



A Newton method for best uniform rational approximation

Irina Georgieva¹ · Clemens Hofreither² 

Received: 3 February 2022 / Accepted: 7 December 2022 / Published online: 27 December 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

We present a novel algorithm, inspired by the recent BRASIL algorithm, for best uniform rational approximation of real continuous functions on real intervals based on a formulation of the problem as a nonlinear system of equations and barycentric interpolation. We derive a closed form for the Jacobian of the system of equations and formulate a Newton's method for its solution. The resulting method for best uniform rational approximation can handle singularities and arbitrary degrees for numerator and denominator. We give some numerical experiments which indicate that it typically converges globally and exhibits superlinear convergence in a neighborhood of the solution. A software implementation of the algorithm is provided. Interesting auxiliary results include formulae for the derivatives of barycentric rational interpolants with respect to the interpolation nodes, and for the derivative of the nullspace of a full-rank matrix.

Keywords Rational approximation · Best uniform rational approximation · Newton's method · Barycentric interpolation

1 Introduction

The best uniform rational approximation of real functions is a classical problem with applications throughout mathematics and the sciences. There has been a recent

✉ Clemens Hofreither
clemens.hofreither@ricam.oeaw.ac.at

Irina Georgieva
irina@math.bas.bg

¹ Bulgarian Academy of Sciences, Institute of Mathematics and Informatics, Acad. G. Bonchev, Bl. 8, 1113, Sofia, Bulgaria

² Johann Radon Institute for Computational and Applied Mathematics (RICAM), Altenberger Str. 69, 4040, Linz, Austria

resurgence of interest in the fast computation of best uniform rational approximations, motivated by applications in solving fractional diffusion problems [17, 19, 22] and more generally in the context of rational Krylov methods for computing matrix functions [8, 9, 16]. The classical algorithm for best uniform rational approximation is the Remez algorithm (see, e.g., [6, 7, 29]), which is based on the idea of iteratively determining the nodes in which the approximation error equioscillates. The severe numerical instabilities which occur in this approach are usually dealt with using extended precision arithmetic (as in [29]). Novel approaches for stabilizing the Remez algorithm based on the so-called barycentric rational formula were recently proposed in [13, 24].

Recently, a new algorithm for computing best rational approximations was proposed based on a different idea [23]: observing that the best approximation must interpolate the target function in a number of nodes, we can take these interpolation nodes as our unknowns to be found, rather than the equioscillation nodes as is done in the Remez algorithm. The iterative BRASIL algorithm [23] achieves this by a simple heuristic, namely, by rescaling the lengths of the intervals between the interpolation nodes so as to equilibrate the local errors. Rational interpolation is performed using the barycentric formula. This novel approach appears to enjoy excellent numerical stability. However, as a fixed-point iteration, its convergence rate is only linear, whereas the Remez algorithm converges quadratically in a neighborhood of the exact solution. Therefore our goal for the present work is to construct an algorithm which seeks the interpolation nodes while converging superlinearly. Furthermore, whereas the BRASIL algorithm usually only works well for computing rational approximations with equal degree for the numerator and denominator, we seek a method which can compute approximations with arbitrary degrees.

To this end, we take as a starting point our recent work on a Newton's method for best polynomial approximation [14] and extend it to the rational case. Just as in this prior work, we reformulate the problem of best uniform approximation as a system of nonlinear equations using the equioscillation property with the interpolation nodes as the unknowns. The computation of the Jacobian matrix of this system is significantly more involved in the rational case. It involves computing derivatives of barycentric interpolants with respect to the interpolation nodes, which in turn requires, in a certain sense, to compute the derivative of the nullspace of a matrix. We will make use of some concepts from differential geometry, interpreting these spaces as points on the Grassmann manifold, in order to derive formulae for such derivatives which are of more general interest in their own right.

A widely used code for best uniform rational approximation is the `minimax` routine of the Matlab package `Chebfun` [11] which is based on a barycentric formulation of the Remez algorithm with adaptively chosen support points [13]. However, it was previously observed [23] that this routine does not seem to work well for functions with singularities on or close to the boundary of the approximation interval, typically breaking down already at low rational degrees. In fact, these issues served as the initial motivation for the development of the BRASIL algorithm.

The remainder of the paper is structured as follows: in Section 2, we state the problem of best uniform rational approximation, reformulate it as a system of nonlinear equations and propose a Newton's method to solve it. In Section 3, we employ the

barycentric rational formula to derive an interpolation scheme with arbitrary degrees of the numerator and denominator. In Section 4, we apply tools from differential geometry to derive formulae for the derivative of the orthogonal complement of first orthogonal and then general matrices. We apply these results to our interpolation formula in order to obtain derivatives of barycentric interpolants in Section 5, which yields an identity for the Jacobian of our nonlinear system. In Section 6, we combine all previous results and state the algorithm for best rational approximation in its entirety, and finally present some numerical examples and concluding remarks in Section 7.

2 Best uniform rational approximation as a system of nonlinear equations

We seek to determine a rational function $r \in \mathcal{R}_{m,n}$ with degree of the numerator at most m and of the denominator at most n which best approximates a given continuous function $f \in C[a, b]$ in a real interval $[a, b]$ in the maximum norm. It is a classical result that such a best approximation exists and is unique [2, 28]. Let $m^* \leq m$ and $n^* \leq n$ denote the actual degrees of the best approximation r ; then $d = \min\{m - m^*, n - n^*\}$ is the so-called *defect* of r . The following equioscillation result characterizes the best rational approximation.

Theorem 1 [2, 28] *A rational function $r \in \mathcal{R}_{m,n}$ is the best uniform rational approximation to $f \in C[a, b]$ if and only if the error $f - r$ equioscillates between at least $m + n + 2 - d$ extreme points, where d is the defect of the best approximation.*

We will in the following always assume that $d = 0$, but problems with $d > 0$ can be treated by reducing the degrees m and n correspondingly. Therefore, we assume that there exist $m + n + 2$ distinct local maxima $(y_j)_{j=0}^{m+n+1}$ in $[a, b]$ of the error function such that

$$f(y_j) - r(y_j) = \lambda(-1)^j, \quad j = 0, \dots, m + n + 1,$$

where $\lambda = \pm\|f - r\|_\infty$. Due to continuity, there must exist $m + n + 1$ distinct interpolation nodes $(x_i)_{i=0}^{m+n}$ in (a, b) with

$$r(x_i) = f(x_i), \quad i = 0, \dots, m + n,$$

interleaving the equioscillation nodes in the sense

$$a \leq y_0 < x_0 < y_1 < \dots < y_{m+n} < x_{m+n} < y_{m+n+1} \leq b.$$

Let $\mathbf{x} \in \mathcal{X}$ denote a vector of interpolation nodes in the admissible set

$$\mathcal{X} := \{\mathbf{x} \in (a, b)^{m+n+1} : x_0 < \dots < x_{m+n}\}$$

of nodes in increasing order. We denote by $y \mapsto \rho(\mathbf{x}, y) \in \mathcal{R}_{m,n}$ the rational function which interpolates f in the nodes \mathbf{x} , that is, $\rho(\mathbf{x}, x_j) = f(x_j)$ for $j = 0, \dots, m + n$. It should be noted that, unlike in the polynomial case, such an interpolating rational function may not exist due to so-called *unattainable points* [27], but this is the

