the number of iterations of PCIP, which rarely exceeds six for dual degenerate problems while it is around eleven in average for primal degenerate test cases.

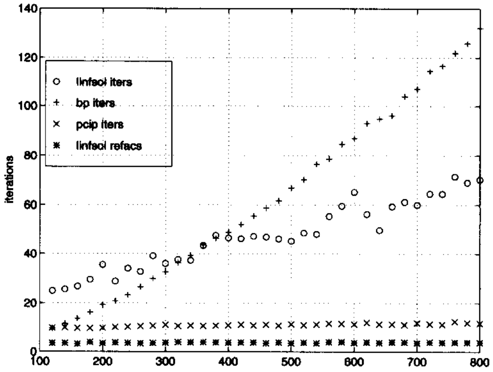

Figure 5.6: Iteration comparison for nondegenerate problems $(m/n = 4)$.

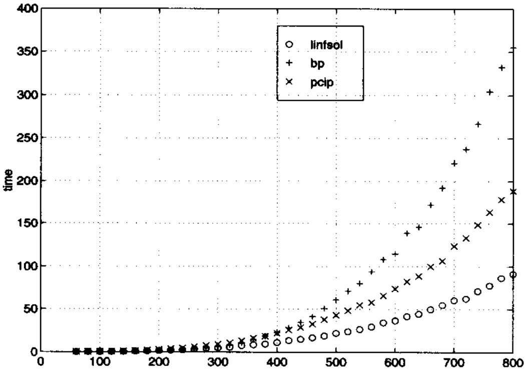

Figure 5.7: Run time comparison for primal degenerate problems $(m/n = 2)$.

The number of t-reductions of LINFSOL for the dual degenerate case varies between 1 and 28 while it is between two and eight for the primal degenerate case. The number of refactorizations is between four and nine for the dual degenerate case whereas it remains constant around two for the primal degenerate case.

We can infer the following conclusions from the above results.

- The penalty method appears to do best on nondegenerate and primal degenerate problems.
- The penalty method is expected to perform better after a certain threshold problem size is reached.
- In all cases, we have observed that the number of refactorizations of the penalty method remains almost constant as the problems become larger.
- The iteration charts reveal that the simplex method has a steady growth

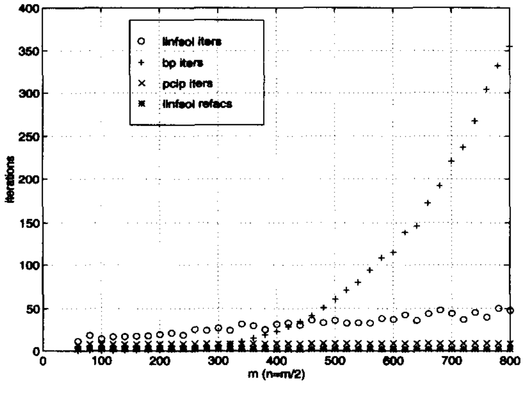

Figure 5.8: Iteration comparison for primal degenerate problems $(m/n = 2)$.

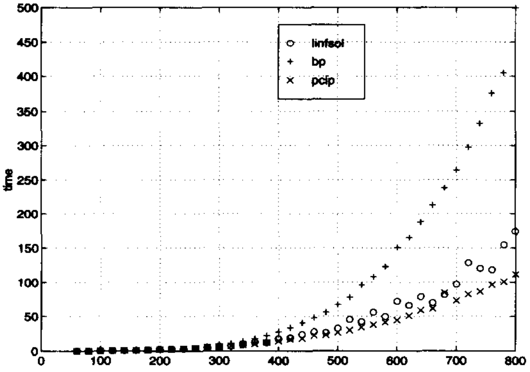

Figure 5.9: Run time comparison for dual degenerate problems $(m/n = 2)$.

in the number of iterations. The interior point method uses an almost constant number of iterations for a given problem class while the penalty method exhibits a very low growth rate in the number of iterations as the problem size is increased.

- The predictor-corrector algorithm seems to do best on dual degenerate problems.

Finally, we summarize our experience with function approximation problems. For this purpose we choose to approximate the exponential, the square root and the sine functions on the interval $[0, 1]$. We have solved 54 problems of varying dimensions altogether, 18 for each function. These experiments showed that BP is about ten times faster than both LINFSOL and PCIP on the average while LINFSOL and PCIP exhibit a similar performance. Two factors play an important role here. The first one is the outstanding performance of BP on these problems.

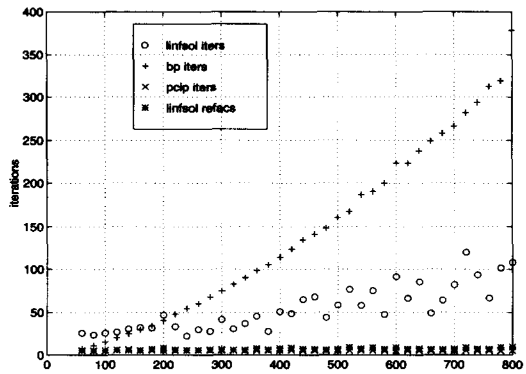

Figure 5.10: Iteration comparison for dual degenerate problems $(m/n = 2)$.

