MATHEMATICS OF COMPUTATION, VOLUME 27, NUMBER 121, JANUARY, 1973

A Comparison of Algorithms for Rational l_∞ Approximation

By C. M. Lee and F. D. K. Roberts

Abstract. Results are reported of a numerical study to compare eight algorithms for obtaining rational l_∞ approximations. The algorithms investigated are Loeb's algorithm, the linear inequality algorithm, the Osborne-Watson algorithm, the differential correction algorithms I, II and III, the Remes algorithm and Maehly's algorithm. The results of the study indicate that the Remes algorithm and the differential correction algorithm III are the most satisfactory methods to use in practice.

1. Introduction. Let $f(x)$ be a given real-valued function defined on a discrete point set $X = \{x_1, x_2, \cdots, x_N\}$. Given nonnegative integers m and n, we form a rational approximating function

$$R(x) = P(x)/Q(x) = \sum_{i=0}^{m} p_i x^i \Big/ \sum_{j=0}^{n} q_j x^j. \tag{1}$$

The rational l_∞ approximation problem is to determine the coefficients p_i^* ($i = 0, 1, \cdots, m$) and q_j^* ($j = 0, 1, \cdots, n$) which minimize the expression

$$\max_{1 \le t \le N} |f(x_t) - R(x_t)| = ||f - R||_\infty, \tag{2}$$

subject to the conditions

$$Q(x_t) > 0, \qquad t = 1, 2, \cdots, N.$$

Since expression (1) is homogeneous in the coefficients p_i and q_j, we may impose a normalization condition, for example $\max_j |q_j| = 1$.

Existence of a solution to this problem is not guaranteed in general. However, for the purpose of this study, we shall assume that $f(x)$ is such that a best approximation exists. We shall assume that the best approximation $R^*(x)$ is expressed in an irreducible form, i.e., $P^*(x)$ and $Q^*(x)$ do not have a common factor. Uniqueness of the best approximation and the following characterization theorem are given in Rivlin [19, p. 131].

THEOREM. *$R^*(x) = P^*(x)/Q^*(x)$ is the best approximation to $f(x)$ on X if and only if $f(x) - R^*(x)$ has an alternating set consisting of at least $2+\max(n+\partial p, m+\partial q)$ points of X, where ∂p and ∂q denote the degrees of $P^*(x)$ and $Q^*(x)$ respectively.*

Barrodale [1] has shown that expression (2) defines a strictly quasi-convex function on the domain $Q(x_t) > 0$, $t = 1, 2, \cdots, N$, and thus, any local minimum to the

Received September 20, 1971, revised June 9, 1972.
AMS (MOS) subject classifications (1969). Primary 4117; Secondary 6520, 6530.
Key words and phrases. Rational approximation, linear programming.

approximation problem is necessarily a global minimum. Various algorithms have been suggested for determining best approximations (see for example Cheney and Southard [7], Rice [18, p. 102], Cheney [4, p. 169]). These fall into two categories. Algorithms in the first category determine best approximations by using the above characterization theorem. The algorithms of Remes and Maehly are of this type. Algorithms in the second category attempt to determine best approximations by solving the following nonlinear programming problem:

$$\text{minimize } w$$

$$\left.\begin{array}{l} \text{subject to } \left| f(x_t) - \sum_{i=0}^{m} p_i x_t^i \Big/ \sum_{j=0}^{n} q_j x_t^j \right| \leqq w \\ \qquad\qquad \sum_{j=0}^{n} q_j x_t^j \geqq 0 \end{array}\right\} \quad t = 1, 2, \cdots, N, \tag{3}$$

$$\max_i |q_i| = 1.$$

Loeb's algorithm, the linear inequality algorithm, the Osborne-Watson algorithm, and the three differential correction algorithms are of this type. These methods solve (3) as a series of linear programming problems.

In this paper, we report results of a numerical study to compare the computational behaviour of these various algorithms. In the next section, we give a brief description of the eight algorithms we have selected for the study. The third section contains details of the study and comments on the results. The fourth section discusses degeneracy, and the last section is concerned with the conclusions.

2. Algorithms.

2.1. *Loeb's Algorithm.* The approximation problem is to minimize

$$\max_{1 \leq t \leq N} |f(x_t) - R(x_t)|,$$

which may be rewritten as

$$\max_{1 \leq t \leq N} \left\{ \frac{1}{|Q(x_t)|} |f(x_t)Q(x_t) - P(x_t)| \right\}.$$

Loeb [13] proposes the following iterative scheme: At the kth stage, polynomials $P^k(x)$ and $Q^k(x)$ are determined which minimize

$$\max_{1 \leq t \leq N} \left\{ \frac{1}{|Q^{k-1}(x_t)|} |f(x_t)Q(x_t) - P(x_t)| \right\}.$$

The term $1/|Q^{k-1}(x_t)|$ acts as a known weight factor and the problem is a linear approximation problem which can be solved by the techniques of linear programming. The normalization is accomplished by setting $q_0 = 1$ at each stage. The algorithm is not guaranteed to converge in general. Computational experience with the algorithm in l_1, l_2 and l_∞ norms is reported by Barrodale and Mason [2].

