

Comparison of Algorithms for Multivariate Rational Approximation

By Jackson N. Henry*

Abstract. Let F be a continuous real-valued function defined on the unit square $[-1, 1] \times [-1, 1]$. When developing the rational product approximation to F , a certain type of discontinuity may arise. We develop a variation of a known technique to overcome this discontinuity so that the approximation can be programmed. Rational product approximations to F have been computed using both the second algorithm of Remez and the differential correction algorithm. A discussion of the differences in errors and computing time for each of these algorithms is presented and compared with the surface fit approximation also obtained using the differential correction algorithm.

1. Introduction. Let F be a continuous real-valued function defined on $D \equiv [-1, 1] \times [-1, 1]$. There are several methods by which we may obtain uniform rational approximations to the function F on D (see [1], [4], [6], [7], [8]). The purpose of this paper is to describe briefly and then to compare several of these methods with respect to their accuracy and the computer time necessary to calculate the approximations. We will make comparisons between the use of the differential correction algorithm and the Remez algorithm in computing the rational product approximation as introduced by M. S. Henry and J. A. Brown [3] and improved by M. S. Henry and S. E. Weinstein [4]. These approximations will be compared to the differential correction algorithm when used as a method of surface fitting as given in Kaufman and Taylor [5].

In Section 2, we briefly describe the rational product approximation. Section 3 is devoted to a description of the manner in which we employ our implementation of the Remez algorithm and Kaufman and Taylor's implementation of the differential correction algorithm to obtain the rational product approximation. In Section 4 we discuss a certain discontinuity which can arise and a method to overcome it. We conclude with a comparison of the algorithms with respect to several examples.

2. Rational Product Approximation. This section presents a brief description of the rational product approximation of a continuous function F . For the theory connected with this definition, the reader is referred to [1], [3] and [4].

Let

$$R(C; x) = \frac{P(A; x)}{Q(B; x)} = \frac{\sum_{i=0}^m a_i x^i}{\sum_{j=0}^k b_j x^j} \neq 0,$$

Received August 19, 1975; revised July 20, 1976.

AMS (MOS) subject classifications (1970). Primary 65D15, 65D20.

*This research supported in part by NSF Grant MCS 76-06553.

Copyright © 1977, American Mathematical Society

where $C = (A; B) = (a_0, \dots, a_m; b_0, \dots, b_k)$ satisfies

- (i) $Q(B; x) > 0$ for all $x \in [-1, 1]$,
- (ii) $P(A; x)$ and $Q(B; x)$ have no common factors other than constants,
- (iii) $\max_{0 \leq j \leq k} |b_j| = 1$.

Definition. Let $\mathbf{R}(m, k)$ denote the class of rational functions consisting of all $R(C; x)$ satisfying (i), (ii), and (iii) above, and the zero function whose unique representation we take to be $R(C_0; x)$ where $C_0 = (0, \dots, 0; 1, 0, \dots, 0)$.

Definition. For each fixed $y \in [-1, 1]$, let $F_y(x)$ be defined by $F_y(x) = F(x, y)$ for $-1 \leq x \leq 1$. Let

$$R(C(y); x) = P(A(y); x)/Q(B(y); x)$$

denote the best uniform approximation to $F_y(x)$ from $\mathbf{R}(m, k)$. Here the best uniform approximation from $\mathbf{R}(m, k)$ means that $\|F_y(x) - R(\bar{C}(y); x)\|$ is a minimum at $C(y) = (A(y); B(y)) = (a_0(y), \dots, a_m(y); b_0(y), \dots, b_k(y))$ where $R(\bar{C}(y); x)$ ranges over all of $\mathbf{R}(m, k)$ and $\|\cdot\|$ is the supremum norm for $C[-1, 1]$, the real-valued continuous functions on $[-1, 1]$.