exceptional case and the arguments above show that the interpolant exists at least for those interpolation nodes associated with the best approximant. We will implicitly assume the existence of the interpolant in the following. In each interval (x_{j-1}, x_j) , $j = 0, \dots, m + n + 1$ (letting $x_{-1} := a$ and $x_{m+n+1} := b$), let

$$y_j := y_j(\mathbf{x}) := \arg \max_{y \in (x_{j-1}, x_j)} |f(y) - \rho(\mathbf{x}, y)|, \quad j = 0, \dots, m + n + 1, \quad (1)$$

denote the abscissa where the interpolation error $|f - \rho|$ is largest. Let

$$\Phi(\mathbf{x}) := (f(y_j) - \rho(\mathbf{x}, y_j))_{j=0}^{m+n+1}, \quad \mathbf{w} := ((-1)^j)_{j=0}^{m+n+1}.$$

Theorem 2 *If there exists $\lambda \in \mathbb{R}$ such that*

$$F(\mathbf{x}, \lambda) := \Phi(\mathbf{x}) - \lambda \mathbf{w} = 0, \quad F : \mathbb{R}^{m+n+2} \rightarrow \mathbb{R}^{m+n+2}, \quad (2)$$

then $\rho(\mathbf{x}, \cdot)$ is the best uniform rational approximation to f with error $|\lambda| = \|f - \rho(\mathbf{x}, \cdot)\|_\infty$.

Proof By definition, $\|f - \rho(\mathbf{x}, \cdot)\|_\infty = \max_{j=0}^{m+n+1} |f(y_j) - \rho(\mathbf{x}, y_j)| = |\lambda|$, and thus the error $f - \rho(\mathbf{x}, \cdot)$ equioscillates in $(y_j)_{j=0}^{m+n+1}$. The statement follows with Theorem 1. □

Thus, if we find a solution (\mathbf{x}, λ) of the nonlinear equation (2), we have also found the best uniform rational approximation $\rho(\mathbf{x}, \cdot)$ to f , and λ is the signed best approximation error.

We propose to solve (2) using Newton’s method. Given initial guesses for the nodes $\mathbf{x}^0 \in \mathcal{X}$ and the signed error $\lambda^0 \in \mathbb{R}$, a Newton step for the solution of (2) is given by

$$\mathbf{d}^0 := (\mathbf{d}_x^0, d_\lambda^0) := -(\nabla F(\mathbf{x}^0, \lambda^0))^{-1} F(\mathbf{x}^0, \lambda^0) \in \mathbb{R}^{m+n+2}.$$

We must keep the interpolation nodes in the admissible set \mathcal{X} , that is, within the interval (a, b) and in increasing order. To this end we take a damped step $\tau \mathbf{d}^0$ with $\tau \in (0, 1]$ chosen according to

$$\tau = \max\{2^{-t} : t \in \mathbb{N}_0, \mathbf{x}^0 + 2^{-t} \mathbf{d}_x^0 \in \mathcal{X}\}. \quad (3)$$

Such a choice always exists since $\mathbf{x}^0 \in \mathcal{X}$ and \mathcal{X} is an open set. We then take the Newton step

$$\mathbf{x}^1 := \mathbf{x}^0 + \tau \mathbf{d}_x^0, \quad \lambda^1 := \lambda^0 + \tau d_\lambda^0$$

and repeat the procedure until convergence. The computation of the Jacobian matrix of F will be discussed in the following sections.

3 Barycentric rational interpolation with arbitrary degrees

We will use the so-called barycentric rational formula to solve the rational interpolation problems that our method relies on; see [4, 5, 25] and the references therein for details on this formula. One of the advantages of the barycentric formula is its superior numerical stability [4, 21, 27]. A recent and popular algorithm for rational

approximation using this representation is the AAA algorithm [26], which however does not compute minimax approximations.

For $N \in \mathbb{N}_0$ and vectors of real nodes, values, and weights, respectively,

$$\mathbf{z} = (z_i)_{i=0}^N, \quad \mathbf{f} = (f_i)_{i=0}^N, \quad \mathbf{w} = (w_i)_{i=0}^N,$$

where the nodes (z_i) are pairwise distinct, the barycentric formula of degree N is given by

$$r(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) := \frac{\sum_{i=0}^N \frac{f_i w_i}{x - z_i}}{\sum_{i=0}^N \frac{w_i}{x - z_i}}. \tag{4}$$

It can represent arbitrary rational functions in $\mathcal{R}_{N,N}$. For any node z_i with $w_i \neq 0$, we have the interpolation condition $r(\mathbf{z}, \mathbf{f}, \mathbf{w}, z_i) = f_i$, and thus we immediately get interpolation in the $N + 1$ nodes (z_i) if we choose the weights nonzero. The barycentric formula is invariant with respect to rescaling of the weights by a nonzero factor.

To achieve interpolation with fixed degrees (m, n) in $m + n + 1$ nodes, we set $N := \max\{m, n\}$ and $\hat{N} := \min\{m, n\} = m + n - N$. We separate the $m + n + 1$ given nodes \mathbf{x} into $N + 1$ primary nodes $\mathbf{z} = (z_i)_{i=0}^N$, which serve as the nodes in the barycentric formula (4), and \hat{N} secondary nodes $\hat{\mathbf{z}} = (\hat{z}_i)_{i=1}^{\hat{N}}$. The vector of values is just $\mathbf{f} = (f(z_i))_{i=0}^N$. In order to enforce interpolation at the secondary nodes $\hat{\mathbf{z}}$, the weight vector \mathbf{w} is chosen to lie in the nullspace of the Löwner matrix $L \in \mathbb{R}^{\hat{N} \times (N+1)}$ with entries

$$L_{k\ell} = \frac{f(\hat{z}_k) - f(z_\ell)}{\hat{z}_k - z_\ell}, \quad k = 1, \dots, \hat{N}, \ell = 0, \dots, N. \tag{5}$$

If $m = n$, the Löwner matrix L has shape $N \times (N + 1)$; assuming it has full rank, then there are no unattainable points, the weight vector \mathbf{w} is unique up to scaling and the rational interpolant is uniquely determined [23–25].

