Preconditioning for rational approximation

Vinesha Peiris[1], Nir Sharon[2†], Nadezda Sukhorukova[3*†], Julien Ugon[4†]

[1]Curtin Centre for Optimisation and Decision Science, Curtin University, Bently, 6102, Western Australia, Australia.
[2]School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.
[3*]Mathematics, Swinburne university of Technology, John Street, Hawthorn, 3122, Victoria, Australia.
[4]School of Info Technology, Deakin University, 221 Burwood Highway, Burwood, 3125, Victoria, Australia.

*Corresponding author(s). E-mail(s): nsukhorukova@swin.edu.au;
Contributing authors: m.peiris@curtin.edu.au;
nir.sharon@math.tau.ac.il; julien.ugon@deakin.edu.au;
†These authors contributed equally to this work.

Abstract

In this paper, we show that minimax rational approximations can be enhanced by introducing a controlling parameter on the denominator of the rational function. This is implemented by adding a small set of linear constraints to the underlying optimization problem. The modification integrates naturally into approximation models formulated as linear programming problems. We demonstrate our approach through several numerical examples, including a multivariate rational approximation of the Korteweg–de Vries (KdV) equation from fluid dynamics.

Keywords: controlling parameter, minimax rational approximation, quasiconvex programming, preconditioning

MSC Classification: 15A60 , 65F35 , 90C47 , 65D15 , 65K10 , 49M37

1 Introduction

Rational approximations offer a strong balance between approximation accuracy and computational efficiency. While polynomial-based methods are simpler and more straightforward to implement, they often fall short in terms of approximation quality. On the other end of the spectrum, free-knot splines can achieve high accuracy but are typically more complex and computationally demanding to construct. Several well-established methods exist for computing rational approximations for the univariate function approximation. These include the AAA algorithm [1], Remez-type methods [2], and approaches based on linear programming, such as the bisection method [3, 4] and the differential correction method [5, 6]. In [7], the authors extended the bisection and differential correction methods to multivariate settings. A comprehensive literature review on uniform approximation by rational function approximation, its geometrical properties and its connection with polynomial approximation can be found in [8].

The main contribution of this paper is the development of a preconditioning technique based on a denominator control parameter. This technique can be efficiently implemented by introducing additional linear constraints, a straightforward modification in linear programming-based frameworks, but one that is not readily applicable to methods such as AAA or Remez-type algorithms. Numerical experiments demonstrate that the proposed approach enhances approximation accuracy and enables an efficient calculation of a larger-scale problem.

Rational approximation is a very attractive option for uniform approximation problems: the corresponding optimisation problems are quasiconvex and can be solved efficiently through the solution of larger linear programming problems. Another attractive approximation technique is neural networks, in particular, deep learning. The efficiency of neural networks was established in the universal approximation theorem [9, 10], which states that neural networks with a single hidden layer can approximate any continuous function to any accuracy, provided that the activation functions are non-polynomials. This result led to a broad area of research on what activation function is best for practical applications.

It appeared that rational functions are an attractive choice for activation functions. This can be explained by the fact that their parameters can be learned by the network itself, and it was shown in [11] that good approximation results can be achieved. Such networks are also known as rational networks. Another example, where rational approximation improves the classification power of deep learning, can be found in [12]. Therefore, rational approximations are also used as auxiliary subproblems in larger approximation problems and hence the improvement of the existing rational approximation approaches as well as the development of new algorithms is crucial.

2 Methodology

A uniform rational approximation can be formulated as follows:

$$\text{minimise} \max_{t \in Q} \left| f(t) - \frac{A^T G(t)}{B^T H(t)} \right|, \tag{1}$$

where Q is a compact convex set (continuous settings) or a finite set of discretisation points from a convex set (discrete settings). In most practical problems, discrete settings are applied. In the case of rational approximation, the components of vectors $G(t)$ and $H(t)$ are the monomials, the rational function degree (m, n) means that the degree of the numerator and denominator polynomials are m and n, respectively.

