FACULTY WORKING PAPER 92-0127

An Interior Point Algorithm for Nonlinear Quantile Regression

Roger Koenker
Department of Economics
University of Illinois

Beum J. Park
Graduate Assistant
University of Illinois

Bureau of Economic and Business Research
College of Commerce and Business Administration
University of Illinois at Urbana-Champaign

BEBR

FACULTY WORKING PAPER NO. 92-0127

College of Commerce and Business Administration

University of Illinois at Urbana-Champaign

May 1992

An Interior Point Algorithm for Nonlinear Quantile Regression

Roger Koenker
Department of Economics

Beum J. Park
Graduate Assistant

An Interior Point Algorithm for Nonlinear Quantile Regression

Roger Koenker*

and

Beum J. Park*

April 1992

Abstract

A new algorithm for computing quantile regression estimates for problems in which the response function is nonlinear in parameters is described. The nonlinear l_1 estimation problem is a special (median) case. The algorithm is closely related to recent developments on interior point methods for solving linear programs. Performance of the algorithm on a variety of test problems including the censored linear quantile regression problem of Powell (1986) is reported.

Keywords: Quantile Regression, Nonlinear Regression, Linear Programming, Interior Point Algorithms, Nonlinear Programming

*Department of Economics, University of Illinois, Champaign, Illinois, 61820. This work was partially supported by NSF grant SES 91-21776.

1. Introduction

About a century ago Edgeworth observed that methods of estimation based upon minimizing sums of *absolute* residuals could be far superior to least-squares methods under non-Gaussian error conditions. Laplace had drawn similar conclusions a century earlier. See Stigler (1986) on this early history. But computation of l_1-estimators, even for linear regression, remained a major impediment to applications until the emergence of the simplex algorithm for linear programming in the 1940's. Papers by Charnes, Cooper and Ferguson (1955) and Wagner (1959) provided a foundation for modern algorithms for linear l_1-regression by Barrodale and Roberts (1973), Bartels and Conn (1980) and others. These algorithms are readily extended to linear quantile regression, Koenker and Bassett (1978), of which l_1-regression is an important (median) special case.

The current state of algorithms for *nonlinear* quantile regression is far less satisfactory. Certainly nothing comparable to the venerable Gauss-Newton algorithm for nonlinear least squares problems has emerged. Despite a flurry of interest by prominent numerical analysts in the 1970's and early 1980's, see, e.g., Osborne and Watson (1971), Murray and Overton (1981) and Bartels and Conn (1982), occasional applications of nonlinear quantile regression have relied on the Nelder and Mead (1965) algorithm and other generic optimization methods. An excellent statement of the current state-of-the-art is provided in the thesis of Busovaca (1985).

In contrast, the statistical theory of nonlinear quantile regression has developed rapidly in recent years. Powell (1986) has emphasized its value in the analysis of censored and truncated responses. Asymptotic theory for independent errors has been developed by Oberhofer(1982), Dupacova(1987), and Powell(1991). Theoretical developments by Weiss (1991) and White (1991) have stressed applications to time-series analysis. Applications of Horowitz and Neu-

mann (1987), Chamberlain (1990), and others have demonstrated its value in applied econometrics.

In this paper we will describe a new approach to the computation of nonlinear quantile regression estimators based on recent interior point methods for solving linear programs. In the next section we review interior point methods for strictly linear problems. Section 3 describes our approach to nonlinear problems, and Section 4 describes our computational experience.

2. Interior Point Methods for Linear Programs

In this section we provide a brief discussion of interior point methods for solving strictly linear programs including the linear quantile regression problem. Our exposition will follow closely that of Vanderbei, Meketon, and Freedman (1986) and Meketon (1986). We should emphasize that *in our experience* interior point algorithms for linear quantile regression do *not* appear to be competitive in efficiency terms with existing simplex method algorithms. See Koenker and d'Orey(1987) for a description of a simplex based algorithm for linear quantile regression. However, unlike simplex based methods they do appear to offer a natural extension to nonlinear problems. Thus a clear understanding of the linear case is an essential first step in our exposition.

2.1. A Canonical LP

Consider the equality constrained linear program

$$\min \{ c'\omega \mid \omega \in \Omega \equiv \{ \omega \in \mathbf{R}^n_+,\ A\omega = b \} \} \tag{2.1}$$

where $\mathbf{R}^n_+$ denotes the positive orthant of $\mathbf{R}^n$. Given a feasible point in the interior of the constraint set, $\omega \in \text{int}\,(\Omega)$, interior point methods proceed in two steps. First we transform coordinates to reposition ω so it is *centered* relative to the set Ω. Then a (projected) gradient step is

taken toward the boundary of Ω. Repeating this process brings us arbitrarily close to a solution, and a stopping criterion is eventually invoked.

To flesh out this brief description, let $D = diag\ (\omega)$ and consider the transformation

$$\omega \rightarrow D^{-1}\omega$$

We have $D^{-1}\omega = 1_n$, an n-vector of ones, so the transformation D has the effect of centering ω relative to the orthant boundaries of Ω. Correspondingly, we may define $\tilde{A} = AD$ and $\tilde{c} = Dc$. In the transformed coordinates we wish to move in the gradient direction $-\tilde{c}$, but to preserve feasibility we should instead project $\tilde{c}$ onto the null space of $\tilde{A}$ to insure that the equality constraints are satisfied.