Under suitable conditions [4], it can be shown that $C(y)$ is continuous on $[-1, 1]$. Thus, the components $a_i(y)$ and $b_j(y)$ possess a best uniform approximation from $\mathbf{R}(m', k')$ which we denote by $P_{a_i}(y)$ and $Q_{b_j}(y)$, i.e.,

$$P_{a_i}(y) = \frac{\sum_{p=0}^{m'} a_{ip} y^p}{\sum_{q=0}^{k'} a_{iq} y^q} \quad \text{and} \quad Q_{b_j}(y) = \frac{\sum_{p=0}^{m'} b_{jp} y^p}{\sum_{q=0}^{k'} b_{jq} y^q}.$$

Definition. For F a continuous real-valued function on D , the best rational product approximation of F is given by

$$T(x, y) = \frac{\sum_{i=0}^m P_{a_i}(y) x^i}{\sum_{j=0}^k Q_{b_j}(y) x^j}.$$

3. Methods of Computing Rational Approximations. Our computation of the rational product approximation using either the differential correction algorithm or the Remez algorithm proceeds as follows:

- (1) Fix m and k and let $N = m + k + 2$.
- (2) Let $W = \{y_1, y_2, \dots, y_M\}$ be a finite subset of $[-1, 1]$ where $-1 = y_1 < \dots < y_M = 1$.
- (3) Starting with y_1 , compute $R(C(y_1); x) \in \mathbf{R}(m, k)$ using one of the above algorithms.
- (4) Continue using the same algorithm to compute $R(C(y_i); x)$ for $i = 2, \dots, M$.

When using the differential correction algorithm, we used $M = 21$ incrementing the y_i 's by .1. For the Remez algorithm we used $M = 201$, incrementing by .01, and making an initial guess at the N alternating points for the approximation of $F_{-1}(x)$.

The above calculations have determined an N by M matrix where the i th column represents $R(C(y_i); x)$. Thus rows 1 through $m + 1$ represent the coefficient functions $a_0(y)$ through $a_m(y)$ and the remaining rows represent $b_0(y)$ through

$b_k(y)$. Figure 1 represents the matrix constructed by the Remez algorithm.

We now choose an m' and k' and use the same algorithm we used above to approximate, from $\mathbf{R}(m', k')$, the functions determined by the rows of this matrix. When these rows are approximated, we have calculated the rational product approximation.

The program we used for the differential correction algorithm was written by E. H. Kaufman, Jr. and G. D. Taylor [5]. This program can also be used to obtain a surface fitting approximation of F on a finite set of grid points contained in D . For further discussion concerning this program, the reader is referred to [5] and [6]. A more complete discussion of the program utilizing the Remez algorithm is presented in [2].

4. The Discontinuity of $C(y)$. We observed in Section 2 that under suitable conditions, $C(y)$ is continuous. When these conditions are not met, M. S. Henry and S. E. Weinstein [4] developed the following technique to overcome a certain type of discontinuity which can arise.

If $C(y)$ is continuous on $[-1, 1]$ except at $y = y^*$, then by approximating $F_y(x)$ from $\mathbf{R}(m, k)$, we have

$$R(C(y); x) = \begin{cases} \frac{P(A_0(y); x)}{Q(B_0(y); x)}, & -1 \leq x \leq 1, -1 \leq y < y^*, \\ \frac{P(A_1(y); x)}{Q(B_1(y); x)}, & -1 \leq x \leq 1, y^* < y \leq 1, \end{cases}$$

where $Q(B_0(y); x)$ and $Q(B_1(y); x)$ exist and are positive on D . Now define

$$P^*(A^*(y); x) = \begin{cases} P(A_0(y); x)Q(B_1(y); x), & -1 \leq x \leq 1, -1 \leq y \leq y^*, \\ P(A_1(y); x)Q(B_0(y); x), & -1 \leq x \leq 1, y^* < y \leq 1, \end{cases}$$

and $Q^*(B^*(y); x) = Q(B_0(y); x)Q(B_1(y); x)$ on D . We then define

$$R^*(C^*(y); x) = P^*(A^*(y); x)/Q^*(B^*(y); x).$$

Now by approximating the components of $C^*(y) = (A^*(y); B^*(y))$, we obtain the modified best rational product approximation of $F(x, y)$.

This method can be applied to a finite number of discontinuities [4], but, for the sake of simplicity, we shall restrict ourselves to the case of one discontinuity.

We observe that $R^*(C^*(y); x)$ is no longer a member of $\mathbf{R}(m, k)$. A discussion of the new class of functions containing $R^*(C^*(y); x)$ is given by M. S. Henry and S. E. Weinstein [4].