It was shown in [13] that the optimal uniform rational approximation is unique as a rational function. However, since the vectors (A, B) and $(\alpha A, \alpha B)$ correspond to the same rational function, we need a scaling approach. Some common methods exist, such as keeping all the coefficients of the denominator between -1 and 1. In our approach, we fix one of the coefficients of the denominator. In addition, we require that $B^T H(t) > 0$. In practice, this can be implemented as the linear constraint $B^T H(t) \geq \delta$, where δ is a small positive number. Therefore, bounding the denominator from above can be considered a preconditioning approach. This approach may improve the performance of the bisection and differential correction methods. Indeed, by governing the ratio

$$C_r = \frac{\max_t B^T H(t)}{\min_t B^T H(t)} \leq \frac{u}{\ell}, \tag{2}$$

we control the constraint matrices appearing in these linear programming problems. Adding the constraint (2) to Problem (1) maintains its quasiconvex structure, and therefore we can apply a bisection method to solve it.

In the following two sections, we present the results of numerical experiments for univariate functions (Section 3) and multivariate functions (Section 4).

3 Numerical performance: univariate approximation

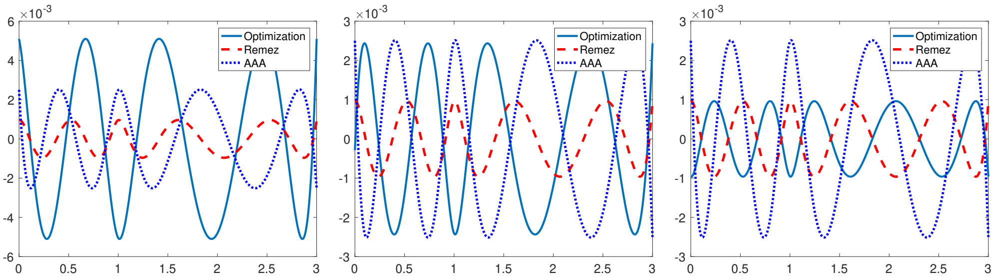
We evaluate three test functions using three different approximation methods: our optimization-based approximation with preconditioning via the bisection method (for details on the bisection, Appendix A), the Remez algorithm, as implemented in [2], and the AAA algorithm from [1].

For our optimization-based method, we set the bisection tolerance near double-precision accuracy (10^{-15}), and we use 400 equidistant sample points over the target interval. The AAA algorithm uses the same set of sample points, while the Remez algorithm operates with a function handle and has access to arbitrary function evaluations. After computing all approximations, we evaluate them at 1000 uniformly spaced points across the interval and report the maximum absolute deviation from the original function as the uniform error. The source code is available at: https://github.com/nirsharon/preconditioningRationalApprox.

The test functions are a cubic spline function (f_1), a cusp (f_2), and an oscillatory function(f_3). Table 1 includes the functions, their domains, and the fundamental parameters: the degrees of the rational approximation (m, n) and the denominator upper bound u. The lower bound is fixed at $\ell = 1$ and therefore $C_r \leq u$.

Test function	$[a, b]$	(m, n)	C_r of (2)
$f_1(x) = \begin{cases} -x^3 + 6x^2 - 6x + 2 & x < 1 \\ x^3 & 1 \leq x \end{cases}$	$[0, 3]$	$(4, 5)$ and $(5, 5)$	2,4,8
$f_2(x) = \lvert x\rvert^{2/3}$	$[-1, 2]$	$(6, 6)$	100
$f_3(x) = \cos 9x + \sin 11x$	$[-1, 1]$	$(7, 7)$	50

Table 1: The test functions, their domains, and approximation parameters.