Let $\hat{c}$ denote this projection, i.e.,

$$\hat{c} = (I - \tilde{A}'(\tilde{A}'\tilde{A})^{-1}\tilde{A})\tilde{c}$$

Clearly, $\hat{c}$ is a direction of descent; and we now move toward the boundary of Ω in this direction. Let

$$\hat{\alpha} = \max_{i=1,\ldots,n} \{e_i'\hat{c}\ \}$$

where e_i is the i^{th} unit basis vector for $\mathbf{R}^n$. For some fixed $\eta \in (0, 1)$, consider

$$\omega \leftarrow \omega - (\eta/\alpha)D\hat{c}$$

which defines a sequence of iterations $\omega_{k+1} = T(\omega_k)$. Since at each iteration

$$c'\omega_{k+1} = c'\omega_k - (\eta/\alpha)c'D\hat{c} = c'\omega_k - (\eta/\alpha)\|\hat{c}\|,$$

we expect to see an improvement in the objective function at each iteration.

Proposition. If $\hat{c} \leq 0$ the problem (2.1) is unbounded, unless $\hat{c} = 0$ in which case every $\omega \in \Omega$ is

optimal. Otherwise, the problem is bounded and the sequence $\{c'\omega_k\}$ is strictly decreasing.

Proof. Since the proof of this proposition, found in Vanderbei, Meketon and Freedman (1986), is both elementary and revealing we repeat it here for the sake of completeness. If $\hat{c} = 0$, there exists a vector z such that $\tilde{c} = \tilde{A}'z$, hence $D\tilde{c} = D\tilde{A}z$ and since $\omega \in int(\Omega)$ it follows that $c = Az$. But then for any $\omega \in \Omega$,

$$c'\omega = z'A\omega = z'b$$

which is independent of ω, establishing that $c'\omega$ is constant on all of Ω. Next consider $\hat{c} \leq 0$. Note that

$$c'\omega_1 \equiv c'\omega - \gamma(\omega)\tilde{c}'\hat{c} = c'\omega - \gamma(\omega) \| \hat{c} \|_2^2 \tag{2.4}$$

where $\gamma(\omega) = \eta/\alpha$. Since $\hat{c} \leq 0$,

$$\omega_\rho = \omega - \rho D\hat{c}$$

is feasible for any $\rho > 0$ and

$$c'\omega_\rho = c'\omega - \rho \| \hat{c} \|_2^2$$

implies that $c'\omega_\rho \to -\infty$ as $\rho \to \infty$. Finally, if $\hat{c} \geq 0$, then since $\gamma(\omega) > 0$ and $c'\omega_1 < c'\omega$ follows from (2.4), establishing that the step is a direction of descent. ■

2.2. Linear l_1-regression

In the linear model

$$y_i = x_i'\beta + u_i \qquad i = 1, \cdots, n,$$

as noted in the introduction, the l_1-estimator of β which minimizes

$$R(b) = \sum_{i=1}^{n} |y_i - x_i'b| .$$

may be formulated as a linear program. The dual problem may be written as

$$\max \{ y'd \mid d \in \Omega = \{ d \in [-1, 1]^n, X'd = 0 \} \}$$

where y is the n-vector of responses, and X is the $n \times p$ design matrix. To solve the dual problem we proceed as before, except that the centering is slightly altered to accommodate the altered form of Ω. For any initial feasible point d, e.g., $d = 0$, following Meketon (1986), set

$$D = diag\ (\min \{ 1+d_i, 1-d_i \}).$$

In the transformed coordinates $D^{-1}d$ the projected gradient is

$$D\hat{u} = (I - DX'(X'D^2X)^{-1}X'D)Dy = D(y - Xb)$$

where $b = (X'D^2X)^{-1}D^2y$. Note that as in the former case the transformation has the effect of centering the point d in the feasible set Ω. Now let

$$\alpha = \max_i \{ \max \{ \frac{e_i'D^2\hat{u}}{1 + d_i}, \frac{-e_i'D^2\hat{u}}{1 - d_i} \} \}$$

and again for $\eta \in (0, 1)$ we take the step

$$d \leftarrow d + (\eta/\alpha)D^2\hat{u}.$$

Note the change in sign since we are now maximizing. The iteration sequence $d_{k+1} = T(d_k)$ in the dual vector implicitly defines a corresponding primal sequence with

$$b_k = (X'D_k^2X)^{-1}X'D_k^2y.$$

As Meketon notes, the duality theory yields a natural stopping criterion. Since

$$y'd_k \leq \sum |y_i - x_i'b_k|$$

with optimality if and only if equality holds, it is reasonable to stop iterating when the difference between the dual and primal values is less than a specified tolerance.