If $m \neq n$, we need $N - \hat{N}$ additional conditions on \mathbf{w} in order to enforce the correct degrees of the numerator and the denominator. Following [3], we introduce the matrices $A(\mathbf{z}, \mathbf{f}, \mu) \in \mathbb{R}^{\mu \times (N+1)}$ with entries

$$[A(\mathbf{z}, \mathbf{f}, \mu)]_{k\ell} = f_\ell z_\ell^k, \quad k = 0, \dots, \mu - 1, \ell = 0, \dots, N. \tag{6}$$

It is shown in [3, Theorem 3.1] that the numerator of r has degree at most m iff $\mathbf{w} \in \ker A(\mathbf{z}, \mathbf{f}, N - m)$, and the denominator of r has degree at most n iff $\mathbf{w} \in \ker A(\mathbf{z}, \mathbf{1}, N - n)$. Here $\mathbf{1} \in \mathbb{R}^{N+1}$ is the constant vector of ones. Thus, by choosing \mathbf{w} from the nullspace of

$$B := \begin{bmatrix} L \\ A(\mathbf{z}, \mathbf{1}, N - n) \\ A(\mathbf{z}, \mathbf{f}, N - m) \end{bmatrix} \in \mathbb{R}^{N \times (N+1)}, \tag{7}$$

we obtain our interpolant $r(\mathbf{z}, \mathbf{f}, \mathbf{w}, x)$ of degree (m, n) through $m + n + 1$ given nodes. The upper block L guarantees the interpolation condition, and the two lower blocks (of which at most one has a nonzero number of rows) enforce the degree constraint.

4 Derivative of the nullspace of a matrix

For computing derivatives of rational interpolants, we will require, given a matrix and the direction of its derivative, respectively, $B, \dot{B} \in \mathbb{R}^{N \times (N+1)}$, a formula for the corresponding derivative \dot{w} of the vector w spanning the nullspace. Equivalently, we can consider the transpose B^T and compute the derivative of the orthogonal complement of its range since $\ker B = (\text{span } B^T)^\perp$. To this end, we will consider the space spanned by B^T as a point of the Grassmannian manifold $G_{N+1,N}$ which consists of all N -dimensional subspaces of \mathbb{R}^{N+1} . Useful concepts for computing with both the Grassmann and the Stiefel manifolds which we will make heavy use of are given in [1, 12].

4.1 Derivative of the orthogonal complement in the Stiefel manifold

We first consider the case of orthonormal matrices. Let $k \leq n$ be integers and denote by

$$V_{n,k} = \{Y \in \mathbb{R}^{n \times k} : Y^T Y = I_k\}, \quad G_{n,k} = \{\text{span } Y : Y \in V_{n,k}\}$$

the (compact) Stiefel manifold of matrices with orthonormal columns and the Grassmann manifold of k -dimensional subspaces of \mathbb{R}^n , respectively. Let a matrix $Y \in V_{n,k}$ be given and choose $Z \in V_{n,n-k}$ such that their concatenation

$$\begin{bmatrix} Y & Z \end{bmatrix} \in \mathbb{R}^{n \times n} \text{ is orthogonal.} \tag{8}$$

In other words, Z forms an orthonormal basis for the orthogonal complement $(\text{span } Y)^\perp$ of the range of Y .

The tangent spaces $T_Y V_{n,k}$ at a point $Y \in V_{n,k}$ of the Stiefel manifold and $T_{\text{span } Y} G_{n,k}$ at a point $\text{span}(Y) \in G_{n,k}$ of the Grassmann manifold may be parameterized as (cf. [12])

$$\begin{aligned} T_Y V_{n,k} &= \{Y \tilde{A} + Z \tilde{B} : \tilde{A} \in \mathbb{R}^{k \times k} \text{ skew-symmetric, } \tilde{B} \in \mathbb{R}^{(n-k) \times k}\}, \tag{9} \\ T_{\text{span } Y} G_{n,k} &= \{Z \tilde{B} : \tilde{B} \in \mathbb{R}^{(n-k) \times k}\}. \tag{10} \end{aligned}$$

We now wish to derive a formula for the derivative of Z as Y varies. From (8) it follows that $Y^T Y = I_k$, $Z^T Z = I_{n-k}$, $Y^T Z = 0$, and $Y Y^T + Z Z^T = I_n$. By differentiation we obtain

$$\dot{Y}^T Y + Y^T \dot{Y} = 0, \quad \dot{Z}^T Z + Z^T \dot{Z} = 0, \quad \dot{Y} Y^T + Y \dot{Y}^T + \dot{Z} Z^T + Z \dot{Z}^T = 0. \tag{11}$$

The first two identities imply that the matrices $Y^T \dot{Y}$ and $Z^T \dot{Z}$ are skew-symmetric. In fact (cf. [12]), this is equivalent to the fact that \dot{Y} lies in the tangent space $T_Y V_{n,k}$ at Y , and analogously $\dot{Z} \in T_Z V_{n,n-k}$. It follows then from (9) that the matrix \dot{Z} has a representation

$$\dot{Z} = ZA + YB, \quad A \in \mathbb{R}^{(n-k) \times (n-k)} \text{ skew-symmetric, } B \in \mathbb{R}^{k \times (n-k)}.$$

Inserting this ansatz into the third identity in (11) and using the skew-symmetry of A , we obtain

$$YBZ^T + ZB^T Y^T = -\dot{Y} Y^T - Y \dot{Y}^T.$$

Multiplying from the left with Y^T and from the right with Z and using orthogonality yields

$$B = -\dot{Y}^T Z.$$

Note that A can be chosen arbitrarily save for being skew-symmetric; therefore, we simply set $A = 0$. In fact, this is the canonical choice here since, due to (10), it means that

$$\dot{Z} = -Y\dot{Y}^T Z \tag{12}$$

lies in the tangent space $T_{\text{span } Z}G_{n,n-k}$ to the space spanned by Z considered as a point of the Grassmannian $G_{n,n-k}$. Since we are only interested in the spanned space and not the concrete matrix representation, this is the correct setting. Likewise, it is easy to see that \dot{Z} depends only on the second term of $\dot{Y} = Y\dot{A} + Z\dot{B}$, namely, the contribution $Z\dot{B} \in T_{\text{span } Y}G_{n,k}$ that induces a change in the space spanned by Y . Therefore, we can consider the mapping $\dot{Y} \mapsto -Y\dot{Y}^T Z$ as a homomorphism (indeed, an isomorphism) between the tangent spaces $T_{\text{span } Y}G_{n,k} \rightarrow T_{\text{span } Z}G_{n,n-k}$ which both have dimension $k(n - k)$. In terms of differential geometry, it represents the derivative of the isomorphism $\text{span } Y \mapsto (\text{span } Y)^\perp$, mapping $G_{n,k} \rightarrow G_{n,n-k}$, which associates to a space its orthogonal complement.

In brief, we have shown the following theorem.

Theorem 3 *Let a k -dimensional subspace $\text{span } Y$ of \mathbb{R}^n and its $(n - k)$ -dimensional orthogonal complement $\text{span } Z$ be given, where $Y \in V_{n,k}$, $Z \in V_{n,n-k}$ are matrices with orthonormal columns. Then, given a tangent direction $\dot{Y} \in T_{\text{span } Y}G_{n,k}$ for the former space, the corresponding derivative of the orthogonal complement has the tangent direction $\dot{Z} = -Y\dot{Y}^T Z \in T_{\text{span } Z}G_{n,n-k}$.*

4.2 Derivative of the orthogonal complement for a general matrix

The results of Section 4.1 require the matrix Y under consideration to be orthonormal. We now extend the result for the derivative of the orthogonal complement to arbitrary matrices of full rank using two different approaches, the singular value decomposition and the QR decomposition.