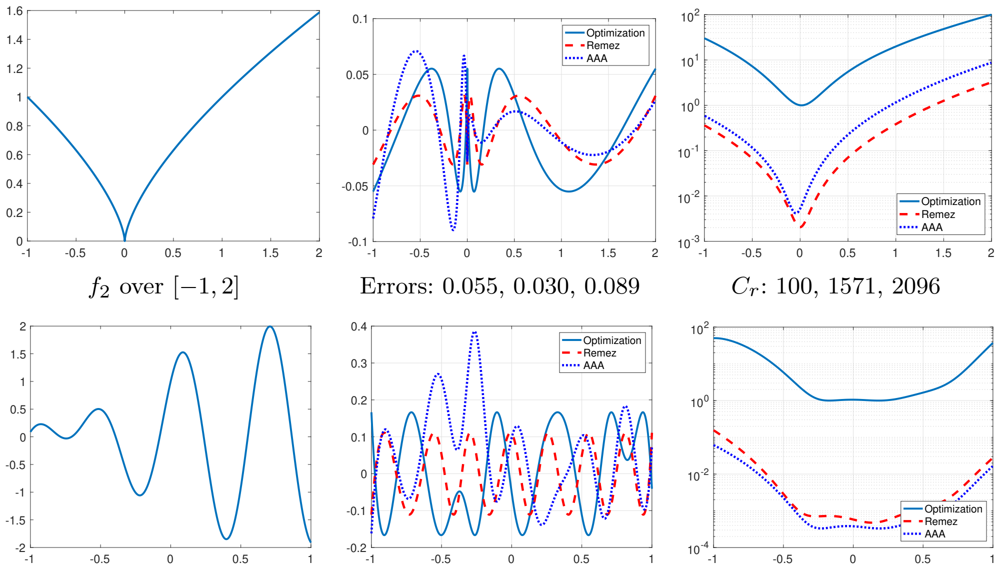


(a) $u = 2$ (b) $u = 4$ (c) $u = 8$

Fig. 1: The effect of changing denominator bounds with fixed degrees. We set the rational polynomial degrees to (4,5) and compare our method with the (5,5) AAA and (4,5) Remez approximations.

In Figure 1, we examine the function f_1 and study the effect of varying the denominator bound u in equation (2), using the degree $(4, 5)$ rational approximations. The Remez algorithm achieves a uniform error of approximately 0.0009, while the AAA method (restricted to $m = n$) yields 0.0025 with a degree $(5, 5)$ approximation. For the optimization-based approach, a tight bound of $u = 2$ results in an error of 0.0051. Increasing the bound to $u = 4$ improves the result, slightly outperforming AAA (Figure 1b). The AAA denominator norm is $C_r = 3.91$, which lies within the feasible region of the optimization, explaining the improved accuracy. At $u = 8$ (Figure 1c), the optimization method matches the minimax performance of Remez, also achieving a uniform error of 0.0009. Notably, the Remez denominator norm is $C_r = 6.86$, which is within the optimization's constraint set and thus explains the result. This shows the agility of the optimization-based approach for rational approximation.

In the second example, we consider the test functions f_2 and f_3, reporting the corresponding error rates and absolute changes in the denominator. The results, shown in Figure 2, demonstrate that the controlling parameter provides an effective means of balancing approximation accuracy with constraints on the denominator, particularly valuable when such restrictions are required. For f_2, the optimization-based method achieves an error rate comparable to that of the Remez algorithm, but using only discrete data and significantly faster optimization, while producing a denominator that varies by an order of magnitude less than those obtained by the other two methods. In the case of f_3, the AAA algorithm performs poorly, while the optimization approach again yields superior accuracy with a well-controlled denominator.



f_2 over $[-1, 2]$ | Errors: 0.055, 0.030, 0.089 | C_r: 100, 1571, 2096

f_3 over $[-1, 1]$ | Errors: 0.16, 0.11, 0.38 | C_r: 50, 327, 185

Fig. 2: Comparison of the three methods for rational approximation — optimization, Remez, and AAA, using the test functions f_2 and f_3.

4 Numerical performance: an application in the multivariate approximation

In this section, we present a multivariate function approximation example, demonstrating how the performance of the optimization method is improved by introducing control parameters into the denominator of the rational function. It should be noted that multivariate function approximation is much harder than it is for univariate functions, even in the simple case of polynomial approximation [14].

Consider the Korteweg–de Vries (KdV) equation which is a nonlinear partial differential equation of the form

$$u_t = -uu_x - u_{xxx},$$

which models the wave propagation on shallow water surfaces. This equation appears in many applications, including fluid dynamics, waves and symmetry. Our goal is to approximate the solution u on the given spatio temporal points, (x, t) by using the multivariate bisection method. The algorithm is implemented in MATLAB (R2023a).

The original dataset was created in [15] and is publicly available. The solution u is computed on the domain $(x, t) \in [-20, 20] \times [0, 40]$ with a discretization of $[512 \times 201]$ points and an initial guess of $u(x, 0) = -\sin(\pi x/20)$. The experiments are based on a subset of the full domain, referred to as the reduced domain, which is constructed by selecting every k^{th} point ($k = 10, 20$) along each dimension along with the corresponding function values. Both the approximation and the uniform error are evaluated over this reduced domain for consistency.

This approach is adopted to prevent excessive data volume from causing crashes in MATLAB during computation. The original values of u on the reduced domain are shown in Figure 3. To approximate u, we apply the multivariate bisection method, using Chebyshev monomials in place of classical monomials within the rational approximation.