2.3 Linear Quantile Regression

If we replace the (symmetric) l_1-criterion with an asymmetric linear criterion so we minimize

$$R_\theta(b) = \sum_{i=1}^{n} \rho_\theta(y_i - x_i b)$$

$\rho_\theta(u) = (\theta - I(u < 0))u$, we obtain the regression quantiles of Koenker and Bassett (1978). The dual problem is now,

$$\max \{ y'd \mid d \in \Omega = \{ d \in [\theta-1, \theta]^n, \; X'd = 0 \} \}$$

This leads to an algorithm identical to the l_1 special case except that now

$$D = diag(\min(\theta - d_i, \; 1 - \theta + d_i))$$

and

$$\alpha = \max_i \left(\max \left\{ \frac{e_i' D \hat{c}}{\theta - d_i}, \; \frac{-e_i' D \hat{c}}{1 - \theta + d_i} \right\} \right).$$

3. Nonlinear Quantile Regression

To extend these ideas to the case of nonlinear response functions we begin by considering the nonlinear l_1 problem

$$\min_{t \in \mathbf{R}^p} \sum |f_i(t)| \tag{3.1}$$

where, for example,

$$f_i(t) = y_i - f_0(x_i, t).$$

As noted by El Attar, *et al* (1979) a necessary condition for t^* to solve (3.1) is that there exists a vector $d \in [-1, 1]^n$ such that

$$J(t^*)'d = 0 \tag{3.2}$$

$$f(t^*)'d = \sum |f_i(t^*)| \tag{3.3}$$

where $f(t) = (f_i(t))$ and $J(t) = (\partial f_i(t)/\partial t_j)$

Thus, as proposed by Osborne and Watson (1971), one approach to solving (3.1) is to solve a succession of linearized l_1 problems minimizing

$$\sum |f_i(t) - J_i(t)'\delta| = \|f - J\delta\|_1,$$

choosing a step length, λ, at each iteration, by line search in the resulting directions δ. The difficulty, as we see it, with this approach is not only that we must expend the effort to solve an l_1 linear program at each iteration, but, perhaps more significantly, the resulting directions may actually be inferior to directions determined by incomplete solutions to the sequence of linearized problems.

Let t be the value of the parameter at the current iteration, and consider the dual problem

$$\max \{f'd \in [-1, 1]^n, \ J'd = 0\}. \tag{3.4}$$

If the model were linear so

$$f(s) = f(t) - K(s - t)$$

for some fixed matrix K, then a solution can be found by applying Meketon's algorithm to find d^* to solve (3.4), computing

$$\delta^* = (K'D^2K)^{-1}K'D^2f.$$

where $D = diag\ (\min\ \{1-d_i^*,\ 1+d_i^*\ \})$ and setting $t^* = t + \delta^*$. When f is nonlinear there is no longer a compelling argument for fully solving (3.4) at each iteration, indeed, in our experience only a few iterations to refine the dual vector is preferable. In the version of the algorithm we have implemented to conduct the tests reported in the next section we take two dual steps between successive updates of f and J. A detailed description of the algorithm is now provided.

3.1. Dual Step

For any feasible d in the interior of the constraint set of (3.4) we refine d, following Meketon, as follows. Let

$$D = diag\ (\min\ \{1-d_i,\ 1+d_i\ \})$$

$$s = D^2(I - J(J'D^2J)^{-1}JD^2)f.$$

$$d \leftarrow d + (\eta/\alpha)s$$

where

$$\alpha = \max\ \{\ \max\ \{s_i/(1 - d_i),\ -s_i/(1 + d_i)\ \}\ \}$$

and $\eta \in (0,\ 1)$ is the constant chosen to insure feasibility. Following Meketon, we use $\eta = .97$. Updating D, s, and the new d continues the iteration. This process is embedded in a sequence of primal iterations in which we update f and J as follows.

3.2. Primal Step

The dual step yields the primal direction

$$\delta = (J'D^2J)^{-1}J'D^2f$$

which we explore by line search. Our current implementation uses Brent's (1973) algorithm from the PORT3 library Fox (1984). Updating we have

$$t \leftarrow t + \lambda^*\delta$$

where $\lambda^* = \underset{\lambda}{\text{argmin}} \|f(t + \lambda\delta)\|_1$, and we then update f and J. However before returning to the dual step we must adjust the current d so that it is feasible for the new value of J. This is accomplished, somewhat naively, by projecting the current d onto the null space of the new J, i.e. $\hat{d} = (I - J(J'J)^{-1}J')d$ and then shrinking it to insure that it lies in $[-1, 1]^n$, so

$$d \leftarrow \hat{d}/(\max_i \{ |\hat{d}_i| \} + \varepsilon)$$

for some tolerance parameter $\varepsilon > 0$. Obviously, when the problem is linear, so J is fixed, this "adjustment" is nugatory since d is already in the null space of J, and the algorithm is essentially like Meketon's.

3.3. Stopping

The algorithm currently terminates when the new iterate fails to improve the objective function by a specified tolerance. Exploration of alternative stopping rules is a topic for future research.

3.4 Related Literature

Gill, Murray, Saunders, Tomlin, and Wright (1986) and Bayer and Lagarias (1991) have recently pointed out the close relationship of "projected Newton barrier" methods (see Fiacco and McCormick (1965)) and interior point methods. Algorithms closely related to the one described above could presumably be formulated employing logarithmic barrier functions in the

dual vector d.

