EXPONENTIALLY CONVERGENT LINEAR RATIONAL INTERPOLATION BETWEEN EQUIDISTANT (AND OTHER) POINTS

Jean-Paul Berrut and Hans D. Mittelmann

ABSTRACT. In [2] we used rational functions with a fixed denominator, i.e., independent of the interpolated function, to construct interpolating linear projections with slowly increasing Lebesgue constants for arbitrary interpolation points. Their drawback was the relatively slow convergence of the interpolant as the number of points increases. In the present work, we demonstrate how the convergence can be dramatically improved by limiting the degree of the denominator.

1. Introduction

In the numerical solution of many kinds of equations, one writes the unknown(s) as linear interpolants, introduces the latter into the equations and determines the values at the interpolation points by solving the resulting discretized equation(s). The error of the approximation then is bounded by a multiple of the error of the interpolant to the exact (unknown) solution of the problem.

Many interpolation schemes are linear projections, and their accuracy can be measured by the following classical theorem [6, p. 24]:

Theorem 1.1. Let \( B \) be a finite-dimensional linear subspace of a normed linear space \( A \), and let \( P \) be a linear projection from \( A \) to \( B \). Let \( f \) be fixed in \( A \) and \( d^* \) the distance

\[
d^* := \min_{b \in B} \| f - b \|
\]

of \( f \) to its best approximation in \( B \).

Then the error of the approximation \( Pf \) to \( f \) satisfies the bound

\[
\| f - Pf \| \leq [1 + \| P \|] d^*.
\]

(1.1)

\( \| P \| \) is called the Lebesgue constant of the projection. The theorem implies that any linear projection with relatively small Lebesgue constant onto a linear subspace that approximates \( A \) well will approximate any function in \( A \) almost as well.

Here the spaces \( B \) will be interpolation spaces. Let \( x_0, x_1, \ldots, x_N \) be \( N + 1 \) distinct points (nodes) in some interval \([a, b] \subset \mathbb{R}\), and \( f_k := f(x_k), k = 0, \ldots, N \), the values of the interpolated function \( f \) at the nodes. Since we want to measure...
the interpolation error in the maximum norm, we take $A := C[a, b]$. In view of the linearity, our interpolant can be written as

$$Pf = \sum_{k=0}^{N} f_k \ell_k(x)$$

where $\ell_k(x)$ is the projection of any function $g_k$ with $g_k(x_j) = \delta_{kj}$. In exactly the same way as in the special case of the polynomial, the norm of $P : A \rightarrow B$ then can be seen to be [6, p. 41]

$$\|P\| = \Lambda_N := \max_{a \leq x \leq b} \sum_{k=0}^{N} |\ell_k(x)|. \quad (1.2)$$

The constant $\Lambda_N$ is attained, i.e., there are functions whose norm is amplified by $\Lambda_N$ in the projection. As usual in this context [7, p.88], we will consider triangular arrays of interpolation points where each row contains the points corresponding to a value of $N$.

In the case $B = P_N$, the space of all polynomials of degree $\leq N$, many exact values and precise bounds for $\Lambda_N$ are known, see, e.g., [7] and [9]. If one can choose the interpolation points in such a way that, once transferred to the circle with diameter $[-1, 1]$ by the application $\phi = \arccos \left[\frac{\frac{2}{b-a}x - \frac{b+a}{b-a}}{b-a}\right]$, they are nearly equidistant, then $\Lambda_N$ growing with $N$ as slowly as $\frac{2}{\pi} \ln N + C$ can be attained. Moreover, in general, e.g., for equidistant or randomly chosen points with uniform distribution on $[a, b]$, $\Lambda_N$ grows exponentially [10]. Polynomial interpolation between such nodes therefore is useless for solving equations.

To circumvent this difficulty, we have suggested in [1, 2] to use linear rational interpolation instead of polynomial interpolation, that is, to replace the fixed denominator 1 of the polynomial by a denominator that depends on the interpolation points (but still not on the interpolated function). In this way, the linearity of the interpolation process with respect to the interpolated functions is preserved. It is natural to choose the fixed denominator such that the corresponding Lebesgue constant $\|P\|$ is minimal, as we did in [1, 2]. The extensive numerical results of the latter work lead us to the following "conjecture": there exists a constant $C$ such that, for "almost all" interpolation points arising in practice, the Lebesgue constant of the above optimal linear rational interpolation grows asymptotically as $\Lambda_N \leq C \cdot \ln N$.