4.2.1 Using the singular value decomposition

Let $A \in \mathbb{R}^{n \times k}$ have full rank k and denote by

$$A = Y\Sigma V^T, \quad Y \in V_{n,k}, \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_k), \quad V \in V_{k,k}$$

its (thin) singular value decomposition (SVD). Choose a corresponding $Z \in V_{n,n-k}$ such that $YY^T + ZZ^T = I_n$ as in Section 4.1. One way to do this is by computing the full, rather than thin, SVD of A and using the $n - k$ last left-singular vectors as Z .

Clearly, $\text{span } A = \text{span } Y$, and we are interested in computing the derivative of the orthogonal complement of that space given a perturbation $\dot{A} \in \mathbb{R}^{n \times k}$ of A . We will achieve this by computing \dot{Z} via identity (12), for which we need an expression for $Z^T \dot{Y}$ in terms of \dot{A} . We use here some ideas for computing the derivative of the SVD (see, e.g., [15]), but our task is simplified by the fact that we require only the

tangent direction to the left-singular space. Since $Y \in V_{n,k}$, we have, arguing as in Section 4.1, that $Y^T \dot{Y}$ is skew-symmetric and therefore

$$\dot{Y} = Y\tilde{A} + Z\tilde{B}, \quad \tilde{A} \in \mathbb{R}^{k \times k} \text{ skew-symmetric, } \tilde{B} \in \mathbb{R}^{(n-k) \times k}.$$

We are only interested in the tangent vector to the space spanned by Y , that is, the component $Z\tilde{B}$. Taking the derivative of the SVD identity $A = Y\Sigma V^T$, we obtain

$$\dot{A} = \dot{Y}\Sigma V^T + Y\dot{\Sigma} V^T + Y\Sigma \dot{V}^T.$$

Inserting the ansatz for \dot{Y} and multiplying from the left with Z^T and from the right with V , we obtain

$$Z^T \dot{A} V = \tilde{B} \Sigma$$

from which we can compute \tilde{B} . We conclude

$$Z^T \dot{Y} = \tilde{B} = Z^T \dot{A} V \Sigma^{-1},$$

which we can insert in (12) to obtain the compact expression

$$\dot{Z} = -Y \Sigma^{-1} V^T \dot{A}^T Z \tag{13}$$

for the derivative of Z given a perturbation of A in the tangent direction \dot{A} . More precisely, $\dot{Z} \in T_{\text{span } Z} G_{n,n-k}$ represents the tangent direction along which the space $\text{span}(Z) = \text{span}(A)^\perp \in G_{n,n-k}$ is perturbed.

4.2.2 Using the QR decomposition

Let $A \in \mathbb{R}^{n \times k}$ have full rank k and denote by

$$A = QR, \quad Q \in V_{n,k}, \quad R \in \mathbb{R}^{k \times k} \text{ upper triangular}$$

its (reduced) QR decomposition. Choose a corresponding $Z \in V_{n,n-k}$ such that $QQ^T + ZZ^T = I_n$ as in Section 4.1. Such a Z may be obtained by computing the full QR decomposition and splitting the obtained orthogonal matrix into the components $[Q \ Z] \in \mathbb{R}^{n \times n}$. Similar to above, we take derivatives to obtain

$$\dot{A} = \dot{Q}R + Q\dot{R}, \quad \dot{Q}^T Q + Q^T \dot{Q} = 0,$$

and we can again conclude from the second identity that \dot{Q} permits a representation

$$\dot{Q} = Q\tilde{A} + Z\tilde{B}, \quad \tilde{A} \in \mathbb{R}^{k \times k} \text{ skew-symmetric, } \tilde{B} \in \mathbb{R}^{(n-k) \times k}.$$

Sine Q here plays the role of Y in (12), we are only interested in $Z^T \dot{Q} = \tilde{B}$. Inserting into the identity for \dot{A} , we obtain

$$\dot{A} = Q(\tilde{A}R + \dot{R}) + Z\tilde{B}R.$$

Multiplying from the left with Z^T , we obtain $\tilde{B} = Z^T \dot{A} R^{-1}$ and finally, by inserting into (12),

$$\dot{Z} = -QR^{-T} \dot{A}^T Z. \tag{14}$$

Either (13) or (14) may be used to compute the Jacobi matrix for our Newton’s method. In our experience, the version (14) based on the QR decomposition is both faster and leads to smaller numerical errors, for which reason we prefer it.

We summarize both main results of this section in the following theorem.

Theorem 4 *Let a k -dimensional subspace $\text{span } A$ of \mathbb{R}^n and its $(n - k)$ -dimensional orthogonal complement $\text{span } Z$ be given, where $A \in \mathbb{R}^{n \times k}$ has full rank and $Z \in V_{n,n-k}$ has orthonormal columns. Further, let a perturbation of the former space in a tangent direction $\dot{A} \in \mathbb{R}^{n \times k}$ be given.*

- *Given the thin singular value decomposition $A = Y \Sigma V^T$, the corresponding derivative of the orthogonal complement has the tangent direction $\dot{Z} = -Y \Sigma^{-1} V^T \dot{A}^T Z \in T_{\text{span } Z} G_{n,n-k}$.*
- *Given the reduced QR decomposition $A = QR$, the corresponding derivative of the orthogonal complement has the tangent direction $\dot{Z} = -QR^{-T} \dot{A}^T Z \in T_{\text{span } Z} G_{n,n-k}$.*

5 Derivatives of the rational interpolant

5.1 Partial derivatives of the barycentric formula

The barycentric formula (4) may be written as

$$r(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) = \frac{q(\mathbf{z}, \mathbf{f}, \mathbf{w}, x)}{q(\mathbf{z}, \mathbf{1}, \mathbf{w}, x)}, \quad q(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) := \sum_{i=0}^N \frac{f_i w_i}{x - z_i}.$$

Since we will require the derivatives of the interpolant only at the local maxima of the error function which lie between the interpolation nodes, we may assume $x \neq z_j$. By elementary calculations we then obtain for $j = 0, \dots, N$ that

$$\begin{aligned} \frac{\partial q}{\partial z_j}(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) &= \frac{f_j w_j}{(x - z_j)^2}, \\ \frac{\partial q}{\partial f_j}(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) &= \frac{w_j}{x - z_j}, \\ \frac{\partial q}{\partial w_j}(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) &= \frac{f_j}{x - z_j} \end{aligned}$$

and therefore

$$\frac{\partial r}{\partial z_j}(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) = \frac{q(\mathbf{z}, (w_j(f_j - f_i))_{i=0}^N, \mathbf{w}, x)}{(x - z_j)^2 q^2(\mathbf{z}, \mathbf{1}, \mathbf{w}, x)}, \tag{15}$$

$$\frac{\partial r}{\partial f_j}(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) = r(\mathbf{z}, \mathbf{e}_j, \mathbf{w}, x), \tag{16}$$

$$\frac{\partial r}{\partial w_j}(\mathbf{z}, \mathbf{f}, \mathbf{w}, x) = \frac{q(\mathbf{z}, (f_j - f_i)_{i=0}^N, \mathbf{w}, x)}{(x - z_j) q^2(\mathbf{z}, \mathbf{1}, \mathbf{w}, x)}, \tag{17}$$

where $\mathbf{e}_j \in \mathbb{R}^{N+1}$ is the unit vector pointing in the j -th coordinate direction (here j is 0-based).