2.2. *The Linear Inequality Algorithm.* The approximation problem may be stated in the following way: Find the smallest value of w such that the following system of inequalities is consistent:

$$\left.\begin{aligned} |f(x_t) - R(x_t)| &\leqq w \\ -Q(x_t) &\leqq 0 \end{aligned}\right\} \quad t = 1, 2, \cdots, N.$$

If the approximation is normalized by setting $q_0 = 1$, then the system may be written as

$$\left.\begin{aligned} [f(x_t) - w] \sum_{j=1}^{n} q_j x_t^j - \sum_{i=0}^{m} p_i x_t^i &\leqq w - f(x_t) \\ [-f(x_t) - w] \sum_{j=1}^{n} q_j x_t^j - \sum_{i=0}^{m} p_i x_t^i &\leqq w + f(x_t) \\ -\sum_{j=1}^{n} q_j x_t^j &\leqq 1 \end{aligned}\right\} \quad t = 1, 2, \cdots, N.$$

This system of inequalities is nonlinear. However, if w is assigned a fixed value then the system becomes linear, and the feasibility may be determined by linear programming. Since the error w^* corresponding to the best approximation lies in the interval $[0, \max_t |f(x_t)|]$, it may be located by the method of bisection. At each iteration, the value of w is chosen to be the midpoint of the interval obtained at the previous iteration. If the system is consistent, the search is restricted to the lower half of the interval. If the system is inconsistent, the search is restricted to the upper half. The method is due to Loeb [14]. The rate of convergence of this algorithm is slow since at each iteration, the interval containing w^* is only reduced by a factor $\frac{1}{2}$. The normalization $q_0 = 1$ may for some problems conflict with the inequalities $Q(x_t) \geqq 0$, $t = 1, 2, \cdots, N$. In these cases, other normalizations may be used (for example $q_0 = -1$).

2.3. *The Osborne-Watson Algorithm.* Osborne and Watson [16] present a general algorithm for solving the nonlinear l_∞ approximation problem. Watson [20] discusses the application of this algorithm to rational l_∞ approximation. The algorithm is an iterative scheme which at the kth iteration determines polynomials $\delta P^k(x)$ and $\delta Q^k(x)$ which minimize the expression

$$\max_{1 \leqq t \leqq N} \left| f(x_t) - \frac{P^{k-1}(x_t)}{Q^{k-1}(x_t)} + \frac{P^{k-1}(x_t)\,\delta Q(x_t) - Q^{k-1}(x_t)\,\delta P(x_t)}{Q^{k-1}(x_t)^2} \right|. \tag{4}$$

The kth approximation $P^k(x)/Q^k(x)$ is given by

$$\frac{P^k(x)}{Q^k(x)} = \frac{P^{k-1}(x) + \lambda^k\,\delta P^k(x)}{Q^{k-1}(x) + \lambda^k\,\delta Q^k(x)},$$

where λ^k is chosen to minimize the expression

$$\max_{1 \leqq t \leqq N} \left| f(x_t) - \frac{P^{k-1}(x_t) + \lambda\,\delta P^k(x_t)}{Q^{k-1}(x_t) + \lambda\,\delta Q^k(x_t)} \right|. \tag{5}$$

Expression (4) is obtained by making a linear approximation to expression (2). The minimization of (4) can be accomplished by linear programming. The normalization is accomplished by setting $q_0 = 1$. The minimization of (5) is approximately obtained by searching for values of λ on the set 0(.1)1, except where this gives a value of zero when a more accurate value is obtained. Convergence of the method is discussed by Osborne and Watson [16].

2.4. *The Differential Correction Algorithm* I. The differential correction algorithm was first discussed by Cheney and Loeb [5]. However, in the later literature, the algorithm is described in a slightly modified form. The later version of the algorithm which is described by Cheney and Loeb [6], Cheney and Southard [7], Rice [18, p. 116], Cheney [4, p. 171], we shall refer to as the differential correction algorithm I. The original version will be referred to as the differential correction algorithm II.

The algorithm is an iterative scheme. At the kth stage, an approximation $P^{k-1}(x)/Q^{k-1}(x)$ is available with error w_{k-1}. The algorithm determines polynomials $P^k(x)$ and $Q^k(x)$ which minimize the expression

$$\max_{1 \leq t \leq N} \{|f(x_t)Q(x_t) - P(x_t)| - w_{k-1}Q(x_t)\}, \tag{6}$$

subject to the normalization that the coefficients of $Q(x)$ are bounded by 1 in modulus. This minimization can be accomplished by linear programming. The method is guaranteed to converge to the best approximation from any initial approximation with positive denominator in at least a linear rate (Cheney [4, p. 171]).

2.5. *The Differential Correction Algorithm* II. The version of the differential correction algorithm presented by Cheney and Loeb [5] has recently been studied by Barrodale, Powell and Roberts [3]. The algorithm differs from the previous algorithm in that expression (6) is replaced by

$$\max_{1 \leq t \leq N} \left\{\frac{|f(x_t)Q(x_t) - P(x_t)| - w_{k-1}Q(x_t)}{Q^{k-1}(x_t)}\right\}. \tag{7}$$

With both versions of the algorithm, the constraints $Q(x_t) > 0$, $t = 1, 2, \cdots, N$, are maintained automatically and hence need not be incorporated into the linear programming formulation of the minimization of (6) and (7). The proof of convergence and the following theorem are given in [3].