The factor containing the Lebesgue constant in the error bound (1.1) therefore does not grow significantly with $N$. On the other hand, the disappointing approximation results of [2] for large $N$ show that the quality of the best approximation with the corresponding fixed denominator $q$ must be poor. At the end of that work, we speculated that "interpolation with a fixed denominator of arbitrary degree (that is, $N$ in most cases) does not leave enough latitude for the numerator to give a good best approximation to $f." \) One possibility to improve on this is to limit the degree of $q$. We show in the following how to obtain much faster convergence of linear rational interpolants by adding linear degree-decreasing side conditions to the Lebesgue constant minimizing optimization problem. A very interesting work in this context is [9] in which it is shown that best rational approximation on $[-1, 1]$ is essentially not better.
than best polynomial approximation for those functions which for $R > 1$ are analytic
and bounded in the ellipse $E_R$ with foci at $+1, -1$ and sum of semi-axes equal to $R$.

### 2. Fast convergence of best approximations by rational approximants with fixed denominators

Let $P_m$ denote the vector space of all polynomials of degree $\leq m$ and let $R_{mn}$ be the
set of all rational functions with numerator degree $m$ and denominator degree $n$.

We will consider a sequence of polynomials $q_n$, one for every natural number $N$
(the number of interpolation points minus 1 in the following sections), with $q_n \in P_n,$
$n = n(N)$. Let $m := m(N) := N - n(N)$, and suppose that $m \to \infty$ as $N \to \infty$.

For every fixed $N$, the set $B_N$ of all $r \in R_{Nn}$ with common denominator $q_n$ is a
vector space of dimension $N + 1$. To every $f \in C[a,b]$, therefore, there exists a best
approximation $r^*_N$ of $f$ in $B_N$ [6, p.6], and we have the following theorem.

**Theorem 2.1.** Let $N, m, n$ be as above and $f \in C[a,b]$. Then, as $N \to \infty$, the
best approximation $r^*_N$ of $f$ in $B_N$ converges toward $f$ at least as fast as the best
approximation to $f$ in $P_m$.

**Proof.** Let $p^*_m$ be the best approximation to $f$ in $P_m$. The conclusion obviously
follows from the fact that $p^*_m = p^*_m q_n/q_n$ is a candidate for the best approximation to
$f$ in $B_N$. \qed

Since the best approximation converges at least as fast as interpolation between
Čebyšev points, this result provides bounds for the convergence of $r^*_N$, toward $f$, e.g.,
if $f \in C^k[a,b]$, then $\|r^*_N - f\|_\infty = O(m^{-k})$, and if $f$ is analytic within an ellipse with
half-axes $\frac{1}{2}(\alpha + \alpha^{-1})$ and $\frac{1}{2}(\alpha - \alpha^{-1})$, $\alpha > 1$, then $\|r^*_N - f\|_\infty = O(\alpha^{-m})$ [5, p. 224].
In the latter case, one thus has quadratic convergence when doubling $m$.

In order to use these encouraging results in conjunction with Theorem 1.1, we must
choose the sequence of polynomials $q_n$ in such a way that the Lebesgue constant of
interpolation in $B_N$ increases with $N$ much slower than $\|r^*_N - f\|$ decreases. The
numerical results in [2] demonstrate that a small Lebesgue constant is almost always
possible with $n = N$, i.e., with no limitation on the degree of the denominator, but
at the cost of an unsatisfactory approximation. At the other extreme, the best
approximation by a polynomial ($n = 0$) is very precise, but the interpolation between
arbitrary points, as measured by its $\Lambda_N$, is disastrous. Our goal is to decrease $n$ from
$N$ as much as possible to improve the precision of the interpolation, while keeping the
Lebesgue constant small. If the latter grows with $N$ much slower than exponentially,
say polynomially, then the interpolation of an analytic $f$ converges exponentially.