5.2 Derivatives of the rational interpolant with respect to the nodes

In the following we assume sufficient smoothness, in particular, $f \in C^1[a, b]$. Taking the total derivative of the interpolating rational function $\rho(\mathbf{x}, x) = \rho((\mathbf{z}, \hat{\mathbf{z}}), x)$ with respect to the primary and secondary nodes yields, by the chain rule,

$$\begin{aligned} \frac{d\rho}{dz_j}((\mathbf{z}, \hat{\mathbf{z}}), x) &= \frac{\partial r}{\partial z_j} + \frac{\partial r}{\partial f_j} f'(z_j) + \sum_{k=0}^N \frac{\partial r}{\partial w_k} \frac{\partial w_k(\mathbf{z}, \hat{\mathbf{z}})}{\partial z_j}, \quad j = 0, \dots, N, \\ \frac{d\rho}{d\hat{z}_j}((\mathbf{z}, \hat{\mathbf{z}}), x) &= \sum_{k=0}^N \frac{\partial r}{\partial w_k} \frac{\partial w_k(\mathbf{z}, \hat{\mathbf{z}})}{\partial \hat{z}_j}, \quad j = 1, \dots, \hat{N}, \end{aligned} \tag{18}$$

where we have omitted the arguments $(\mathbf{z}, \mathbf{f}, \mathbf{w}, x)$ of r for brevity. It therefore remains to compute the partial derivatives of the weight vector, which spans the nullspace of B , with respect to the interpolation nodes. To this end, we first compute the partial derivatives of the matrix B from (7) and then apply Theorem 4.

The derivatives of the Löwner matrix L from (5) with respect to the primary and secondary nodes, respectively, are

$$\begin{aligned} \frac{\partial L_{k\ell}}{\partial z_j} &= \frac{\delta_{j\ell}}{\hat{z}_k - z_\ell} (L_{k\ell} - f'(z_\ell)), \quad j = 0, \dots, N, \\ \frac{\partial L_{k\ell}}{\partial \hat{z}_j} &= \frac{\delta_{jk}}{\hat{z}_k - z_\ell} (f'(\hat{z}_k) - L_{k\ell}), \quad j = 1, \dots, \hat{N}. \end{aligned} \tag{19}$$

Note that the derivative of L by a primary node contains only a single nonzero column, whereas the derivative by a secondary node contains only a single nonzero row.

The derivatives of the degree constraint matrices (6) are, for $k = 0, \dots, \mu - 1$ and $\ell = 0, \dots, N$, given by

$$\frac{\partial [A(\mathbf{z}, \mathbf{f}, \mu)]_{k\ell}}{\partial z_j} = f'(z_\ell)z_\ell^k + kf(z_\ell)z_\ell^{k-1}, \quad \frac{\partial [A(\mathbf{z}, \mathbf{1}, \mu)]_{k\ell}}{\partial z_j} = kz_\ell^{k-1}, \quad j = 0, \dots, N, \tag{20}$$

where in both cases the last term is omitted for $k = 0$. Here we used $\mathbf{f} = (f(z_i))_{i=0}^N$. Since these matrices do not depend on the secondary nodes, the derivatives by \hat{z}_j , $j = 1, \dots, \hat{N}$, are 0.

For each primary or secondary node z_j or \hat{z}_j , we can now compute the corresponding partial derivative of the matrix B from (7) with respect to that node,

$$\dot{B}_{z_j} := \frac{\partial B}{\partial z_j} \in \mathbb{R}^{N \times (N+1)}, \quad \dot{B}_{\hat{z}_j} := \frac{\partial B}{\partial \hat{z}_j} \in \mathbb{R}^{N \times (N+1)},$$

by simply combining the formulae (19) and (20) for the partial derivatives of its components.

As described in Section 3, the weight vector $\mathbf{w} = \mathbf{w}(\mathbf{z}, \hat{\mathbf{z}})$ lies in the nullspace of the matrix B . Therefore, we are interested in the derivative $\dot{\mathbf{w}}$ of the vector spanning the nullspace when a derivative of B is taken, for which we use Theorem 4.

Let $\mathbf{w} \in \ker B$ with $\|\mathbf{w}\| = 1$, and let $\dot{B} \in \mathbb{R}^{N \times (N+1)}$ denote one of the derivatives \dot{B}_{z_j} or $\dot{B}_{\hat{z}_j}$. Denoting by

$$B^T = QR$$

the reduced QR decomposition of B^T with $Q \in \mathbb{R}^{(N+1) \times N}$ orthonormal and $R \in \mathbb{R}^{N \times N}$ upper triangular, we obtain from (14) that the corresponding derivative of the nullspace vector is

$$\dot{\mathbf{w}} = -QR^{-T} \dot{B} \mathbf{w}.$$

Thus,

$$\frac{\partial w_k(\mathbf{z}, \hat{\mathbf{z}})}{\partial z_j} = [-QR^{-T} \dot{B}_{z_j} \mathbf{w}]_k, \quad \frac{\partial w_k(\mathbf{z}, \hat{\mathbf{z}})}{\partial \hat{z}_j} = [-QR^{-T} \dot{B}_{\hat{z}_j} \mathbf{w}]_k, \quad (21)$$

and Theorem 4 is applicable since the barycentric formula depends only on the one-dimensional space spanned by \mathbf{w} , not the vector itself.

By combining (18), (15)–(17) and (21) we can now compute, for fixed $y \in [a, b]$ which does not coincide with any of the primary nodes, the gradient

$$\frac{d\rho}{dx_j}(\mathbf{x}, y), \quad j = 0, \dots, m + n + 1$$

of the interpolating rational function ρ with respect to the interpolation nodes.

5.3 Computing the Jacobian

For computing the Jacobian of Φ and therefore F , we require the total derivatives $\frac{d\rho}{dx_j}(\mathbf{x}, y_i(\mathbf{x}))$ with $\mathbf{x} = (\mathbf{z}, \hat{\mathbf{z}})$, where the abscissae of the local maxima y_i , $i = 0, \dots, m + n + 1$, themselves depend on \mathbf{x} ; see (1). By the chain rule,

$$\frac{d}{dx_j} \rho(\mathbf{x}, y_i(\mathbf{x})) = \frac{\partial \rho}{\partial y}(\mathbf{x}, y_i) \frac{\partial y_i}{\partial x_j} + \frac{d\rho}{dx_j}(\mathbf{x}, y_i),$$

where in the first term the derivative of the rational interpolant is taken with respect to the evaluation point y_i , whereas in the second term the derivative is taken with respect to the interpolation node x_j while the evaluation point y_i is kept constant. It follows that

$$\frac{\partial \Phi_i}{\partial x_j}(\mathbf{x}) = \frac{\partial}{\partial y_i} (f(y_i) - \rho(\mathbf{x}, y_i)) \frac{\partial y_i}{\partial x_j} - \frac{d\rho}{dx_j}(\mathbf{x}, y_i).$$