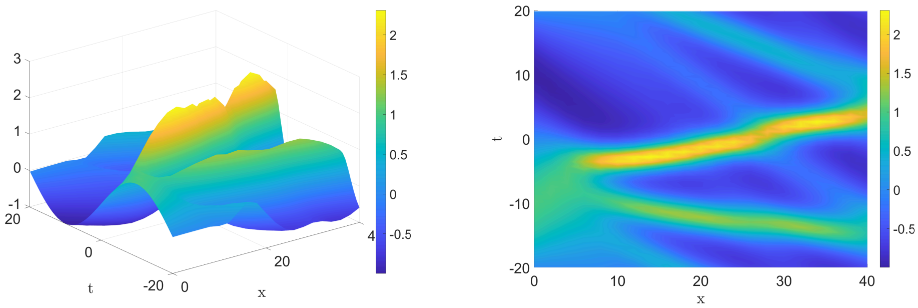


Fig. 3: KdV solution plotted over the reduced domain: 3D (on left) and 2D (on right) landscape view of the original solution.

As the degree of the rational function increases, MATLAB code can not handle its calculation properly and crashes. In particular, in our experiments, MATLAB could handle the approximations up to the degree $(18, 18)$ without any preconditioning, the uniform error went down to 0.034389 ($k = 10$) and 0.014062 ($k = 20$). Any higher degrees could not be processed. As a remedy, we restricted the upper bound of the denominator to $l = 100$ and increased the degree to $(20, 20)$. The program completed the tasks with the uniform error being 0.027058 ($k = 10$) and 0.010817 ($k = 20$).

We now present an example comparing the two approximations: one with degree $(18, 18)$ and the other with degree $(20, 20)$. The latter experiment tends to crash with the usual setting of the algorithm (MATLAB) and therefore, we introduce an additional constraint that restricts the denominator from above (upper bound of 100). This additional constraint is simple and can be simply added to the constrain set of our optimization problem and the problem remains in the class of linear programming problems. The uniform errors are recorded in Table 2 and the approximations are presented in Figure 4.

Adding the extra linear constraint prevented the code from crashing. The program terminated normally and found the optimal solution. As expected, the uniform approximation error is lower in the case of the higher degree polynomials in the numerator and denominator of the rational approximation: 0.034 for degree $(18, 18)$ and 0.027 for degree $(20, 20)$. Figure 3 (exact function, landscape view) is very similar to Figure 4 (right).

Therefore, we can conclude that the application of the preconditioning approach, proposed in this paper, is also efficient for multivariate function approximation.

Degree (n,m)	Uniform error
(18,18)	0.034389
(20,20)	0.027058

Table 2: Uniform error

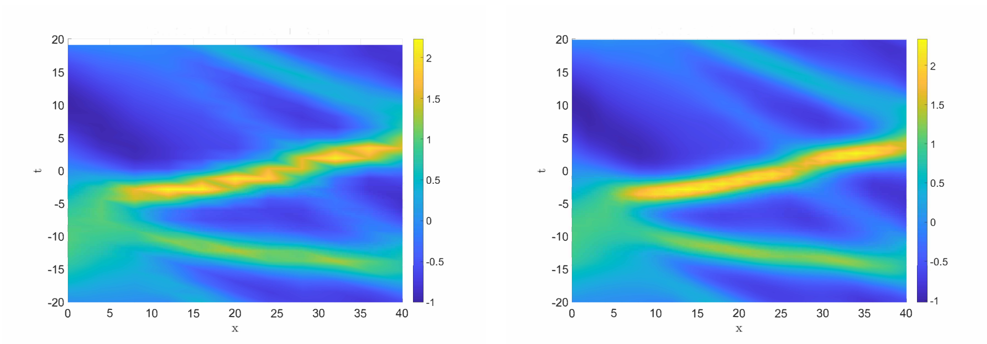


Fig. 4: KdV solution approximated over the reduced domain: degree (18,18) on left and (20,20) on right (landscape view).

5 Conclusions and future research

Our numerical experiments show that the controlling parameters can lead to more accurate approximations. In particular, our algorithm can accommodate higher-degree polynomials in both the numerator and the denominator. Similar results were observed in our experiments with the differential correction method, but this problem requires more detailed study.