This is partially due to the fact these problems have at most eight variables. This is a range where BP is very efficient. Furthermore, Bartels, Conn and Li [4] prove that the high efficiency of the BP algorithm on function approximation problems is due to the fact that BP exploits the special structure of function approximation problems. It is proved in [4] that after the first stage (Phase I) BP finds a feasible solution that satisfies the sign alternating property as described in Section 5.3. The second factor is that the optimal solution is usually near degenerate for these problems as defined in Section 5.3. This degrades the performance of LINFSOL considerably. We substantiated this observation by Theorem 5.1 in Section 5.3. We would like to note, however, that the accuracy of the solution is maintained in all the cases, and LINFSOL solves all these problems within at most three CPU seconds. Interestingly, the performance of PCIP deteriorates considerably on these problems as well. A similar degradation in performance with function approximation problems is reported in [9] for the exact penalty method and is attributed to near degeneracy as well.

6 Conclusions.

In this paper, a penalty continuation algorithm was designed for linear ℓ_∞ problems. The computational results indicate that the algorithm is stable and accurate on different kinds of problems. Furthermore, it was shown to successfully compete with Barrodale-Phillips algorithm and the predictor-corrector primal-dual interior point algorithm of Zhang on a wide range of random problems. Based on our tests, the algorithm seems to perform best on problems with no special structure (e.g., problems not derived from function approximation) where the number of variables exceeds a certain threshold.

Acknowledgement.

The assistance of the associate editor K. Madsen, and the careful reviews of two anonymous referees are gratefully acknowledged.

REFERENCES

1. R. H. Bartels and G. H. Golub, *Stable numerical methods for obtaining the Chebyshev solution to an overdetermined system of equations*, Comm. ACM., 11 (1968), pp. 401–406.
2. I. Barrodale and C. Phillips, *An improved algorithm for discrete Chebychev linear approximation*, in Proc. Fourth Manitoba Conf. on Numerical Mathematics, University of Manitoba, Winnipeg, Canada, 1974, pp. 177–190.
3. R. H. Bartels, A. R. Conn, and C. Charambous, *On Cline's direct method for solving overdetermined linear systems in the ℓ_∞ sense*, SIAM J. Numer. Anal., 15 (1978), pp. 225–270.
4. R. H. Bartels, A. R. Conn, and Y. Li, *Primal methods are better than dual methods for solving overdetermined linear systems in the ℓ_∞ sense?*, SIAM J. Numer. Anal., 26 (1989), pp. 693–726.
5. D. P. Bertsekas, *Newton's method for linear optimal control problems*, in Proc. of Symposium on Large Scale Systems, Udine, Italy, 1976, pp. 353–359.
6. T. F. Coleman and Y. Li, *A global and quadratically convergent method for linear ℓ_∞ problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1166–1186.
7. J. Dongarra, J. Du Croz, S. Hammarling, and R. Hansen, *An extended set of basic linear algebra subprograms*, ACM Trans. Math. Software, 14 (1988), pp. 1–17.
8. R. Fletcher, *Practical Methods of Optimization*, Second Edition, John Wiley & Sons, Chichester, 1987.
9. B. Joe and R. Bartels, *An exact penalty method for constrained, dicrete, linear ℓ_∞ data fitting*, SIAM J. Sci. Comput., 4 (1983), pp. 69–84.
10. K. Madsen and H. B. Nielsen, *Finite algorithms for robust linear regression*, BIT, 30 (1990), pp. 682–699.
11. K. Madsen and H. B. Nielsen, *A finite smoothing algorithm for linear ℓ_1 estimation*, SIAM J. Optim., 3 (1993), pp. 223–235.
12. K. Madsen, H. B. Nielsen, and M.Ç. Pınar, *New characterizations of ℓ_1 solutions of overdetermined linear systems*, Oper. Res. Letters, 16 (1994), pp. 159–166.
13. K. Madsen, H. B. Nielsen, and M.Ç. Pınar, *A new finite continuation algorithm for linear programming*, SIAM J. Optim., 6 (1996), pp. 600–616.
14. NAG Fortran Library Routine Document, Numerical Analysis Group, 1981.
15. H. B. Nielsen, *AAFAC: A Package of Fortran 77 Subprograms for Solving $A^TAx = c$*, Report NI-90-11, Institute for Numerical Analysis, Technical University of Denmark, 1990.
16. M. Ç. Pınar, *Piecewise linear pathways to the optimal solution set in linear programming*, J. Optim. Theory Appl., 93 (1997), pp. 619–634.
17. S. A. Ruzinsky and E. T. Olsen, *ℓ_1 and ℓ_∞ minimization via a variant of Karmarkar's algorithm*, IEEE Transactions on Acoustics, Speech and Signal Processing, 37 (1989), pp. 245–253.
18. G. A. Watson, *Approximation Theory and Numerical Methods.* John Wiley, New York, 1980.
19. Y. Zhang, *Primal-dual interior point approach for computing the ℓ_1-solutions and ℓ_∞-solutions of overdetermined linear systems*, J. Optim. Theory Appl., 77 (1993), pp. 323–341.