Since the nodes y_i are local extrema of the error $f - \rho(\mathbf{x}, \cdot)$, the term $\frac{\partial}{\partial y_i} (f(y_i) - \rho(\mathbf{x}, y_i))$ vanishes whenever $y_i \in (a, b)$. On the boundary, that is if $y_i \in \{a, b\}$, the product of $\frac{\partial}{\partial y_i} (f(y_i) - \rho(\mathbf{x}, y_i))$ and $\frac{\partial y_i}{\partial x_j}$ is 0 by a duality argument. Thus,

$$[\nabla \Phi]_{ij} = \frac{\partial \Phi_i}{\partial x_j}(\mathbf{x}) = -\frac{d\rho}{dx_j}(\mathbf{x}, y_i), \quad i = 0, \dots, m + n + 1, \quad j = 0, \dots, m + n,$$

meaning that the dependence of the local maxima y_i on \mathbf{x} can be ignored while computing the Jacobian matrix. These expressions are now computed as described in Section 5.2. The Jacobian of F is then given by

$$\nabla F = \left[\nabla \Phi \quad -\mathbf{w} \right] \in \mathbb{R}^{(m+n+2) \times (m+n+2)}. \quad (22)$$

6 The algorithm for best uniform rational approximation

The complete procedure for best uniform rational approximation is given in Algorithm 1. In the following we discuss some implementation issues.

- The partitioning of the node indices $k = 0, \dots, m + n + 1$ into primary and secondary ones can be done in various ways as long as it is done consistently. For stability, we enforce that each primary node interval (z_{j-1}, z_j) contains at most one secondary node, which is possible since there are always fewer secondary than primary nodes. Such alternating partitions were described in [24], where it was demonstrated numerically that this significantly improves the conditioning of the interpolation matrix.
- The local maxima y_i may be computed efficiently by means of a golden section search; see [23] for details.
- An initial guess for the error λ is obtained in the first iteration of the algorithm by taking the mean of the local error maxima.
- For the evaluation of formulae (18)–(20), the algorithm requires the first derivative f' , which we assume to be specified along with f itself. If it is not available, finite difference approximations could be used.

For higher values of the degrees m and n , numerical issues arise, and therefore an implementation of the algorithm with extended precision arithmetic is required in these cases. In particular, these issues arise in two places: while computing the decomposition (SVD or QR) of the interpolation constraint matrix B^T and while solving the linear system with the Jacobian ∇F . From (7), we see that B^T contains two Vandermonde-like matrices with $N - n$ and $N - m$ columns, respectively, which are well-known to be poorly conditioned. The problem is perhaps easier to observe in the SVD variant (13): although computing the SVD is a numerically stable operation, only the absolute error in each singular value is in general uniformly bounded (see, e.g., [10, 20]), which may incur arbitrarily large relative error for very small singular values. Thus, large errors may occur when forming the inverse Σ^{-1} of the singular values. The QR variant (14) exhibits similar issues with some diagonal entries of R becoming very small. It should be noted that numerically more advantageous procedures for finding the nullspace of these Vandermonde-like matrices have been proposed in [3, 13], but at present it is unclear how to compute the analytical Jacobian of F when using these approaches.

The solution of the linear system with the Jacobian ∇F is another source of potentially large numerical errors. In particular if the interpolation nodes have strongly varying orders of magnitude (as is the case when the nodes are clustered towards a boundary singularity of f), this matrix may become very poorly conditioned, often reaching condition numbers significantly larger than 10^{16} and thus necessitating the use of extended precision arithmetic.

Finally, here are some options for choosing the initial nodes at the start of the algorithm:

- equispaced, i.e., $x_i = a + (b - a) \frac{i+1}{m+n+2}$, $i = 0, \dots, m + n$;

function BESTRATIONAL($f \in C^1[a, b]$, degrees $m, n \in \mathbb{N}$, tolerance $\varepsilon > 0$, initial nodes $\mathbf{x}_0 \in \mathbb{R}^{m+n+1}$)

$N \leftarrow \max\{m, n\}$

for $k = 0, 1, 2, \dots$ **do**

partition the nodes $(\mathbf{z}, \hat{\mathbf{z}}) \leftarrow \mathbf{x}_k$ into $N+1$ primary and $m+n-N$ secondary nodes

compute $\mathbf{f} \leftarrow (f(z_j))_{j=0}^N$

compute $B = B(\mathbf{z}, \hat{\mathbf{z}}, \mathbf{f}) \in \mathbb{R}^{N \times (N+1)}$ from (7)

compute full QR decomposition

$$B^T = [Q \ \mathbf{w}] \begin{bmatrix} R \\ 0 \end{bmatrix}$$

$r \leftarrow$ barycentric rational function (4) with $(\mathbf{z}, \mathbf{f}, \mathbf{w})$

compute interval-wise local maxima of the error ($x_{k,-1} := a, x_{k,m+n+1} := b$)

$$y_i = \arg \max_{x \in (x_{k,i-1}, x_{k,i})} |f(x) - r(x)|, \quad i = 0, \dots, m+n+1$$

if $k = 0$ **then**

$\lambda \leftarrow$ mean of the local errors $|f(y_i) - r(y_i)|$

end if

compute nonlinear residual $\mathbf{r}_k \leftarrow F(\mathbf{x}_k, \lambda)$ from (2)

if $\|\mathbf{r}_k\| < \varepsilon$ **then**

return r

end if

compute Jacobian $\nabla F(\mathbf{x}_k, \lambda)$ from (18), (21), (22)

compute Newton step $(\mathbf{d}_x, d_\lambda) \leftarrow -(\nabla F(\mathbf{x}_k, \lambda))^{-1} \mathbf{r}_k$

determine step size τ by rule (3)

update

$$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k + \tau \mathbf{d}_x, \quad \lambda \leftarrow \lambda + \tau d_\lambda$$

end for

end function

Algorithm 1 Newton’s method for best uniform rational approximation.

- Chebyshev nodes of the first kind, i.e., $x_i = a + (b - a)(1 - \frac{1}{2} \cos(\frac{2i+1}{2(m+n+1)}\pi))$, $i = 0, \dots, m+n$;
- via the initialization procedure for the BRASIL algorithm [23] where nodes are iteratively placed at the abscissa of the largest interpolation error;
- by performing a number of iterations of the BRASIL algorithm [23] itself. Although the BRASIL algorithm was originally only specified for degrees (n, n) , it can straightforwardly be extended to degree (m, n) rational approximation by adopting an interpolation routine along the lines of that described in Section 3, although it tends to be numerically less robust in this case;
- by re-using the interpolation nodes of a ”neighboring” approximation when computing several approximations with varying degree. For instance, after computing the nodes associated with the best approximation with degrees (m, n) , these

nodes will often be good starting values for the best approximation with degree $(m + 1, n - 1)$ or $(m - 1, n + 1)$, which both have the same total number of nodes.