3.5. Quantile Regression

As in the case of the linear problem the generalization of the l_1 problem to other quantiles is straightforward involving only a modification of the constraint set $[-1, 1]^n$ to $[\theta-1, \theta]^n$ for some $\theta \in (0, 1)$.

4. Numerical Experience

In this section we describe our computational experience with a variety of test problems. To facilitate comparison with existing results in the literature we have chosen problems from Busovaca(1985) and Wormersley(1986). We focus exclusively on the l_1 case since there are no comparable results in the literature for other quantiles. The problems used are described in detail in Appendix A. We have attempted to investigate all of the problems reported on by Busovaca, however in several cases we were unable find a complete description of the problem. The problem taken from Wormersley is included to explore the important special case of piecewise linear reponse functions which arise in Powell's(1986) formulation of the quantile regression problem for censored data.

In Appendix B we provide explicit versions of our algorithm in both S, Becker, Chambers and Wilks(1988), and Gauss. All of the reported tests were carried out in S on a Sun 3/50. Note that the line search algorithm in the S and Gauss versions are different. To implement a simple version of the Osborne and Watson(1971) algorithm in S we employ the S function `l1fit` which does l_1 regression using the Barrodale and Roberts(1973) algorithm. The S function `lsfit` carries out the corresponding weighted least squares computations for the interior point algorithm.

A summary of our experience on the test problems appears in Tables 5.1 and 5.2.. For Wormersley's(1986) censored regression problem (Problem 1) our version of the interior point algorithm converges to Wormersley's reported solution. However, it should be noted that the solution to this problem is notoriously nonunique. Busovaca's algorithm cannot be employed on Problem 1 due to the fact that the Hessian of the response function is identically zero almost everywhere. The remaining problems are all taken from Busovaca, and generally our interior point solutions correspond closely to his. In Problems 7 and 13 there are small discrepancies favoring Busovaca; in Problem 9 there is a larger descrepancy favoring the interior point method. Results for our implementation of the Osborne and Watson algorithm are somewhat less satisfactory. It fails completely on Problems 11 and 12, performs poorly in Problems 1, 4b, and 13, but does slightly better than the interior point method on Problem 5. All three algorithms fail for Problem 4a which is highly degenerate at the specified initial point. At an alternative starting point, the interior point algorithm performs well.

5. Some Concluding Remarks

We have described a simple approach to computing quantile regression estimates for problems with nonlinear response functions. The approach is based on recent developments on interior point methods for linear programming, but may be viewed as a variant of well-known iteratively reweighted least squares. While the algorithm seems to perform well on a variety of test problems, there is considerable room for improvement. Handling rank deficiency in the model Jacobian is critical. Alternative stopping criteria also should be explored.

References

Bard, Y. (1970). Comparison of gradient methods for the solution of nonlinear parameter estimation problems. *SIAM J. Numer. Anal.* 7, pp. 157-186.

Barrodale, I and F.D.K. Roberts (1974), Solution of an overdetermined system of equations in the l_1 norm, *Communications ACM, 17*, 319-320.

Bartels, R.H. and A.R. Conn (1980), Linearly constrained discrete l1 problems, *ACM Trans. on Math. Software* 6, 594-608.

Bartels, R.H. and A.R. Conn (1982), An approach to nonlinear l1 data fitting, *Lecture Notes in Mathematics and Numerical Analysis,* J.P. Hennart, ed., Vol. 909, 48-58.

Bayer, D.A. and J.C. Lagarias (1991), Karmarkar's linear programming algorithm and Newton's method, *Mathematical Programming,* 50, 291-330.

Beale, E.M.L. (1958). On an iterative method of finding a local minimum of a function of more than one variable. Tech. Rep. No. 25, Statistical Techniques Research Group, Princeton University.

Becker, R.A., Chambers, J.M., and Wilks, A.R. (1988) *The New S Language,* Wadsworth: Pacific Grove.

Biggs, M.C. (1971). Minimization algorithms making use of non-quadratic properties of the objective function. *J. Inst. Math. Appl.*, 8, pp. 315-327.

Brown, K.M. and J.E. Dennis (1971). New computational algorithms for minimizing a sum of

squares of nonlinear functions. Rep. No. 71-6, Yale University, Dept. of Comput. Science.

Busovaca, S. (1985). Handling degeneracy in a nonlinear l_1 algorithm, Technical Report CS-85-34, Dept. of Computer Science, Univ. of Waterloo.

Chamberlain, G. (1990), Quantile regression, censoring and the structure of wages, Invited paper presented at the 6th World Congress of the Econometric Society.

Charnes, A., W.W. Cooper and R.O. Ferguson (1955), Optimal Estimation of executive compensation by linear programming, *Mgt. Sci.*, 1, 138-151.

Colville, A.R. (1968). A comparative study of nonlinear programming codes. Rep. 320-2949, IBM New York Scientific Center.