### 3. Lebesgue constant minimizing linear rational interpolation

**with limited denominator degree**

In view of its many advantages for the solution of the optimization problem (guaranteed
interpolation, signs of the coefficients given a priori, small modulus of these
coefficients allowing a search for them in a small hypercube), we chose in [1, 2] to use the
barycentric representation of rational interpolants:

$$r^*(u)(x) := \sum_{k=0}^{N} \frac{u_k}{x - x_k} f_k / \sum_{k=0}^{N} \frac{u_k}{x - x_k}.$$
Every rational interpolant in $\mathcal{R}_{NN}$ can be written this way with certain $u_k$ s (Lemma 2.1 in [2]). These weights determine the (unreduced) denominator as

$$q_N(x) = \ell(x) \sum_{k=0}^{N} \frac{u_k}{x-x_k}, \quad \ell(x) := \prod_{i=0}^{N}(x-x_i).$$

The Lebesgue constant of the interpolation with $r \in B_N$ then is [2]

$$\Lambda^{(u)}_N = \sup_{a \leq x \leq b} \left| \prod_{k=0}^{N} (x-x_k) \right| = \sup_{a \leq x \leq b} \left| \sum_{k=0}^{N} \frac{u_k}{x-x_k} \right|. $$

We now will limit the degree of the denominator of $r^{(u)}_N$ and denote the resulting rational in $\mathcal{R}_{NN}$ by $r^{(u)}_N$. As has been shown in [3], $m = N - n$ conditions, which guarantee that the degree of $q_n$ is successively $\leq N - 1, N - 2, \ldots, n$, are given by $\sum_{k=0}^{N} x_k^j u_k = 0, j = 0, \ldots, N - n - 1$. Therefore, the matrix of the linear constraints is made up of the first $N - n$ rows of the transposed Vandermonde corresponding to the $x_k$ s:

$$A := \begin{bmatrix}
1 & 1 & \cdots & 1 \\
x_0 & x_1 & \cdots & x_N \\
x_0^2 & x_1^2 & \cdots & x_N^2 \\
\vdots & \vdots & \ddots & \vdots \\
x_0^{N-n-1} & x_1^{N-n-1} & \cdots & x_N^{N-n-1}
\end{bmatrix}. $$

We therefore solve the following minimization problem with simple bounds and side conditions:

$$\minimize \Lambda^{(u)}_N$$

with respect to the $u_k$, $k \neq i$, subject to

$$m \leq u_k \leq M, \quad k \neq i,$$

and

$$Au = 0, \quad u := [u_0, u_1, \ldots, u_N]^T. $$

(3.1)

$i$ is the index of any one of the weights (which are determined only up to a constant); $u_i$ is kept fixed during the optimization process.

We solved the problem numerically using the same method as in [2]. We will denote the resulting optimal coefficients by $u^*$. As $N$ becomes large, the norm of the pseudoinverse of $A$ becomes large, and the constraints (3.1) ill-conditioned. For that reason, we also have implemented a method using the coefficients in the canonical representation of $r^{(u)}_N$:

$$r^{(u)}_N(x) = \frac{p_N(x)}{\sum_{k=0}^{n} a_k x_k}. $$

(3.2)

One observes from the proof of Theorem 2.1 in [2] that $u_k = w_k q_n(x_k)$ where $w_k = 1/\prod_{j=0, j \neq k}^{N} (x_k - x_j)$. Thus, for given $n$ and $a_k$ s, the $q_n(x_k)$ can be evaluated using
Horner's method, and from these the $u_k$s can be obtained by multiplication with $w_k$. In contrast to the constrained optimization problem of minimizing the Lebesgue constant with respect to the $u_k$s and subject to the constraints (3.1), one now solves an unconstrained problem with the coefficients $a_k$ as variables, automatically ensuring that the denominator polynomial has degree $n$.