Generally, we observe that the step size choice (3) is remarkably effective at establishing global convergence for our Newton's method, and in our examples we obtained convergence for all possible choices of initial nodes as long as the resulting interpolant does not possess a pole within the interval (a, b) . However, the number of iterations can be significantly reduced by choosing good initial nodes, and either the BRASIL initialization routine or BRASIL itself usually do a good job at this.

7 Numerical examples

In the following we demonstrate the performance of our novel method, Algorithm 1, in several examples. The algorithm was implemented in Python using the second author's `baryrat`¹ software package for barycentric rational approximation. The used implementation of the algorithm is now also publicly available in that package. For extended precision calculations, the `gmpy2`² package was used. In all examples, the stopping criterion used for the algorithm was the reduction of the Euclidean norm of the residual (2) below $\varepsilon = 10^{-16}$.

The results were obtained on a Linux workstation with an Intel Xeon E-2276G CPU and 32 GB RAM.

7.1 Example 1

We compute the best rational approximation in $\mathcal{R}_{m,n}$ of the function

$$f_1(x) = \frac{x^{1/4}}{1 + 10x^{1/4}}, \quad x \in [0, 1],$$

where we vary the degrees m and n both of the numerator and the denominator. This example is motivated by applications in fractional diffusion equations and has been studied in several past publications [14, 18, 23]. As initial nodes, we choose the nodes obtained using 100 iterations of the BRASIL algorithm [23], or those obtained using only the initialization routine of BRASIL if the former encountered numerical difficulties. The Newton algorithm was executed using 150 digits of precision. The chosen tolerance $\varepsilon = 10^{-16}$ leads to very accurate results: for example, for $m = n = 10$, the values of the local maxima of the error function $|f_1(x) - r(x)|$ differ from each other by less than $6 \cdot 10^{-23}$. In other words, the equioscillation property of Theorem 1 holds to machine precision.

In Table 1, we list both iteration numbers and obtained maximum norm errors for varying degrees m and n . The very low iteration numbers on the diagonal are owed to the fact that BRASIL performs very well in the case $m = n$, so the starting nodes it produces are sufficiently close to the exact solution that the Newton algorithm can

¹<https://github.com/c-f-h/baryrat>

²<https://pypi.org/project/gmpy2/>

Table 1 Iteration numbers (in bold) and maximum errors for best rational approximation with varying degrees m and n of the numerator and denominator, respectively, for Example 1

n	10	20	30	40
m				
10	4 6.25727e-5	10 3.06698e-5	24 2.45576e-5	31 2.13897e-5
20	9 3.02712e-5	3 1.39512e-6	16 6.91611e-7	47 5.42020e-7
30	20 2.41140e-5	18 6.86436e-7	3 6.85694e-8	31 3.50347e-8
40	27 2.09309e-5	46 5.36097e-7	28 3.48498e-8	3 5.22661e-9

immediately take full steps, $\tau = 1$, and converge superlinearly. On the other hand, for the off-diagonal cases, BRASIL does not produce good initial nodes, and therefore the Newton algorithm has an initial phase of roughly linear convergence with step sizes τ from (3) often significantly smaller than 1 before entering the superlinear convergence phase. This behavior will be demonstrated in a plot for Example 2 below.

In general, we observe that if the degrees m and n differ significantly, best uniform rational approximations are significantly more difficult to compute in terms of numerical stability and iteration numbers. A large part of the numerical issues stems from the matrix B given in (7) which contains as a submatrix a Vandermonde-like matrix with $\max\{m, n\} - \min\{m, n\}$ rows, leading to the well-known poor conditioning associated with such matrices.

7.2 Example 2

To demonstrate that the algorithm also works in the case of reduced smoothness, namely, $f \notin C^1$, we perform best rational approximation in $\mathcal{R}_{m,n}$ of the absolute value function,

$$f_2(x) = |x|, \quad x \in [-1, 1],$$

where in place of the first derivative, we use the sign function, $f'_2(x) = \text{sign}(x)$.

Due to f_2 being an even function, we can only apply our algorithm with degrees (m, n) such that $m+n$ is odd and the total number of nodes is even. For our examples, we always choose the degree of the denominator to be $n = m - 1$.

Since BRASIL cannot approximate the function f_2 , we instead use Chebyshev nodes of the first kind as our starting nodes in this example. The calculations were performed using 100 digits of precision.

In Fig. 1, left, we display the maximum norm error, iteration numbers and used computation time for varying degrees m . Despite the poor choice of initial nodes, the iteration numbers increase only moderately with the degree; we observe a roughly linear dependence. A plot of the convergence history, meaning the residual norm plotted over the iterations for varying degrees $m \in \{10, 20, 30, 40\}$, is shown in the right-hand side of the figure. Again, we observe that after an initial phase of

m	error	iter	time (s)
10	4.39366e-4	16	0.33
15	4.60366e-5	19	0.88
20	6.91304e-6	26	2.3
25	1.30438e-6	36	5.4
30	2.89265e-7	45	9.8
35	7.24786e-8	48	15.4
40	1.99997e-8	57	25.9

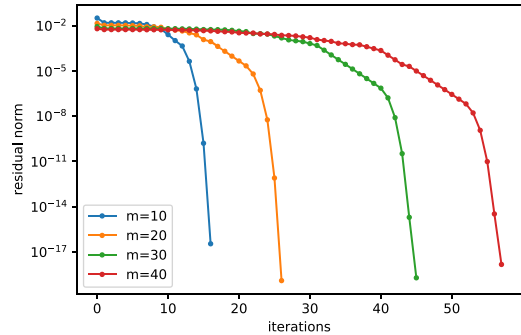


Fig. 1 Results for Example 2. *Left:* Degree m , maximum error, number of iterations, and computation time, with $n = m - 1$. *Right:* Convergence history of the residual norm for four different choices of m

roughly linear convergence, where the step size τ is smaller than 1, we then enter the superlinear regime where $\tau = 1$ and full Newton steps are taken once sufficiently close to the solution.

7.3 Conclusions and outlook

We have presented a Newton's method for best uniform rational approximation with arbitrary degrees (m, n) of numerator and denominator. The main advantages of the novel method over the BRASIL algorithm [23] are its superlinear convergence and the fact that it works well for arbitrary degrees, whereas BRASIL performs best in the case $m = n$. The main drawback is the requirement of using extended precision arithmetic, whereas BRASIL can be performed using standard IEEE double precision.

The goal of an algorithm based on the determination of the interpolation nodes with the excellent numerical stability of BRASIL, but superlinear convergence, remains elusive. One step towards the stabilization of our method would be to use the Arnoldi process described in [13] for determining the nullspace of (7), but applying Newton's method requires computing also the derivative of this procedure, which appears challenging. Nevertheless, this may be a suitable topic for future work.

We stress that Theorems 3 and 4 on the derivatives of orthogonal complements of full-rank matrices may be of general interest, and we could not find these formulae in the literature.

Funding This work was supported by the bilateral project KP-06-Austria/8/2019 (WTZ BG 03/2019), funded by Bulgarian National Science Fund and OeAD (Austria). The second author received support from the Austrian Science Fund (FWF) grant P 33956-NBL.