THEOREM. *If $N \geqq m + n + 1$, if a best approximation exists, and if the best approximation is not degenerate, then the rate of convergence of the algorithm is at least quadratic.*

2.6. *The Differential Correction Algorithm* III. Computational experience with the differential correction algorithm II indicates that even though the ultimate convergence rate is quadratic, the convergence rate in the early iterations can be slow. The method can be substantially improved if a good initial approximation is available. We have therefore tried the algorithm using as initial approximations the polynomials $P^1(x)$ and $Q^1(x)$ which minimize the expression

$$\max_{1 \leq t \leq N} \{|f(x_t)Q(x_t) - P(x_t)|\},$$

subject to the normalization $q_0 = 1$. This linear problem is identical to the first iteration of Loeb's algorithm, and our numerical study indicates that the solution frequently yields an excellent approximation. Provided that $Q^1(x)$ does not change sign on the discrete point set X, after a suitable normalization, $P^1(x)$ and $Q^1(x)$ may be used as input to the differential correction algorithm II. In the cases where $Q^1(x)$ does change sign, we have initiated the algorithm with the values $P^1(x) = 0$, $Q^1(x) = 1$.

2.7. *The Remes Algorithm.* This is perhaps the most popular method for ob-

taining rational approximations. There are many variations of the algorithm (see for example Rice [18, p. 109], Fraser and Hart [11], Ralston [17, p. 301], Werner [21]). The version we consider assumes that the best approximation $R^*(x)$ is not degenerate, and, hence, by the characterization theorem, the error function $\epsilon^*(x) = f(x) - R^*(x)$ alternates at least $m + n + 2$ times. A reference set X^k is defined to be a set of $m + n + 2$ distinct ordered points $(y_1^k, y_2^k, \cdots, y_{m+n+2}^k)$ of X. The algorithm is an iterative scheme which is implemented in two stages.

(i) Given a reference set X^{k-1} at the $(k-1)$st stage, an approximation $R^k(x)$ is obtained such that its error function $\epsilon^k(x) = f(x) - R^k(x)$ alternates $m + n + 2$ times on X^{k-1}.

(ii) The extreme points of the error function $\epsilon^k(x)$ yield a new reference set X^k.

The second stage of the algorithm is straightforward in the discrete case since the new reference set can be obtained by a direct search over the N points of X. The first stage requires the solution of the following system of nonlinear equations

$$f(y_i) - R(y_i) = (-1)^i\lambda, \qquad i = 1, 2, \cdots, m + n + 2, \tag{8}$$

or, equivalently,

$$f(y_i)Q(y_i) - P(y_i) = (-1)^i\lambda Q(y_i), \qquad i = 1, 2, \cdots, m + n + 2. \tag{9}$$

These equations are normalized by setting $q_0 = 1$, and the solution is obtained by Newton's method.

2.8. *Maehly's Second Algorithm.* This method was proposed by Maehly [15]. The algorithm assumes that the error function $\epsilon^*(x)$ of the best approximation has exactly $m + n + 1$ zeros $z_1^*, z_2^*, \cdots, z_{m+n+1}^*$. It may therefore be written in the form

$$\epsilon^*(x) = G(x) \prod_{l=1}^{m+n+1} (x - z_l^*),$$

where $G(x)$ is a positive (or negative) function. The method is an iterative scheme which is implemented in two stages.

(i) Let $x_1 < z_1 < z_2 < \cdots < z_{m+n+1} < x_N$ be estimates of the zeros of the error curve $\epsilon^*(x)$. An approximation $R(x)$ is obtained by solving the equations

$$R(z_l) = f(z_l), \qquad l = 1, 2, \cdots, m + n + 1. \tag{10}$$

(ii) The extreme points $x_1, x_2, \cdots, x_{m+n+2}$ of the error function $\epsilon(x) = f(x) - R(x)$ are then used to obtain corrections $\delta z_1, \delta z_2, \cdots, \delta z_{m+n+1}$ to the zeros.

The system (10) is normalized by setting $q_0 = 1$ and may be solved as a system of linear equations. The corrections to the zeros in the second stage are obtained by solving the linear equations

$$\sum_{l=1}^{m+n+1} \frac{\delta z_l}{(x_i - z_l)} = \log |\epsilon(x_i)| - \log |\lambda|, \qquad i = 1, 2, \cdots, m + n + 2.$$

Convergence of the method is discussed by Dunham [8], [9].

3. Numerical Results. To test the algorithms numerically, we selected 11 data sets (see Table 1), and approximated each data set by rational functions P_0/Q_2, P_1/Q_1, P_2/Q_2, P_1/Q_3, P_4/Q_2, where the subscripts denote the degrees of the polynomials. The six algorithms which require linear programming techniques were

solved by applying the revised simplex method (see for example Gass [12, p. 106]) to the dual formulation of the linear programming problem. The two algorithms which require the solution of a system of linear equations were solved using the method given in Forsythe and Moler [10, p. 68]. All algorithms were programmed in FORTRAN and run in double precision arithmetic (16 digits) on an IBM 360/50. The initial approximations for Loeb's algorithm, the Osborne-Watson algorithm, and the differential correction algorithms I and II were taken to be $P^0(x) = 0$, $Q^0(x) = 1$. The initial reference set for the Remes algorithm was taken to be the points of X closest to the extrema of the $(m + n + 1)$st Chebyshev polynomial shifted to the interval $[x_1, x_N]$, and the initial zeros for Maehly's algorithm were taken to be the zeros of this polynomial. The algorithms (except the Remes algorithm) were terminated when either the relative change in error in two successive iterations was less than 10^{-7}, or after 50 iterations. The Remes algorithm was terminated when two successive reference sets were identical.