To illustrate Theorem 2.1, we also have computed for each optimal denominator $g_n^*$ the best approximation $r_{Nn}^{(u^*)}$ to $f$ in $B_N^*$, the set of all $r \in R_{Nn}$ with common denominator $g_n^*$. Theorem 1.1 implies

**Corollary 3.1.** The error of the approximation to $f$ by the linear rational interpolant $r_{Nn}^{(u^*)}$ satisfies

$$\|f - r_{Nn}^{(u^*)}\| \leq \left[1 + \Lambda_N^{(u^*)}\right] \cdot d_{u^*},$$

where $d_{u^*}$ is given by

$$d_{u^*} = \|f - r_{Nn}^{(u^*)}\| = \min_{p_N \in \mathcal{P}_N} \left\| f - \frac{p_N}{g_n^*} \right\|.$$

We have computed $d_{u^*}$ in the numerical experiments reported in §4. For that purpose, we have again used the equivalent barycentric form

$$d_{u^*} = \min \left\{ y_k \right\} \left\| f - \frac{\sum_{k=0}^{N} u_k}{x - x_k} \cdot y_k \right\|,$$

i.e., we have determined the values $y_k$ of $r_{Nn}^{(u^*)}$ at the $x_k$s.

### 4. Numerical experience

Results such as those presented in the previous sections, while theoretically correct, may not always be achieved to a degree desirable in practical applications. As is typical, also in the problem under investigation, a tradeoff may be expected. Specifically, the first factor containing the Lebesgue constant in the estimate (1.1) may be expected to increase for decreasing denominator degree $n$ while a growing number $N - n$ of degrees of freedom in $r_{Nn}$ would allow for a better approximation to $f$, thereby reducing the size of the second factor. The question is how low the minimum of the product of these factors dips for a given $N$. Due to this tradeoff, a dramatically improved approximation may not be expected if $n$ is chosen lightly. However, when a substantial gain in difficult cases can be achieved compared with polynomial or the Lebesgue constant minimizing linear rational interpolation of [2], the slight additional effort of having the additional linear constraints (3.1) will have been worthwhile.

In the following, we denote by $\beta$ the ratio $n/N$ with $n$ and $N$ as in Theorem 2.1. If $\beta \equiv \text{const.}$ for $N \to \infty$ then $m = (1 - \beta)N \to \infty$ as required there. We first choose equidistant nodes $x_i = -1 + ih$, $i = 0, \ldots, N, h = 2/N$ in $[-1, 1]$, and we also restrict consideration to even functions and polynomials. Although only powers of 2 appear in the tables, there are no restrictions on the value of $N$.

Table 4.1 shows the growth of the Lebesgue constant for decreasing $\beta$ and different values of $N$. From this it is clear that values of $\beta$ not too much smaller than 1 should be considered for larger $N$. A lone star as entry means that the corresponding value
could not be computed. The entry 7.9 with the star means that this entry could not be found, but that 7.9 is the result for \( \beta = 53/64 \) instead of \( \beta = 52/64 \).

For test functions, we have considered the Runge example \((1 + bx^2)^{-1}\), for which the polynomial interpolating between equidistant points diverges at the extremities of \([-1,1]\) if \( b > (1/\xi^2) \) with \( \xi \approx .5255 \), i.e., if \( b > 3.622 \) [4]. Table 4.2 displays the maximal errors with equidistant as well as randomly chosen points for \( b = 5 \) and \( b = 25 \).

Table 4.3 lists for \( \beta = 3/4 \) the values of \(|f - r_n^{(u^*})|\) as well as of the error of the best approximation \( r_n^{*} \) in \( \mathcal{B}^*_{N} \) for two different functions \( f \) and various \( N \). The reader may check that the product of the latter error with the corresponding Lebesgue constant in Table 4.1 is always larger than the interpolation error.

In view of Theorem 2.1, the approximation result with \( r_n^{*} \) for a given number \( N + 1 \) of nodes should be at least as good as the best polynomial approximation of degree \( m = (1 - \beta)N \). To allow for the comparison, we give in Table 4.4 the approximation errors of the best polynomial approximations by polynomials of various degrees \( m \).