Declarations

Data Availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Absil, P.-A., Mahony, R., Sepulchre, R.: Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. *Acta Appl. Math.* **80**(2), 199–220 (2004). <https://doi.org/10.1023/b:acap.0000013855.14971.91>
2. Achieser, N.I.: Theory of approximation. dover books on advanced mathematics. Dover Publications. ISBN 9780486671291 (1992)
3. Berrut, J.-P., Mittelmann, H.D.: Matrices for the direct determination of the barycentric weights of rational interpolation. *J. Comput. Appl. Math.* **78**(2), 355–370 (1997). [https://doi.org/10.1016/s0377-0427\(96\)00163-x](https://doi.org/10.1016/s0377-0427(96)00163-x)
4. Berrut, J.-P., Trefethen, L.N.: Barycentric Lagrange interpolation. *SIAM Rev.* **46**(3), 501–517 (2004). <https://doi.org/10.1137/s0036144502417715>
5. Berrut, J.-P., Baltensperger, R., Mittelmann, H.D.: Recent developments in barycentric rational interpolation. In: Trends and applications in constructive approximation, pp. 27–51. Birkhäuser Basel. https://doi.org/10.1007/3-7643-7356-3_3 (2005)
6. Braess, D.: Nonlinear approximation theory. Springer Berlin Heidelberg. ISBN 978-3-642-64883-0. <https://doi.org/10.1007/978-3-642-61609-9> (1986)
7. Carpenter, A.J., Ruttan, A., Varga, R.S.: Extended numerical computations on the $1/9$ conjecture in rational approximation theory. In: Rational approximation and interpolation, pp. 383–411. Springer Berlin Heidelberg. <https://doi.org/10.1007/bfb0072427> (1984)
8. Danczul, T., Hofreither, C.: On rational Krylov and reduced basis methods for fractional diffusion. *Journal of Numerical Mathematics.* <https://doi.org/10.1515/jnma-2021-0032> (2021)
9. Danczul, T., Hofreither, C., Schöberl, J.: A unified rational Krylov method for elliptic and parabolic fractional diffusion problems. Technical Report arXiv:2103.13068. <https://arxiv.org/abs/2103.13068> (2021)
10. Demmel, J., Gu, M., Eisenstat, S., Slapničar, I., Veselić, K., Drmač, Z.: Computing the singular value decomposition with high relative accuracy. *Linear Algebra. Appl.* **299**(1–3), 21–80 (1999). [https://doi.org/10.1016/s0024-3795\(99\)00134-2](https://doi.org/10.1016/s0024-3795(99)00134-2)
11. Driscoll, T.A., Hale, N., Trefethen, L.N.: Chebfun Guide. Pafnuty Publications. <http://www.chebfun.org/docs/guide/> (2014)
12. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.* **20**(2), 303–353 (1998). <https://doi.org/10.1137/s0895479895290954>
13. Filip, S.-I., Nakatsukasa, Y., Trefethen, L.N., Beckermann, B.: Rational minimax approximation via adaptive barycentric representations. *SIAM J. Sci. Comput.* **40**(4), A2427–A2455 (2018). <https://doi.org/10.1137/17m1132409>
14. Georgieva, I., Hofreither, C.: A newton’s method for best uniform polynomial approximation. Technical Report 2021-46, RICAM. <https://www.ricam.oeaw.ac.at/files/reports/21/rep21-46.pdf> (2021)
15. Giles, M.: An extended collection of matrix derivative results for forward and reverse mode automatic differentiation. Technical report, University of Oxford. <https://ora.ox.ac.uk/objects/uuid:8d0c0a29-c92b-4153-a1d2-38b276e93124> (2008)
16. Güttel, S.: Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM-Mitteilungen* **36**(1), 8–31 (2013). <https://doi.org/10.1002/gamm.201310002>
17. Harizanov, S., Lazarov, R., Margenov, S., Marinov, P., Vutov, Y.: Optimal solvers for linear systems with fractional powers of sparse SPD matrices. *Numerical Linear Algebra with Applications* **25**(5), e2167 (2018). <https://doi.org/10.1002/nla.2167>
18. Harizanov, S., Lazarov, R., Margenov, S., Marinov, P.: The best uniform rational approximation: applications to solving equations involving fractional powers of elliptic operators. Number 9 in Lectures Notes in Computer Science and Technologies of the IICT-BAS. Institute of Information and Communication Technologies at the Bulgarian Academy of Sciences. <https://pa.iict.bas.bg/lcst/2019/09-2019-HLMM.pdf> (2019)
19. Harizanov, S., Lazarov, R., Margenov, S., Marinov, P., Pasciak, J.: Analysis of numerical methods for spectral fractional elliptic equations based on the best uniform rational approximation. *Chin. J. Comput. Phys.* **408**, 109285 (2020). <https://doi.org/10.1016/j.jcp.2020.109285>
20. Higham, N.J.: QR factorization with complete pivoting and accurate computation of the SVD. *Linear Algebra. Appl.* **309**(1–3), 153–174 (2000). [https://doi.org/10.1016/s0024-3795\(99\)00230-x](https://doi.org/10.1016/s0024-3795(99)00230-x)
21. Higham, N.J.: The numerical stability of barycentric Lagrange interpolation. *IMA J. Numer. Anal.* **24**(4), 547–556 (2004). <https://doi.org/10.1093/imanum/24.4.547>

22. Hofreither, C.: A unified view of some numerical methods for fractional diffusion. *Comput. Math. Appl* **80**(2), 332–350 (2020). <https://doi.org/10.1016/j.camwa.2019.07.025>
23. Hofreither, C.: An algorithm for best rational approximation based on barycentric rational interpolation. *Numerical Algorithms*. <https://doi.org/10.1007/s11075-020-01042-0> (2021)
24. Ionitã, A.C.: Lagrange Rational Interpolation and its Applications to Approximation of Large-Scale Dynamical Systems. PhD Thesis, Rice University, Houston (2013)
25. Knockaert, L.: A simple and accurate algorithm for barycentric rational interpolation. *IEEE Signal Process. Lett.* **15**, 154–157 (2008). <https://doi.org/10.1109/lsp.2007.913583>
26. Nakatsukasa, Y., Sète, O., Trefethen, L.N.: The AAA algorithm for rational approximation. *SIAM J. Sci. Comput* **40**(3), A1494–A1522 (2018). <https://doi.org/10.1137/16m1106122>
27. Schneider, C., Werner, W.: Some new aspects of rational interpolation. *Math. Comput.* **47**(175), 285–285 (1986). <https://doi.org/10.1090/s0025-5718-1986-0842136-8>
28. Trefethen, L.N.: *Approximation Theory and Approximation Practice*. Other Titles in Applied Mathematics. SIAM. ISBN 9781611972405 (2013)
29. Varga, R.S., Carpenter A.J.: Some numerical results on best uniform rational approximation of x^α on $[0, 1]$. *Numerical Algorithms* **2**(2), 171–185 (1992). <https://doi.org/10.1007/bf02145384>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.