Complete details of the numerical study appear in the microfiche section of this issue. Tables 2.1–2.8 record the number of iterations and the central processor time required for convergence of each of the algorithms, and also indicate the examples for which an algorithm fails to produce the best approximation. Table 3 of the microfiche section records the errors of best approximation.

Table 4 attempts to summarize these results. We list the average central processor time required by each algorithm to solve each of the 55 examples, and also the number

TABLE 1
Data Sets Used in the Numerical Study

	Function $f(x)$	$[x_1, x_N]$	*Number of Points (Equally Spaced)*
f_1:	e^x	$[-1, 1]$	51
f_2:	$\sin(x)$	$[-3, 3]$	21
f_3:	$\sqrt{x}$	$[0, 1]$	11
f_4:	1	$[0, 0.5)$	21
	0	$x = 0.5$	
	-1	$(0.5, 1]$	
f_5:	x	$[0, 1]$	51
	$0.5x + 0.4$	$(1, 2]$	
f_6:	e^x	$[0, 1]$	21
	$e^{-x} - e^{-1} + e$	$(1, 2]$	
f_7:	$\log(1 + x)$	$[0, 1]$	51
f_8:	$\operatorname{erf}(x)$	$[0, 2]$	21
f_9:	e^{-x^2}	$[0, 2]$	11
f_{10}:	$\Gamma(x)$	$[2, 3]$	51
f_{11}:	$\Gamma(x)$	$[2, 3]$	101

of examples for which an algorithm fails to produce the best approximation. We make the following comments on the results:

Loeb's Algorithm. For the smooth data sets, this algorithm usually converges very rapidly. However, the algorithm may fail to converge, may converge very slowly, and may converge to an approximation with a pole in the interval $[x_1, x_N]$. More seriously, the algorithm may converge to a pole-free approximation which is not the best approximation. Conditions under which the algorithm is convergent are unknown.

The Linear Inequality Algorithm. At each iteration of this algorithm, the interval containing w^* is only reduced by a factor $\frac{1}{2}$, and hence the convergence is quite slow. We encountered one minor difficulty with the algorithm when approximating $\Gamma(x)$ (f_{10} and f_{11}) by P_1/Q_3. The best approximation using 101 points (f_{11}) is

$$\frac{-0.04076 - 0.23422x}{1 - 1.76828x + 0.64651x^2 - 0.06983x^3}.$$

The denominator in this expression is negative in the interval [2, 3]. Using the algorithm with the normalization $q_0 = 1$ and the constraints $Q(x_t) \geqq 0$, $t = 1, 2, \cdots, N$, produces an incorrect solution. However, using the normalization $q_0 = -1$ produces the best approximation.

The Osborne-Watson Algorithm. The ultimate convergence rate for this algorithm appears to be quadratic, although, in the early iterations, the convergence can be fairly slow. We encountered one example where the algorithm converges to an approximation with a pole in $[x_1, x_N]$. When the linear constraints $Q(x_t) \geqq 0$, $t = 1, 2, \cdots, N$, are included in the linear programming formulation of (4), the algorithm converges to the best approximation for this example. Also, when approximating $\Gamma(x)$ (f_{10} and f_{11}) by P_1/Q_3, the algorithm does not converge to the best approximation. In order to obtain convergence, it is necessary to change the values of λ used in the linear search from 0(.1)1 to -1(.1)1.

The Differential Correction Algorithm I. The convergence of this method is

TABLE 4
Summary of Numerical Results

Algorithm	*Average Central Processor Time in Seconds*	*Number of Failures (Total Number of Examples = 55)*
Loeb	19.2	9
Linear Inequality	46.4	0
Osborne-Watson	17.0	1
Differential Correction I	65.9	0
Differential Correction II	21.5	0
Differential Correction III	12.8	0
Remes	2.8	5
Maehly	3.9	24

guaranteed. However, the convergence rate is quite slow, and we encountered 13 cases where the algorithm does not converge in 50 iterations.

The Differential Correction Algorithm II. This method is guaranteed to converge from any starting approximation with positive denominator, and the ultimate convergence rate is quadratic. In the early iterations, however, the convergence can be slow. The errors obtained in successive iterations, when $\Gamma(x)$ (f_{11}) is approximated by P_2/Q_2, are typical of the convergence of the algorithm.

Iteration	*Error*	*Iteration*	*Error*
1	0.100000000 (1)	7	0.463845537 (−3)
2	0.376442069 (0)	8	0.126144341 (−3)
3	0.114598168 (0)	9	0.370570234 (−4)
4	0.435131594 (−1)	10	0.364318280 (−4)
5	0.840774023 (−2)	11	0.364317143 (−4)
6	0.147065926 (−2)	12	0.364317143 (−4)

The Differential Correction Algorithm III. This algorithm appears to be the most satisfactory of the linear programming methods. For the smooth data sets, the initial iteration usually gives an excellent starting approximation for the algorithm. We encountered 11 cases where the initial iteration produces an approximation with a pole in $[x_1, x_N]$. In these cases, the algorithm was restarted from the approximation $P^1(x) = 0$, $Q^1(x) = 1$. We list below the errors obtained in successive iterations when $\Gamma(x)$ (f_{11}) is approximated by P_2/Q_2.