In all cases but the last, our Lebesgue-constant minimizing rational interpolation yields a smaller approximation error than a polynomial with the same number of degrees of freedom. Several hints led us to believe that numerical difficulties in the
solution of the optimization problem are responsible for the slowing down of the convergence as \(N\) becomes large. In particular, equioscillation of the Lebesgue function (see [2]), still present here for small \(N\), is less and less satisfied as \(N\) grows. Some of the results were verified by reproducing them with quadruple precision. Although this was prohibitively slow for the largest cases, \(N = 128, 256\) in particular, the agreement seemed to diminish as \(N\) was increased.

Table 4.5 shows the results analogous to those in Table 4.3, but for \(\beta = 7/8\).

To show that the method is not restricted to equidistant points, but could be used with most matrices of interpolation points, we display in Tables 4.6 and 4.7 the results analogous to those in Tables 4.3 and 4.5 for random interpolation points. The latter were generated on the interval \((0,1)\) by the routine \texttt{drand48} of the \texttt{f77}-compiler on the HP9000, initialized by the command \texttt{srand48(\(\cdot\))}, ordered and reflected with respect to the origin.

5. Solution of the computational problems

Since all the numerical problems which needed to be solved to obtain the results of the previous section are standard, we wanted to use as much as possible standard methods and approaches that are implemented in easily accessible, preferably public domain computer programs. This could be accomplished, and we describe them in
the following in such a way that our results may be reproduced. It should be noted, however, that the computations were delicate and time-consuming. This does not diminish the usefulness of our approach in applications because the optimal weights

<table>
<thead>
<tr>
<th>(N)</th>
<th>(|f - r_{Nn}^{(u^*)}|)</th>
<th>(|f - r_{Nn}^*|)</th>
<th>(|f - r_{Nn}^{(u^*)}|)</th>
<th>(|f - r_{Nn}^*|)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>2.4e-3</td>
<td>1.8e-3</td>
<td>4.3e-3</td>
<td>2.7e-3</td>
</tr>
<tr>
<td>32</td>
<td>2.1e-4</td>
<td>1.5e-4</td>
<td>3.7e-4</td>
<td>2.3e-4</td>
</tr>
<tr>
<td>64</td>
<td>8.6e-5</td>
<td>4.1e-5</td>
<td>2.5e-4</td>
<td>1.5e-4</td>
</tr>
<tr>
<td>128</td>
<td>3.7e-6</td>
<td>2.2e-6</td>
<td>8.0e-5</td>
<td>5.3e-5</td>
</tr>
<tr>
<td>256</td>
<td>8.1e-9</td>
<td>*</td>
<td>1.2e-5</td>
<td>8.5e-6</td>
</tr>
</tbody>
</table>

**Table 4.5.** Error of Lebesgue-constant minimizing rational interpolation and of best approximation in the same subspace for \(\beta = 7/8\).

<table>
<thead>
<tr>
<th>(N)</th>
<th>(|f - r_{Nn}^{(u^*)}|)</th>
<th>(|f - r_{Nn}^*|)</th>
<th>(|f - r_{Nn}^{(u^*)}|)</th>
<th>(|f - r_{Nn}^*|)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1.8e-1</td>
<td>8.4e-2</td>
<td>3.2e-1</td>
<td>1.3e-1</td>
</tr>
<tr>
<td>16</td>
<td>9.5e-2</td>
<td>3.6e-2</td>
<td>2.3e-1</td>
<td>9.6e-2</td>
</tr>
<tr>
<td>32</td>
<td>4.5e-2</td>
<td>8.8e-3</td>
<td>2.0e-1</td>
<td>6.0e-2</td>
</tr>
<tr>
<td>64</td>
<td>4.0e-4</td>
<td>1.3e-4</td>
<td>3.4e-2</td>
<td>9.7e-3</td>
</tr>
<tr>
<td>128</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