Dupacova, J. (1987). Asymptotic properties of restricted L_1-estimates of regression *Statistical Data Analysis Based on the L_1-norm and Related Methodsr,* Y. Dodge (Ed); Elsevier North-Holland: Amsterdam,

El-Attar, R.A., M. Vidyasagar and S.R.K. Dutta (1979). An algorithm for l_1-norm minimization with application to nonlinear l_1-approximation. *SIAM J. Numer. Anal.*, 16, pp. 70-86.

Fiacco, A.V. and G.P. McCormick (1965), *Nonlinear programming: Sequential unconstrained minmization techniques,* John Wiley, New York.

Fox, P.A. (1984). *The PORT Mathematical Subroutine Library,* Bell Laboratories.

Gill, P.E., W. Murray, M.A. Saunders, J.A. Tomlin and M.M. Wright (1986), On projected New-

ton barrier methods for linear programming and an equivalence to Karmarkar's projective method, *Mathematical Programming*, 36, 183-209.

Horowitz, J. and G. Neumann (1987), Semiparametric estimation of employment duration models, *Econometric Reviews*, 6 , 5-46.

Koenker, R. and V. d'Orey (1987), Computing Regression Quantiles, *Applied Statistics*, 36, 383-390.

Koenker, R.W. and G.W. Bassett (1978), Regression quantiles. *Econometrica*. 46, 33-50.

Kowalik, J.S. and M.R. Osborne (1968). *Methods for Unconstrained Optimization Problems*. Elsevier North-Holland, New York.

Meketon, M. (1986), Least absolute value regression. AT&T Bell Lab. Tech. Rep., Holmdel, NJ.

Murray, W and M. Overton (1981), A projected lagrangian algorithm for nonlinear l1 optimization, *SIAM J. Scientific and Statistical Computing*, 1, 207-224.

Nelder, J. and R. Mead (1965), A simplex method for function minimization, *Computer J.*, 7, 308-313.

Oberhofer, W. (1982). The consistency of nonlinear regression minimizing the L_1-norm *Ann. of Statistics*, 10, 316-319.

Osborne, M.R. (1972). Some aspects of nonlinear least squares calculations. In *Numerical Methods for Nonlinear Optimization*, E.A. Lootsma (Ed.). Academic Press, New York, pp. 171-189. .sp

Osborne, M.R. and G.A. Watson (1971), On an algorithm for discrete nonlinear l1 approximation, *Comput. J.*, 14, 184-188.

Overton, M.L. and W. Murray (1981). A projected Lagrangian algorithm for nonlinear l_1 optimization. *SIAM J. of Scientific & Statistical Computing*, 2. p. 207-224.

Powell, J. (1991). Estimation of monotone regression models under quantile restrictions, in *Nonparametric and Semiparametric Methods in Econometrics and Statistics*, W.A. Barnett, J. Powell, and G. Tauchen (eds.), Cambridge University Press.

Powell, J. (1986), Censored regression quantiles, *J. of Econometrics*, 32, 143-155.

Powell, M.J.D. (1962). An iterative method for finding stationary values of a function of several variables. *Computer J.*, 5, pp. 147-151.

Rosenbrock, H.H. (1960). An automatic method for finding the greatest or least value of a function. *Computer J.*, 3, pp. 175-184.

Stigler, S.M. (1986), *The History of Statistics*. Cambridge: Harvard Press.

Vanderbei, R.J., M.J. Meketon and B.A. Freedman (1986), A modification of Karmarkar's linear programming algorithm, *Algorithmica*, 1, 395-407.

Wagner, H.M. (1959), Linear programming techniques for regression analysis, *J. Amer. Stat. Assoc*. 54,206-212.

Weiss, A. (1991), Estimating nonlinear dynamic models using least absolute error estimation, *Econometric Theory*, 7, 46-68.

White H. (1991), Nonparametric estimation of conditional quantiles using neural networks, unpublished.

Wormersley, R.S. (1986). Censored discrete linear l_1 approximation. *SIAM J. of Scientific & Statistical Computing*, 7, pp. 105-122.

Figure 5.1
Algorithmic Performance on Several Test Problems

		Interior Point Algorithm		Osborne-Watson Algorithm		Busovaca Algorithm	
Example	Starting Point	Optimal Objective	Number of Iterations	Optimal Objective	Number of Iterations	Optimal Objective	Number of Iterations
1.Wormersley	(0,0)	3.032544	3	5.234825	3	NA	
2.Bard	(1,1,1)	0.1243555	6	0.1243383	5	0.1243406	13
3.Beale	(1,0.1)	0.2928905e-07	6	0.1547611e-06	6	0.3695488e-05	8
4.a.Biggs	(1,1,1,1,1,1)	F		F		F	
b.Biggs	(1,8,2,2,2,2)	0.0	11	0.7289559	45	NA	
5.Brown&Dennis	(25,5,-5,-1)	903.3648	29	903.2406	113	903.2343	2
6.El-Attar 5.1	(1,2)	0.47042	10	0.4704267	6	0.4704247	8
7.El-Attar 5.2	(1,1,1)	7.902733	13	7.904731	22	7.894227	5
8.Madsen	(3,1)	1.0	16	1.000010	11	1.000002	13
9.Osborne 1	(0.5,1.5,-1,0.01,0.02)	0.0293912	14	0.0293914	10	0.8203727	55
10.Osborne 2	(1.3,0.65,0.65,0.7,0.6, 3,5,7,2,4.5,5.5)	F		F		F	
11.Powell	(3,-1,0,1)	0.1272e-07	15	F		0.29039e-08	3
12.Rosenbrock	(-1.2,1)	0.0	15	F		0.506642e-06	51
13.Watson	(1,1,1,1)	0.6487749	25	1.278432	5	0.6018584	24
14.Wood	(0,0,0,0)	0.0	7	0.000003	10	0.0	25

See Appendix A for a detailed description of the test problems.