Iteration	*Error*
1	0.436948626 (−4)
2	0.364453365 (−4)
3	0.364317144 (−4)
4	0.364317143 (−4)

These results are typical of the fast convergence obtained with this algorithm.

The Remes Algorithm. The convergence of this method is very rapid; the reference set for the best approximation usually being obtained in a few iterations. The two cases where the algorithm fails to converge are due to degeneracy, corresponding to an error of best approximation which alternates less than $m + n + 2$ times. We encountered three cases where the algorithm converges to a solution with a pole in the interval $[x_1, x_N]$.

Maehly's Algorithm. Of the algorithms we tested, this method is the least successful. For the smooth data sets, the algorithm frequently converges quite rapidly. In the cases where the method fails to converge, this is due either to an iteration producing an error function which alternates more (or less) than $m + n + 2$ times on X, or the corrections to the zeros lying outside the interval $[x_1, x_N]$.

4. Degeneracy. Degeneracy occurs when the degrees of $P^*(x)$ and $Q^*(x)$ are strictly less than m and n, respectively. The error function $f(x) - R^*(x)$ may then alternate less than $m + n + 2$ times. This can cause difficulties with the popular Remes algorithm. Rice [18, p. 77] states "Some of the other computational methods proposed for rational approximation do not, in theory, encounter these difficulties,

but this has not yet been verified by actual experiment". To test the algorithms on a near degenerate example, we consider the approximation of $\Gamma(x)$ by P_1/Q_2, using 101 equally spaced points in [2, 3]. The best approximation by P_0/Q_1 has three error alternations, and almost alternates one extra time. The best approximation by P_1/Q_2 is

$$\frac{0.49405 - 0.16436x}{1 - 0.58424x + 0.08369x^2}$$

with error 0.56739 (−2). The numerator and denominator in this approximation have factors $(x - 3.00598)$ and $(x - 3.00604)$ respectively, and thus the approximation is nearly degenerate. The Remes algorithm, Maehly's algorithm and Loeb's algorithm all fail to produce this best approximation. The five other algorithms all converge without apparent difficulty. The differential correction algorithms II and III, for example, produce the five error alternations correct to twelve decimal places.

An example frequently quoted in the literature of a degenerate approximation is the approximation of $\Gamma(x)$ in the interval [1.9507, 3]. Many authors state that the best approximation by P_0/Q_1 has four error alternations *in the continuous case*, and thus the best approximation by P_1/Q_2 is degenerate. However, this is false. The value 1.9507 is obtained by considering the error of the best approximation to $\Gamma(x)$ in the interval [2, 3] by P_0/Q_1, and then locating the point outside this interval at which the fourth alternation occurs. Unless this point is identified *exactly*, however, the best approximation will only alternate three times. Our calculations indicate that the correct value (to 10 decimal points) is 1.9507929092. Using the value 1.9507 (or 1.95), for example, produces only three alternations (and nearly four). Thus, the best approximation by P_1/Q_2 is not degenerate, although it is very nearly degenerate.

In the discrete case, using 101 equally spaced points in [1.9507, 3], the best approximation by P_0/Q_1 alternates three times, and almost alternates one extra time. Thus, the best approximation by P_1/Q_2 cannot be degenerate. In fact, computational experience with this problem indicates that no best approximation exists! Watson [20] reports results for the Osborne-Watson algorithm applied to this example. His results are only recorded to six decimal places and so we have repeated his experiment to obtain more accuracy. The approximation we obtain has a pole at the point $x = 2.98705$, which is clearly unacceptable. The five error alternations are obtained to eleven decimal places. We strongly believe that no (pole-free) best approximation exists for this problem.

To test the linear programming algorithms on a degenerate problem, we consider the approximation of $f(x) = 3/(1 + 2x) + g(x)$ using 101 equally spaced points in [0, 1] by P_1/Q_2. The values $g(x_i)$ are $+1$ at the 11th and 31st points, -1 at the 21st and 41st points, and uniform random numbers in $[-\frac{1}{2}, \frac{1}{2}]$ at all other points. The best approximation by P_0/Q_1 is $3/(1 + 2x)$ with error 1. This approximation has 4 error alternations and thus the best approximation by P_1/Q_2 is also $3/(1 + 2x)$, which is degenerate with defect 1. Loeb's algorithm does not converge for this problem. The linear inequality algorithm converges to the approximation (within round-off error) $\{3(1 - x)\}/\{(1 + 2x)(1 - x)\}$. The differential correction algorithms I, II, III converge to the approximation $\{3(1 + x)\}/\{(1 + 2x)(1 + x)\}$. The Osborne-Watson algorithm experiences some numerical instability for this example. By terminating the method when the relative change in the error in two successive approximations

is less than 10^{-5} (in place of 10^{-7}) produces the approximation

$$\{3(1 - 0.94178x)\}/\{(1 + 2x)(1 - 0.94178x)\}.$$

It appears that the three differential correction algorithms are the only techniques which do not encounter difficulties with this degenerate example.