**Table 4.6.** Same as Table 4.3 (\(\beta = 3/4\)) but for random interpolation points.

<table>
<thead>
<tr>
<th>(N)</th>
<th>(|f - r_{Nn}^{(u^*)}|)</th>
<th>(|f - r_{Nn}^*|)</th>
<th>(|f - r_{Nn}^{(u^*)}|)</th>
<th>(|f - r_{Nn}^*|)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>1.3e-1</td>
<td>5.7e-2</td>
<td>2.6e-1</td>
<td>1.1e-1</td>
</tr>
<tr>
<td>32</td>
<td>5.9e-4</td>
<td>2.7e-4</td>
<td>9.3e-4</td>
<td>5.1e-4</td>
</tr>
<tr>
<td>64</td>
<td>1.3e-2</td>
<td>4.9e-3</td>
<td>5.1e-2</td>
<td>2.5e-2</td>
</tr>
<tr>
<td>128</td>
<td>4.5e-5</td>
<td>2.3e-5</td>
<td>4.9e-3</td>
<td>2.0e-3</td>
</tr>
<tr>
<td>256</td>
<td>6.4e-7</td>
<td>*</td>
<td>1.4e-3</td>
<td>4.0e-4</td>
</tr>
</tbody>
</table>

**Table 4.7.** Same as Table 4.4 (\(\beta = 7/8\)) but for random interpolation points.
need to be computed only once, what we already did for two types of nodes, and can then be stored and used for the approximation of arbitrary functions.

The core problem is the linearly as well as bound-constrained minimization problem (3.1). For this, the sequential quadratic programming method was used as implemented in FFSQP/CFSQP [11]. We refer to [2] where details are given for the case without linear constraints. It suffices to say that a discontinuity inherent in this minimax problem was overcome by using the capability of FFSQP to minimize multiple objective functions. Here, the objective functions were chosen as the local maxima of the quotient expression in the definition of the Lebesgue constant between the interpolation points. Again, it was crucial to use continuation with respect to the nodes from a starting distribution with known feasible coefficients, here, as in [2], Čebyšev nodes, to the desired points.

The errors in Table 4.2 were computed using the barycentric formula of Section 3 and evaluation at the points used in Section 5 of [2]. The entries in the second and fourth columns of Tables 4.3, 4.5-4.7 were obtained by minimizing \( \| f - r \| \) over all \( r \in R_N \) with the fixed denominator. Precisely, the barycentric formula was used with the optimal weights in both numerator and denominator while the function values in the numerator were treated as free parameters, cf. the end of Section 3. The error norm as measured over the set of points from Section 5 of [2] then was minimized by varying these function values. The values in Table 4.4 were computed with continuous Čebyšev approximation programs for smaller \( m \). However, the program, DRATCH from the IMSL library, failed for larger \( m \), and a discrete Čebyšev approximation program, here algorithm 501 from the TOMS collection in NETLIB, had to be used. A very large number of discrete nodes, \( 10^4-10^5 \), was chosen in order to minimize the deviation from continuous approximation.

6. Conclusion

The numerical results given above show that, for any array of nodes arising in practice, best approximation decreasing for small \( N \) as fast as polynomial approximation can be achieved in practice. The small Lebesgue constants for these values of \( N \) yield a very good linear rational interpolant for functions like \( 1/(1 + bx^2) \) (see Tables 4.3 and 4.5), for which the interpolating polynomial is useless (Table 4.2).

On the other hand, exponential convergence for difficult functions like \( 1/(1 + 25x^2) \) does not seem to be easily attainable. At present we are unclear about the exact reason: is it merely the stability of the algorithm, or is the whole problem ill-conditioned?

The validity of Theorem 2.1 relies heavily on the fact that the sequence of denominators has precise degree \( n \), in our examples \( n = \beta N \). We have achieved the corresponding degree reduction by the transposed Vandermonde linear constraints (3.1) in one method, by computing the \( u_k \)s from the \( a_k \)s in (3.2) in another. Both methods probably become ill-conditioned as \( N \) increases. Our calculations indeed seem to show that the number of identical digits in \( u^* \) resulting from different methods seems to decrease as \( N \) becomes larger. It therefore is possible that, if better degree reduction can be achieved, e.g., by working in higher precision, then \( u^* \)s with much better asymptotic approximation power can be computed. Our hope is that the main problem is not too ill-conditioned, i.e., that arrays of FORTRAN double precision \( u^* \)s with small Lebesgue constant and precisely satisfied degree reduction conditions exist.
Acknowledgement. This work has been supported by the Swiss National Science Foundation, Grant No. 21-42097.94.

References