We note from our methods of computing the rational product approximation that $Q(B_0(y); x)$ can only be computed for values of y in $[-1, y^*)$ and $Q(B_1(y); x)$ can only be computed for values of y in $(y^*, 1]$ (see Figures 1 and 2). Because of this computational difficulty, a new technique to compute $Q(B_0(y); x)$ and $Q(B_1(y); x)$ was developed in [2].

| | COLUMNS | | | | | | | | | | | |
|-------|---|----|----|----|-----|-----|-----|-----|-----|-----|-----|---|
| | 1 | 50 | 51 | 52 | 100 | 101 | 102 | 150 | 151 | 152 | 201 | |
| 1 | $a_0(-1) \dots a_0(-.51) a_0(-.5) a_0(-.49) \dots a_0(-.01) a_0(0) a_0(.01) \dots a_0(.49) a_0(.5) a_0(.51) \dots a_0(1)$ | | | | | | | | | | | |
| | . | . | . | . | . | . | . | . | . | . | . | . |
| | . | . | . | . | . | . | . | . | . | . | . | . |
| | . | . | . | . | . | . | . | . | . | . | . | . |
| m+1 | $a_m(-1) \dots a_m(-.51) a_m(-.5) a_m(-.49) \dots a_m(-.01) a_m(0) a_m(.01) \dots a_m(.49) a_m(.5) a_m(.51) \dots a_m(1)$ | | | | | | | | | | | |
| m+2 | $b_0(-1) \dots b_0(-.51) b_0(-.5) b_0(-.49) \dots b_0(-.01) b_0(0) b_0(.01) \dots b_0(.49) b_0(.5) b_0(.51) \dots b_0(1)$ | | | | | | | | | | | |
| | . | . | . | . | . | . | . | . | . | . | . | . |
| | . | . | . | . | . | . | . | . | . | . | . | . |
| | . | . | . | . | . | . | . | . | . | . | . | . |
| m+k+2 | $b_k(-1) \dots b_k(-.51) b_k(-.5) b_k(-.49) \dots b_k(-.01) b_k(0) b_k(.01) \dots b_k(.49) b_k(.5) b_k(.51) \dots b_k(1)$ | | | | | | | | | | | |

FIGURE 1

| | | COLUMNS | | | | |
|------|-------|-------------------------|----|----|----|-------------------------|
| | | 1 | 50 | 51 | 52 | 201 |
| ROWS | m+1 | P(A ₀ (y);x) | | | | P(A ₁ (y);x) |
| | m+2 | ----- | | | | |
| | m+k+1 | Q(B ₀ (y);x) | | | | Q(B ₁ (y);x) |
| | m+k+2 | ----- | | | | |

FIGURE 2

We proceed as follows. Suppose y^* is such that $-1 < y^* \leq 0$. Then there exists a smallest positive integer N such that $y^* + N(y^* + 1) \geq 1$. Let $y_i = iy^* + i - 1$ for $i = 0, 1, \dots, N$ and $y_{N+1} = 1$. Define $B_{0,1}(y) = B_0(y)$ for $-1 \leq y \leq y_1$ and $B_{0,i+1}(y) = B_{0,i}(2y_i - y)$ for $y_i < y \leq y_{i+1}$ where $i = 1, 2, \dots, N$. Now define $B'_0(y)$ by

$$B'_0(y) = B_{0,i}(y) \quad \text{where } y_{i-1} \leq y \leq y_i, \quad i = 1, \dots, N+1.$$

An easy argument shows that $B'_0(y)$ is continuous and positive on $[-1, 1]$. We also define $B'_1(y)$ as follows:

$$B'_1(y) = \begin{cases} B_1(2y^* - y), & -1 \leq y < y^*, \\ B_1(y), & y^* \leq y \leq 1. \end{cases}$$

Again, $B_1(y)$ is continuous and positive on $[-1, 1]$.

If $0 < y^* < 1$, then $B'_0(y)$ and $B'_1(y)$ are defined in a similar manner.