5. Conclusions. The results of this study indicate that the linear programming algorithms cannot compete with the Remes algorithm in terms of computer time. However, it does appear that cases which are awkward for the Remes algorithm do not present problems for some of the other algorithms. Of these methods, the differential correction algorithm III appears to be the most satisfactory.

Acknowledgement. We wish to thank the National Research Council of Canada for the financial support provided by NRC grant A7143.

Department of Computer Science
University of Alberta
Edmonton, Alberta, Canada

Department of Mathematics
University of Victoria
Victoria, B. C., Canada

1. I. BARRODALE, *Best Rational Approximation and Strict Quasi-Convexity*, M.R.C. Report 1157, University of Wisconsin, Madison, Wis., 1971.
2. I. BARRODALE & J. C. MASON, "Two simple algorithms for discrete rational approximation," *Math. Comp.*, v. 24, 1970, pp. 877–891.
3. I. BARRODALE, M. J. D. POWELL & F. D. K. ROBERTS, *The Differential Correction Algorithm for Rational l_∞ Approximation*, Math. Report No. 54, University of Victoria, Victoria, British Columbia, 1971.
4. E. W. CHENEY, *Introduction to Approximation Theory*, McGraw-Hill, New York, 1966. MR **36** #5568.
5. E. W. CHENEY & H. L. LOEB, "Two new algorithms for rational approximation," *Numer. Math.*, v. 3, 1961, pp. 72–75. MR **22** #12692.
6. E. W. CHENEY & H. L. LOEB, "On rational Chebyschev approximation," *Numer. Math.*, v. 4, 1962, pp. 124–127. MR **27** #2088.
7. E. W. CHENEY & T. H. SOUTHARD, "A survey of methods for rational approximation, with particular reference to a new method based on a formula of Darboux," *SIAM Rev.*, v. 5, 1963, pp. 219–231. MR **28** #1754.
8. C. B. DUNHAM, "Convergence problems in Maehly's second method," *J. Assoc. Comput. Mach.*, v. 12, 1965, pp. 181–186. MR **31** #5312.
9. C. B. DUNHAM, "Convergence problems in Maehly's second method. II," *J. Assoc. Comput. Mach.*, v. 13, 1966, pp. 108–113. MR **32** #6116.
10. G. FORSYTHE & C. B. MOLER, *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1967. MR **36** #2306.
11. W. FRASER & J. F. HART, "On the computation of rational approximations to continuous functions," *Comm. Assoc. Comput. Mach.*, v. 5, 1962, pp. 401–403.
12. S. I. GASS, *Linear Programming. Methods and Applications*, 3rd ed., McGraw-Hill, New York, 1969. MR **42** #1509.
13. H. L. LOEB, *On Rational Fraction Approximations at Discrete Points*, Convair Astronautics, Math. Preprint No. 9, 1957.
14. H. L. LOEB, "Algorithms for Chebyshev approximations using the ratio of linear forms," *J. Soc. Indust. Appl. Math.*, v. 8, 1960, pp. 458–465. MR **22** #10147.
15. H. J. MAEHLY & C. WITZGALL, "Methods for fitting rational approximations. II, III," *J. Assoc. Comput. Mach.*, v. 10, 1963, pp. 257–277. MR **28** #707.
16. M. R. OSBORNE & G. A. WATSON, "An algorithm for minimax approximation in the nonlinear case," *Comput. J.*, v. 12, 1969/70, pp. 63–68. MR **39** #6625.
17. A. RALSTON, *A First Course in Numerical Analysis*, McGraw-Hill, New York, 1965. MR **32** #8479.

18. J. R. RICE, *The Approximation of Functions.* Vol. 2: *Nonlinear and Multivariate Theory,* Addison-Wesley, Reading, Mass., 1969. MR **39** #5989.

19. T. J. RIVLIN, *An Introduction to the Approximation of Functions,* Blaisdell, Waltham, Mass., 1969. MR **40** #3126.

20. G. A. WATSON, "On an algorithm for nonlinear minimax approximation," *Comm. Assoc. Comput. Mach.,* v. 13, 1970, pp. 160–162.

21. H. WERNER, "Tschebyscheff-Approximation im Bereich der rationalen Funktionen bei Vorliegen einer guten Ausgangsnäherung," *Arch. Rational Mech. Anal.,* v. 10, 1962, pp. 205–219. MR **26** #1993.

A COMPARISON OF ALGORITHMS

FOR RATIONAL $\mathcal{L}_\infty$ APPROXIMATION

BY

C. M. LEE & F. D. K. ROBERTS

TABLES 2.1-2.8

and 3

TABLES 2.1-2.8

Results of numerical study

The following notation is used:

I - number of iterations required for convergence.

C.P. - central processor time in seconds.

F - algorithm fails to converge.

* - algorithm converges to an approximation with a pole in $[x_1, x_N]$.