F indicates the algorithm failed to meet convergence criteria for the problem.

NA indicates results are not available for this entry.

Table 5.2
Optimal Points for Several Test Problems

Example	Interior Point Algorithm	Osborne-Watson Algorithm	Busovaca Algorithm
1.Wormersley	-6.74166, 4.59299	-7.29804, 4.74178	NA
2.Bard	0.10095, 1.52545, 1.97182	0.10094, 1.52516, 1.97211	0.10094, 1.52513, 1.97214
3.Beale	3, 0.5	3, 0.5	2.99999, 0.49999
4a.Biggs	F	F	F
b.Biggs	1, 10, 1, 5, 4, 3	1.82143, 81.94978, 2.27882	NA
5.Brown&Dennis	-9.70273, 11.74096, -0.44204 0.55827	-10.0227, 11.91354, -0.44026 0.55823	-10.2236, 11.90843, -0.45804 0.58032
6.El-Attar 5.1	2.84250, 1.92018	2.84250, 1.92018	2.84250, 1.92018
7.El-Attar 5.2	0.53558, -0.00139, 0.02871	0.53148, -0.00004, 0.02751	0.53606, 0.0, 0.00319
8.Madsen	0.0, 0.00016	0.0, 0.0022	0.0, -0.00205
9.Osborne 1	0.37706, 2.19244, -1.72549 0.01332, 0.02129	0.37706, 2.19246, -1.72552 0.01332, 0.02129	1.06716, 1.80257, -1.80731 0.00345, 0.00109
10.Osborne 2	F	F	F
11.Powell	0.7268e-04, -0.7268e-05	F	0.5588e-08, -0.3725e-09
	0.1162e-04, 0.1163e-04		0.1250e-08, 0.1716e-08
12.Rosenbrock	1.0, 1.0	F	0.99999, 0.99999
13.Watson	-0.37526, 1.14089 -0.42239, 0.39683	-0.23584, 1.03241 -0.22747, 0.41384	-0.44271, 1.19321 -0.47676, 0.38449
14.Wood	1.0, 1.0, 1.0, 1.0	1.0, 1.0, 1.0, 1.0	1.0, 1.0, 1.0, 1.0

Appendix A

Test problem 1 (Wormersley, 1986)

The results of temperature accelerated life tests on electrical insulation in 40 motorettes are recorded in Table A.1. This data is originally from Schmee and Hahn (1979). Ten motorettes were tested at each of four temperatures. Testing was terminated at different times at each temperature. The model used to fit the data is

$$log_{10}H = x_1 + \frac{1000x_2}{(T + 273.2)} + \epsilon,$$

where H is the failure time and T is the temperature.

Table A.1 : Data for motorettes example

	Test temperature $T°C$			
	150	170	190	200
Failure times H in hours		1764	408	408
		2772*	408	408
		3444	1344	504
		3542	1344	504
		3780	1440	504
		4860		
		5196		
Termination time $\bar{H}$	8064	5448	1680	528
	10 units	3 units	5 units	5 units

* Wormersley gives the second failure time at 170° as 2722, but his results are consistent with the values recorded here from Schmee and Hahn.

At each temperature there is an upper bound $\bar{H}$ (the time at which testing was stopped) on the observed failure times, so the algorithms of the observed failure times are given by

$$min\left(log_{10}\bar{H},\ x_1 + \frac{1000x_2}{(T+273.2)} + \epsilon\right).$$

Test problem 2 (Bard, 1970)

$$f_i(x) = y_i - \left(x_1 + \frac{u_i}{v_i x_2 + w_i x_3}\right)$$

where $i = 1, 2, \cdots, 15$, $u_i = i$, $v_i = 16 - i$, $w_i = min(u_i, v_i)$, and

i	y_i	i	y_i	i	y_i
1	0.14	6	0.32	11	0.73
2	0.18	7	0.35	12	0.96
3	0.22	8	0.39	13	1.34
4	0.25	9	0.37	14	2.10
5	0.29	10	0.58	15	4.39

Test problem 3 (Beale, 1958)

$$f_i(x) = y_i - x_1(1 - x_2^i),$$

where $i = 1, 2, 3$, $y_1 = 1.5$, $y_2 = 2.25$ and $y_3 = 2.625$.