Now define

$$P'(A'(y); x) = \begin{cases} P(A_0(y); x)Q(B'_1(y); x), & -1 \leq x \leq 1, -1 \leq y \leq y^*, \\ P(A_1(y); x)Q(B'_0(y); x), & -1 \leq x \leq 1, y^* < y \leq 1, \end{cases}$$

and

$$Q'(B'(y); x) = Q(B'_0(y); x)Q(B'_1(y); x) \quad \text{on } D$$

and

$$R'(C'(y); x) = P'(A'(y); x)/Q'(B'(y); x).$$

Since $B'_0(y) = B_0(y)$ on $[-1, y^*]$ and $B'_1(y) = B_1(y)$ on $[y^*, 1]$, it follows that $R'(C'(y); x) = R^*(C^*(y); x)$; and by virtue of Remark 1 of [4], $R'(C'(y); x) = R(C(y); x)$ on D except possibly at $y = y^*$. It follows that we obtain the modified best rational product approximation

$$T'(x, y) = \frac{\sum_{i=0}^{m+k} P'_{a_i}(y)x^i}{\sum_{j=0}^{2k} Q'_{b_j}(y)x^j},$$

where $P'_{a_i}(y)$ and $Q'_{b_j}(y)$ are the best uniform approximations to $a'_1(y)$ and $b'_j(y)$ from $R(m', k')$.

With this new method, we now have the means to determine the values of $B'_0(y)$ and $B'_1(y)$ for all values of y in $[-1, 1]$ using only the values determined on $[-1, y^*]$ and $[y^*, 1]$.

We conclude this section with an example to illustrate the differences between $C^*(y)$ and $C'(y)$. Let

$$F(x, y) = \begin{cases} \frac{y + 2 + \frac{1}{2}x}{y + 2 + \frac{1}{2}(y + 1)x}, & -1 \leq x \leq 1, -1 \leq y < 0, \\ 1, & -1 \leq x \leq 1, y = 0, \\ \frac{2y + 1 + \frac{1}{3}x}{\frac{1}{2}y + 1 + \frac{1}{3}(y + 1)x}, & -1 \leq x \leq 1, 0 < y \leq 1. \end{cases}$$

Then,

$$C(y) = \begin{cases} \left(y + 2, \frac{1}{2}; y + 2, \frac{1}{2}(y + 1)\right), & -1 \leq y < 0, \\ (1, 0; 1, 0), & y = 0, \\ \left(2y + 1, \frac{1}{3}; \frac{1}{2}y + 1, \frac{1}{3}(y + 1)\right), & 0 < y \leq 1. \end{cases}$$

Apply both techniques and we obtain

$$C^*(y) = \begin{cases} \left(2 + 2y + \frac{1}{2}y^2, \frac{1}{3}y^2 + \frac{5}{4}y + \frac{7}{6}, \frac{1}{6}y + \frac{1}{6}; \right. \\ \left. \frac{1}{2}y^2 + 2y + 2, \frac{7}{12}y^2 + \frac{7}{4}y + \frac{7}{6}, \frac{1}{6}y^2 + \frac{1}{3}y + \frac{1}{6}\right), & -1 \leq y \leq 0, \\ \left(2 + 4y + 2y^2, y^2 + \frac{11}{6}y + \frac{7}{6}, \frac{1}{6}y + \frac{1}{6}; \right. \\ \left. \frac{1}{2}y^2 + 2y + 2, \frac{7}{12}y^2 + \frac{7}{4}y + \frac{7}{6}, \frac{1}{6}y^2 + \frac{1}{3}y + \frac{1}{6}\right), & 0 \leq y \leq 1, \end{cases}$$

and

$$C'(y) = \begin{cases} \left(-\frac{1}{2}y^2 + 2, -\frac{1}{3}y^2 - \frac{7}{12}y + \frac{7}{6}, -\frac{1}{6}y + \frac{1}{6}; \right. \\ \quad \left. -y^2 + 2, -\frac{7}{12}y^2 - \frac{1}{6}y + \frac{7}{6}, \frac{1}{6} - \frac{1}{6}y^2 \right), & -1 \leq y \leq 0, \\ \left(-2y^2 + 3y + 2, -y^2 + \frac{1}{6}y + \frac{7}{6}, -\frac{1}{6}y + \frac{1}{6}; \right. \\ \quad \left. -\frac{1}{2}y^2 + 2, -\frac{7}{12}y^2 + \frac{1}{6}y + \frac{7}{6}, \frac{1}{6} - \frac{1}{6}y^2 \right), & 0 \leq y \leq 1. \end{cases}$$