** - algorithm converges to a pole-free non-best approximation.

TABLE 2.1

Loeb's Algorithm

	P_1/Q_1		P_0/Q_2		P_2/Q_2		P_1/Q_3		P_4/Q_2	
	I	C.P.	I	C.P.	I	C.P.	I	C.P.	I	C.P.
f_1	7	7.5	8	7.8	6	12.5	6	13.3	5	16.2
f_2	2	2.2	2	2.4	F		F		6	10.7
f_3	10	3.0	23	7.1	8	4.6	7	3.8	7	6.1
f_4	3*	2.0	3	2.5	6*	5.1	14	13.0	11	21.1
f_5	3**	5.2	41	43.7	F		14**	25.2	>50	>212.9
f_6	F		10	4.7	24	18.1	17*	17.3	15	28.6
f_7	5	5.9	23	24.2	5	9.2	4	8.4	5	16.7
f_8	11	6.2	25	10.9	8	7.0	6	5.1	6	8.8
f_9	14	3.9	13	4.1	8	4.6	9	4.9	6	5.9
f_{10}	7	8.9	5	5.9	6	11.8	6	11.8	5	16.9
f_{11}	7	15.7	5	11.1	6	24.0	6	23.4	5	31.4

TABLE 2.2

The Linear Inequality Algorithm

	P_1/Q_1		P_0/Q_2		P_2/Q_2		P_1/Q_3		P_4/Q_2	
	I	C.P.	I	C.P.	I	C.P.	I	C.P.	I	C.P.
f_1	31	17.6	30	25.7	39	70.3	38	60.1	47	110.9
f_2	24	8.8	24	7.8	25	22.5	25	32.5	31	38.6
f_3	29	4.5	26	6.6	34	11.8	32	10.9	40	19.9
f_4	24	7.1	24	6.6	26	16.7	26	22.2	28	42.1
f_5	28	26.6	26	22.0	28	38.2	29	88.4	30	362.3
f_6	27	6.5	27	10.4	29	23.7	29	30.5	30	47.5
f_7	33	21.4	27	31.2	42	64.1	40	54.1	51	114.9
f_8	28	8.0	26	12.0	33	18.5	34	19.3	38	38.7
f_9	28	5.5	28	5.0	32	12.2	32	12.7	38	20.1
f_{10}	32	20.1	32	22.2	39	61.9	39	59.8	47	131.0
f_{11}	32	36.8	32	42.3	39	130.6	39	131.2	47	276.7

TABLE 2.3

The Osborne-Watson Algorithm

	P_1/Q_1		P_0/Q_2		P_2/Q_2		P_1/Q_3		P_4/Q_2	
	I	C.P.	I	C.P.	I	C.P.	I	C.P.	I	C.P.
f_1	6	11.5	8	14.2	7	20.2	8	23.5	7	31.0
f_2	2	2.4	1	1.8	8	10.4	8	11.6	6	13.7
f_3	7	3.5	7	3.8	11	8.1	7	4.9	13*	14.6
f_4	2	2.8	1	1.5	9	10.6	7	8.4	13	27.0
f_5	6	10.7	7	12.1	8	20.8	7	23.8	10	52.1
f_6	7	5.6	6	5.3	7	9.4	10	14.5	7	14.1
f_7	6	11.3	8	14.5	8	24.9	6	16.1	8	38.8
f_8	7	5.9	7	5.9	7	10.0	7	9.7	7	14.0
f_9	6	3.5	8	4.2	9	7.0	8	6.9	6	7.6
f_{10}	7	11.9	8	12.4	8	24.2	9	38.0	7	34.7
f_{11}	7	21.3	8	23.1	8	46.3	11	82.8	7	63.8

TABLE 2.4

The Differential Correction Algorithm I

	P_1/Q_1		P_0/Q_2		P_2/Q_2		P_1/Q_3		P_4/Q_2	
	I	C.P.	I	C.P.	I	C.P.	I	C.P.	I	C.P.
f_1	23	25.6	37	40.8	31	83.4	47	122.4	>50	>278.2
f_2	2	2.1	1	1.8	28	34.2	27	35.0	17	40.3
f_3	28	12.0	22	9.4	>50	>54.6	39	50.4	>50	>94.1
f_4	3	2.4	1	1.2	>50	>63.3	>50	>140.2	>50	>239.5
f_5	13	17.0	33	34.8	41	87.1	20	54.3	>50	>221.8
f_6	24	15.2	25	13.2	>50	>69.1	>50	>75.6	>50	>166.4
f_7	16	14.9	29	40.8	28	103.9	21	48.4	39	247.7
f_8	24	11.5	24	14.5	39	42.7	30	48.3	>50	>100.3
f_9	14	5.0	47	14.7	48	2[illegible].8	>50	>49.4	>50	>60.3
f_{10}	20	26.4	19	21.1	19	52.5	20	62.3	19	91.8
f_{11}	20	49.7	19	37.3	19	92.8	20	116.3	19	158.6

TABLE 2.5

The Differential Correction Algorithm II

	P_1/Q_1		P_0/Q_2		P_2/Q_2		P_1/Q_3		P_4/Q_2	
	I	C.P.	I	C.P.	I	C.P.	I	C.P.	I	C.P.
f_1	7	8.1	7	10.4.	10	27.4	11	35.8	11	54.9
f_2	2	2.1	1	1.9	7	10.4	8	13.5	10	25.3
f_3	9	4.0	9	4.7	12	13.0	10	11.1	14	29.8
f_4	6	3.8	1	1.4	10	12.7	8	10.3	13	28.4
f_5	7	8.9	9	12.1	8	17.9	10	27.7	18	84.5
f_6	8	4.7	9	5.9	9	11.4	12	14.2	13	31.2
f_7	6	7.5	9	13.3	8	15.9	8	21.3	9	39.9
f_8	7	4.1	8	5.7	8	8.8	9	9.7	10	29.8
f_9	7	2.9	7	3.6	8	8.9	8	9.1	10	22.9
f_{10}	9	12.4	8	11.9	12	32.4	12	33.9	13	70.7
f_{11}	9	21.6	8	21.0	12	59.8	12	59.5	13	130.5