Test problem 4 (Biggs, 1971)

$$f_i(x) = x_3 exp(-t_i x_1) - x_4 exp(-t_i x_2) + x_6 exp(-t_i x_5) - y_i,$$

where $i = 1, \cdots, 13$, $t_i = (0.1)i$ and

$$y_i = exp(-t_i) - 5exp(-10t_i) + 3exp(-4t_i).$$

Test problem 5 (Brown and Dennis, 1971)

$$f_i(x) = (x_1 + t_i x_2 - exp(t_i))^2 + (x_3 + x_4 sin(t_i) - cos(t_i))^2,$$

Test problem 6 (El-Attar 5.1, 1979)

$$f_1(x) = x_1^2 + x_2 - 10$$
$$f_2(x) = x_1 + x_2^2 - 7$$
$$f_3(x) = x_1^2 - x_2^3 - 1$$

Test problem 7 (El-Attar 5.2)

$$f_1(x) = x_1^2 + x_2^2 + x_3^2 - 1$$
$$f_2(x) = x_1^2 + x_2^2 + (x_3 - 2)^2$$
$$f_3(x) = x_1 + x_2 + x_3 - 1$$
$$f_4(x) = x_1 + x_2 - x_3 + 1$$
$$f_5(x) = 2x_1^3 + 6x_2^2 + 2(5x_3 - x_1 + 1)^2$$
$$f_6(x) = x_1^2 - 9x_3$$

Test problem 8 (Madsen, see Overton and Murray, 1981)

$$f_1(x) = x_1^2 + x_2^2 + x_1 x_2$$
$$f_2(x) = sin(x_1)$$
$$f_3(x) = cos(x_2)$$

Test problem 9 (Osborne 1, 1972)

$$f_i(x) = y_i - (x_1 + x_2 exp(-t_i x_4) + x_3 exp(-t_i x_5))$$

where $i = 1, 2, \cdots, 33$, $t_i = 10(i-1)$, and

i	y_i	i	y_i	i	y_i
1	0.844	12	0.718	23	0.478
2	0.908	13	0.685	24	0.467
3	0.932	14	0.658	25	0.457
4	0.936	15	0.628	26	0.448
5	0.925	16	0.603	27	0.438
6	0.908	17	0.580	28	0.431
7	0.881	18	0.558	29	0.424
8	0.850	19	0.538	30	0.420
9	0.818	20	0.522	31	0.414
10	0.784	21	0.506	32	0.411
11	0.751	22	0.490	33	0.406

Test problem 10 (Osborne 2)

$$\begin{aligned} f_i(x) =& y_i - \big(x_1 exp(-t_i x_5) + x_2 exp(-(t_i - x_9)^2 x_6) \\ &+ x_3 exp(-(t_i - x_{10})^2 x_7) + x_4 exp(-(t_i - x_{11})^2 x_8)\big) \end{aligned}$$

where $i = 1, 2, \cdots, 65$, $t_i = (i-1)/10$, and

i	y_i	i	y_i	i	y_i
1	1.366	23	0.694	45	0.672
2	1.191	24	0.644	46	0.708
3	1.112	25	0.624	47	0.633
4	1.013	26	0.661	48	0.668
5	0.991	27	0.612	49	0.645
6	0.885	28	0.558	50	0.632
7	0.831	29	0.533	51	0.591
8	0.847	30	0.495	52	0.559
9	0.786	31	0.500	53	0.597
10	0.725	32	0.423	54	0.625
11	0.746	33	0.395	55	0.739
12	0.679	34	0.375	56	0.710
13	0.608	35	0.372	57	0.729
14	0.655	36	0.391	58	0.720
15	0.616	37	0.396	59	0.636
16	0.606	38	0.405	60	0.581
17	0.602	39	0.428	61	0.428
18	0.626	40	0.429	62	0.292
19	0.651	41	0.523	63	0.162
20	0.724	42	0.562	64	0.098
21	0.649	43	0.607	65	0.054
22	0.649	44	0.653		

Test problem 11 (Powell, 1962)

$$f_1(x) = x_1 + 10x_2$$
$$f_2(x) = 5^{1/2}(x_3 - x_4)$$
$$f_3(x) = (x_2 - 2x_3)^2$$
$$f_4(x) = 10^{1/2}(x_1 - x_4)^2$$

Test problem 12 (Rosenbrock, 1960)

$$f_1(x) = 10(x_2 - x_1^2)$$
$$f_2(x) = 1 - x_1$$

Test problem 13 (Watson, see Kowalik and Osborne, 1968)

$$f_i(x) = \sum_{j=2}^{n}(j-1)x_j t_i^{j-2} - \left(\sum_{j=1}^{n} x_j t_i^{j-1}\right)^2 - 1,$$

where $i = 1, \cdots, 29$, $t_i = i/29$, $f_{30}(x) = x_1$ and $f_{31}(x) = x_2 - x_1^2 - 1$.