5. Comparisons. This section is devoted to making comparisons among the three different algorithms. In the first example, we used $m = k = m' = k' = 1$ for the rational product approximations. In the remaining examples we used $m = k = m' = k' = 2$. For the surface fit approximation using the differential correction algorithm, we approximated all of the examples by rational functions of the form

$$\frac{a_0 + a_1x + a_2x^2 + a_3y + a_4y^2 + a_5xy + a_6x^2y + a_7xy^2 + a_8x^2y^2}{b_0 + b_1x + b_2x^2 + b_3y + b_4y^2 + b_5xy + b_6x^2y + b_7xy^2 + b_8x^2y^2}.$$

We used 121 evenly spaced grid points in $[-1, 1] \times [-1, 1]$ for this approximation in each of the examples. All of the examples were executed on the Xerox Sigma 7 located at Montana State University using double precision arithmetic.

Table I gives the amount of time (in minutes) used by the central processor unit (CPU) to compute the rational approximation for the three algorithms. From this table we see that using the Kaufman-Taylor implementation of the differential correction algorithm in conjunction with rational product approximation used the least amount of CPU time. However, this version of the differential correction algorithm produced a discontinuity in the coefficient functions of e^{xy} and $\sin(xy)$ which the Remez algorithm did not. For example, when approximating $\sin(xy)$, the implementation of both the Remez algorithm and the differential correction algorithm produced coefficient functions $a_i(y)$ and $b_j(y)$ which computationally appeared to satisfy

$$\lim_{y \rightarrow 0} a_i(y) = 0 = \lim_{y \rightarrow 0} b_j(y) \quad \text{for } i = 0, 1, 2; j = 1, 2,$$

and

$$\lim_{y \rightarrow 0} b_0(y) = 1.$$

Both algorithms computed $a_0(0) = a_1(0) = a_2(0) = 0$ and $b_0(0) = 1$. However, our implementation of the Remez algorithm calculated $b_1(0) = b_2(0) = 0$ while the Kaufman-Taylor implementation of the differential correction algorithm calculated $b_1(0) = b_2(0) = 1$. Because of the discontinuity produced by the differential correction algorithm in the functions $b_1(y)$ and $b_2(y)$, the error of the approximation was greater than when either the Remez algorithm or the surface fit approximations were

| $F(x,y)$ | REMEZ RPA CPU | DIFF-CORR RPA CPU | SURFACE FIT CPU |
|---|------------------|----------------------|--------------------|
| $\frac{2y+2+.1x}{1+.1x} \quad -1 \leq y \leq -.5$ $\frac{2y+2+.2x}{1+.2x} \quad -.5 \leq y \leq 1$ | .7399 | .3525 | 2.0179 |
| e^{-x^2-y} | 3.7815 | 1.0134 | 2.2306 |
| e^{xy} DIFF-CORR RPA DISCONTINUITY | 2.9869 | 1.1483 | 2.1669 |
| $\sin(xy)$ DIFF-CORR RPA DISCONTINUITY | 2.6931 | 1.1916 | 2.1858 |
| e^{x+y} | 2.8808 | .9331 | 1.9176 |

TABLE I. CPU time used to compute approximations

| $F(x,y)$ | REMEZ RPA ERROR | DIFF-CORR RPA ERROR | SURFACE FIT ERROR |
|---|--------------------|------------------------|----------------------|
| $\frac{2y+2+.1x}{1+.1x} \quad -1 \leq y \leq -.5$ $\frac{2y+2+.2x}{1+.2x} \quad -.5 \leq y \leq 1$ | 2.03664E-2 | 2.71356E-2 | 1.71996E-2 |
| | CPU=1.9445 | CPU=1.7779 | CPU=.5976 |
| e^{-x^2-y} | 4.36716E-3 | 4.43229E-3 | 4.95098E-3 |
| | CPU=3.1884 | CPU=2.4824 | CPU=.6974 |
| e^{xy} DIFF-CORR RPA DISCONTINUITY | 3.80978E-4 | 1.10875E-2 | 1.01874E-4 |
| | CPU=2.8307 | CPU=4.0424 | CPU=.6716 |
| $\sin(xy)$ DIFF-CORR RPA DISCONTINUITY | 1.54656E-3 | 1.43324E-2 | 1.35252E-3 |
| | CPU=2.8302 | CPU=3.9858 | CPU=.5835 |
| e^{x+y} | 4.72317E-4 | 4.60886E-4 | 4.26705E-4 |
| | CPU=2.9871 | CPU=2.4821 | CPU=.6540 |

TABLE II. Error of approximation and CPU time used to determine error

used (see Table II). This increase in the error is a result of introducing a nondifferentiable point in $[-1, 1]$ when the technique of Section 4 is applied.