TABLE 2.6

The Differential Correction Algorithm III

	P_1/Q_1		P_0/Q_2		P_2/Q_2		P_1/Q_3		P_4/Q_2	
	I	C.P.	I	C.P.	I	C.P.	I	C.P.	I	C.P.
f_1	5	6.8	5	7.9	4	12.6	5	19.5	4	20.6
f_2	2	2.1	2	2.4	8	12.6	9	14.9	5	13.2
f_3	5	2.8	10	5.0	6	8.0	5	5.6	6	15.8
f_4	7	4.5	2	1.6	6	7.4	7	8.8	7	18.1
f_5	5	7.1	10	13.3	9	21.7	6	23.7	14	77.0
f_6	9	5.0	5	3.8	6	9.4	13	16.0	6	17.8
f_7	4	5.3	10	14.7	4	9.9	4	11.5	4	22.0
f_8	5	3.5	9	6.5	5	6.6	4	5.8	4	16.4
f_9	5	2.5	8	4.0	5	7.0	5	6.7	4	11.7
f_{10}	5	8.2	4	6.7	4	13.6	4	14.1	4	25.7
f_{11}	5	14.1	4	11.3	4	24.3	4	23.2	4	44.7

TABLE 27

The Remes Algorithm

	P_1/Q_1		P_0/Q_2		P_2/Q_2		P_1/Q_3		P_4/Q_2	
	I	C.P.	I	C.P.	I	C.P.	I	C.P.	I	C.P.
f_1	3	2.7	3	2.9	3	3.2	3	3.2	3	3.6
f_2	2	1.3	F		2*	2.0	2*	2.1	1	1.7
f_3	2	1.4	2	1.5	3	2.3	3	2.1	3	2.9
f_4	2	1.4	F		3	2.5	3	2.5	4	3.7
f_5	4	3.7	4	3.3	4*	4.4	4	4.1	6	7.3
f_6	2	1.8	3	2.0	4	3.3	3	2.7	5	4.7
f_7	2	2.2	3	2.8	2	2.4	2	2.4	3	3.9
f_8	2	1.6	2	1.7	3	2.3	3	2.2	2	2.3
f_9	2	1.6	2	1.6	3	2.2	3	2.1	2	2.2
f_{10}	2	2.2	2	2.2	2	2.6	2	2.6	2	3.1
f_{11}	3	4.1	3	4.1	2	3.7	3	4.7	2	3.8

TABLE 2.8

Maehly's Algorithm

	P_1/Q_1		P_0/Q_2		P_2/Q_2		P_1/Q_3		P_4/Q_2	
	I	C.P.	I	C.P.	I	C.P.	I	C.P.	I	C.P.
f_1	8	5.5	8	5.5	6	4.8	7	5.5	6	5.6
f_2	7	2.9	F		F		F		5	3.2
f_3	F		12	3.2	F		F		F	
f_4	F		F		F		F		F	
f_5	F		28	16.8	F		F		F	
f_6	F		F		F		F		F	
f_7	6	4.4	15	9.3	6	4.9	6	4.8	6	5.5
f_8	13	4.7	23	7.6	11	5.0	11	5.2	9	5.4
f_9	12	3.2	F		F		9	3.7	F	
f_{10}	6	4.2	6	4.3	6	4.9	6	4.7	6	5.8
f_{11}	6	7.3	6	7.1	5	6.7	6	8.1	5	7.3

TABLE 3

Errors of Best Approximation

	P_1/Q_1	P_0/Q_2	P_2/Q_2	P_1/Q_3	P_4/Q_2
f_1	0.20932(-1)	0.34791(-1)	0.86644(-4)	0.12392(-3)	0.21037(-6)
f_2	0.62542(0)	0.99749(0)	0.30608(0)	0.30608(0)	0.66482(-2)
f_3	0.36243(-1)	0.18078(0)	0.77019(-3)	0.37281(-2)	0.10202(-4)
f_4	0.81818(0)	0.10000(1)	0.26923(0)	0.26923(0)	0.70465(-1)
f_5	0.58916(-1)	0.22594(0)	0.54260(-1)	0.48581(-1)	0.18717(-1)
f_6	0.30372(0)	0.20697(0)	0.86504(-1)	0.95354(-1)	0.30919(-1)
f_7	0.85978(-3)	0.92869(-1)	0.17028(-5)	0.74224(-5)	0.58255(-8)
f_8	0.44085(-1)	0.19844(0)	0.13754(-2)	0.92931(-3)	0.44515(-4)
f_9	0.72164(-1)	0.69042(-1)	0.25586(-2)	0.41421(-2)	0.38213(-4)
f_{10}	0.64376(-2)	0.64307(-2)	0.36395(-4)	0.55096(-4)	0.17428(-6)
f_{11}	0.64420(-2)	0.64351(-2)	0.36432(-4)	0.55160(-4)	0.17660(-6)