Test problem 14 (Wood, see Colville, 1968)

$$f_1(x) = 10(x_2 - x_1^2)$$
$$f_2(x) = 1 - x_1$$
$$f_3(x) = 90^{1/2}(x_4 - x_3^2)$$
$$f_4(x) = 1 - x_3$$
$$f_5(x) = 10^{1/2}(x_2 + x_4 - 2)$$
$$f_6(x) = 10^{-1/2}(x_2 - x_4)$$

Appendix B

```
"nlrq" <-
function(x, y, model, t, theta, k = 2, eps = 1e-06, big = 1e+20, eta = 0.97)
{
# This is a function to compute nonlinear l1 estimate.
#    Input
#       model - user-provided function which returns components
#               f=(f_i (x_i , t)
#               J=(grad f_i )
#       t     - vector of initial values of the unknown parameters
#       theta - desired quantile
#       k     - number of Meketon's iterations used to calculate the
#               dual step
#       eps   - small positive number
#       big   - big positive number
#       eta   - 0.97
#    Output
#       t     - vector of estimated parameters
#       f     - function value at minimum
#
        n <- length(y)
        zero <- rep(0, n)
        w <- zero
        d <- rep(1, n)
        m <- model(x, y, t, theta)
        snew <- sum(abs(m$f))
        sold <- big
        while(sold - snew > eps) {
                z <- mekrq(m$J, m$f, w, theta, k, int = F, eps, big, eta)
                step <- z$coef
                # Calculate an optimal step length lambda
                lambda <- step.length(t,step)
                t <- t + lambda * step
                m <- model(x, y, t, theta)
                sold <- snew
                snew <- sum(abs(m$f))
                w <- z$w
                w <- lsfit(m$J, w, int = F)$resid
                if(max(abs(w)) >= 1)
                        w <- w/(max(abs(w)) + eps)
        }
        f <- snew
        return(t, f)
}

"mekrq" <-
function(x, y, w, theta, kmax = 1000, int = T, eps, big, eta)
{
# Compute linear regression quantile estimate (Meketon, 1985).
# However, note that W is not initialized and the maximum number of
# iteration is given by kmax.
#
        if(int == T) x <- cbind(1, x)
        sr <- big
        k <- 1
        while(k <= kmax & sr - crossprod(y, w) > eps) {
                d <- pmin(theta - w, 1 - theta + w)
                z <- lsfit(x, y, d^2, int = F)
```

```
            sr <- sum(abs(z$resid))
            k <- k + 1
            s <- z$resid * d^2
            alpha <- max(pmax(s/(theta - w),  - s/(1 - theta + w)))
            w <- w + (eta/alpha) * s
        }
        coef <- z$coef
        return(coef, w)
}
```

```
/*
**  This procedure is a function to compute nonlinear regression
**  quantile estimate.
**
**  PROC NLRQ
**  FORMAT
**     { t,f } = nlrq( x,y,&model,start,theta,k,eps,big,eta )
**  INPUT
**     model - user-provided function which returns components
**             f = (f_i (x_i,t))
**             J = (grad f_i)
**     start - vector of initial values of the unknown parameters
**     theta - desired quantile
**         k - number of Meketon's iterations used to calcuate
**             the dual step
**       eps - small positive number (1e-06)
**       big - big positive number (1e+20)
**       eta - 0.97
**  OUTPUT
**         t - vector of estimated parameters
**         f - function value at minimum
**
 *
 *    The following procedure is a function to compute linear
 *     regression quantile estimate (Meketon, 1985). However, note
 *    that W is not initialized and the maximum number of
 *    iterations is given by kmax.
 */
      proc(2) = mekrq(x,y,w,theta,kmax,inter,eps,big,eta);
           local sr,k,wy,wx,d,t,r,s,alpha;
           if inter eq 1;
              x = ones(rows(x),1)~x;
           endif;
           sr = big;
           k = 1;
           do while k <= kmax and sr - y'*w > eps;
               d = minc((theta-w)'|(1-theta+w)');
               d = vec(d);
               wx = x.*d;
               wy = y.*d;
               { t } = olsqr(wy,wx);
               r = y - x*t;
               sr = sumc(abs(r));
               k = k + 1;
               s = r.*d^2;
               alpha = maxc(maxc((s./(theta-w))'|
                       (-s./(1-theta+w))'));
               w = w + eta/alpha*s;
           endo;
           retp(t,w);
      endp;
/*
 *
 */
```

```
proc(2) = nlrq(x,y,&model,t,theta,kmax,eps,big,eta);
    local n,d,w,f,J,snew,sold,step,lambda,t,t1,p,inter,
    model:proc;
    n = rows(y);
    w = zeros(n,1);
    d = ones(n,1);
    { f,J } = model(x,y,t);
    snew = sumc(abs(f));
    sold = big;
    inter = 0;
    do while sold - snew > eps;
        { step,w } = mekrq(J,f,w,theta,kmax,inter,eps,
             big,eta);
        /* Calculate an optimal step length lambda */
        { lambda } = stepl(J,f,t,step,&model,eps);
        t = t + lambda*step;
        { f,J } = model(x,y,t);
        sold = snew;
        snew = sumc(abs(f));
        { t1,w,p } = olsqr2(w,J);
        if maxc(abs(w)) >= 1;
           w = w/(maxc(abs(w))+eps);
        endif;
    endo;
    retp(t,snew);
endp;
```