The error listed in Table II was obtained by finding the maximum difference between the rational approximation and $F(x, y)$ on $Z \times Z$, where $Z = \{-1., -.99, \dots, .99, 1.\}$. The CPU time given in Table II is the amount of time needed to compute the error of the approximation obtained. When the CPU times of Table I and Table II are added, we see that when the error is also calculated, the surface fit approximation uses the least amount of time in all examples except the first. For those functions in which no discontinuities were present in the coefficient functions, the errors for the algorithms are all of the same magnitude.

It is clear that calculation of the maximum error on 40, 401 grid points is very time consuming for rational product approximation. By using an increment of .1 in both the x direction and y direction, we significantly reduce the time needed. For example, in the rational product approximation using the differential correction algorithm, the CPU time used to compute both the approximation *and* the error is reduced to 1.0549 from 3.4152 for e^{x+y} with increments of .1.

From the data we have obtained from our examples, it appears that one should use the rational product approximation with the differential correction algorithm, especially if checking the error at a relatively small number of grid points. The main drawback to this approach is the introduction of discontinuities in the coefficient functions. If a discontinuity does appear, our results suggest that a check on this approximation should be made by utilizing one or both of the other algorithms. The use of the Remez algorithm may result in the disappearance of this discontinuity and an improvement in the accuracy of the approximation. The main disadvantage of the Remez algorithm is that of the initial guess at the alternating points. In a few cases we have had difficulty in finding a good enough guess for the coefficient functions. Clearly, the surface fitting approximation presents no discontinuity problems. In checking the error on a large number of grid points, this is apparently the best algorithm to use.

Based on our experience with these and other examples, none of the algorithms appear to be consistently superior. Each has its advantages and disadvantages. As stated above, we would begin by using the differential correction algorithm in rational product approximation and then use one of the other methods if a discontinuity occurred.

The author gratefully acknowledges Professor Myron S. Henry, coinvestigator with the author under NSF Grant MCS 76-06553, for his assistance during the preparation and revision of this paper.

Department of Mathematics
California State College
Dominguez Hills, California 90747

1. J. A. BROWN & M. S. HENRY, "Best Chebyshev composite approximation," *SIAM J. Numer. Anal.*, v. 12, 1975, pp. 336–344. MR 52 #3828.

2. J. N. HENRY, "Computation of rational product approximations," *Internat. J. Numer. Methods Engrg.*, v. 10, 1976, pp. 1289–1298.

3. M. S. HENRY & J. A. BROWN, "Best rational product approximations of functions," *J. Approximation Theory*, v. 9, 1973, pp. 287–294.
4. M. S. HENRY & S. E. WEINSTEIN, "Best rational product approximations of functions. II," *J. Approximation Theory*, v. 12, 1974, pp. 6–22.
5. E. H. KAUFMAN, JR. & G. D. TAYLOR, "An application of linear programming to rational approximation," *Rocky Mountain J. Math.*, v. 4, 1974, pp. 371–373. MR 49 #4213.
6. E. H. KAUFMAN, JR. & G. D. TAYLOR, "Uniform rational approximation of functions of several variables," *Internat. J. Numer. Methods Engrg.*, v. 9, 1975, pp. 297–323.
7. G. A. WATSON, "A multiple exchange algorithm for multivariate Chebyshev approximation," *SIAM J. Numer. Anal.*, v. 12, 1975, pp. 46–52. MR 51 #9430.
8. S. E. WEINSTEIN, "Approximation of functions of several variables: Product Chebyshev approximations. I," *J. Approximation Theory*, v. 2, 1969, pp. 433–447. MR 40 #7683.