The additional linear constraints can be naturally added to the corresponding optimization problems without paying a significant computational price: the problems remain linear programming problems. This approach can be seen as a preconditioning technique for solving large linear programming problems, appearing in the bisection and differential correction methods.

Declarations

- Funding
  Nir Sharon is partially supported by the NSF-BSF award 2019752 and the DFG award 514588180. Vinesha Peiris, Nadezda Sukhorukova, and Julien Ugon are supported by the Australian Research Council (ARC), Solving hard Chebyshev approximation problems through nonsmooth analysis (Discovery Project DP180100602).

- Conflict of interest/Competing interests
  No conflict of interest and/or competing interests.
- Ethics approval and consent to participate
  This research does not need ethic approval or consent.
- Consent for publication
  All authors give their consent for publication.
- Data availability
  All references to the data sources are provided. The data are available.
- Materials availability
  Not applicable.
- Code availability
  Code can be available on request.
- Author contribution
  Equal contribution from the authors. During the preparation of this work, the authors used Overleaf to improve the readability of the paper. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

A The Bisection method for rational approximation

Historically, the implementation of many minimax rational approximations is based on solving linear programming problems [16, 17]. The bisection method is looking at the problem from the point of view of quasiconvexity. The problem can be reformulated as:

$$\min \quad z \tag{3}$$

subject to

$$f(t) - \frac{A^T G(t)}{B^T H(t)} \leq z, \tag{4}$$

$$\frac{A^T G(t)}{B^T H(t)} - f(t) \leq z, \tag{5}$$

$$B^T H(t) > 0. \tag{6}$$

Define a bisecting interval $z \in [l, u]$ and assign $z = \frac{1}{2}(u + l)$. If the set of constraints in (4) has a feasible solution, update the upper bound $u = z$; otherwise, update the lower bound $l = z$. Terminates when $u - l < \varepsilon$, for a predefined threshold ε.

References

[1] Nakatsukasa, Y., Sète, O., Trefethen, L.N.: The AAA algorithm for rational approximation. SIAM Journal on Scientific Computing **40**(3), 1494–1522 (2018)

[2] Pachón, R., Trefethen, L.N.: Barycentric-Remez algorithms for best polynomial approximation in the chebfun system. BIT Numerical Mathematics **49**(4), 721 (2009)

[3] Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York, NY, USA (2010)

[4] Peiris, V., Sharon, N., Sukhorukova, N., Ugon, J.: Generalised rational approximation and its application to improve deep learning classifiers. Applied Mathematics and Computation **389**, 125560 (2021) https://doi.org/10.1016/j.amc.2020.125560

[5] Cheney, E.W., Southard, T.: A survey of methods for rational approximation, with particular reference to a new method based on a forumla of darboux. SIAM Review **5**(3), 219–231 (1963)

[6] Barrodale, I.: Best rational approximation and strict quasi-convexity. SIAM Journal on Numerical Analysis **10**(1), 8–12 (1973)

[7] Aghili, A., Sukhorukova, N., Ugon, J.: Bivariate rational approximations of the general temperature integral. Journal of Mathematical Chemistry **59**(9), 2049–2062 (2021)

[8] Alimov, A., Tsar'kov, I.: Geometric Approximation Theory. Springer, Switzerland (2021)

[9] Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems **2**, 303–314 (1989)

[10] Leshno, M., Vladimir, Y.L., Pinkus, A., Schocken, S.: Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Networks **6**(6), 861–867 (1993) https://doi.org/10.1016/S0893-6080(05)80131-5

[11] Boullé, N., Nakatsukasa, Y., Townsend, A.: Rational neural networks. Advances in Neural Information Processing Systems **33**, 14243–14253 (2020)

[12] Peiris, V., Roshchina, V., Sukhorukova, N.: Artificial neural networks with uniform norm-based loss functions. Advances in Computational Mathematics **50**(3), 31 (2024)

[13] Achieser, N.I.: Theory of Approximation. Frederick Ungar, New York, New York (1965)

[14] Roshchina, V., Sukhorukova, N., Ugon, J.: Uniqueness of solutions in multivariate chebyshev approximation problems. Optimization Letters **18**, 33–55 (2024)

[15] Raissi, M.: Deep hidden physics models: Deep learning of nonlinear partial differential equations. Journal of Machine Learning Research **19**(25), 1–24 (2018)

[16] Barrodale, I., Powell, M., Roberts, F.: The differential correction algorithm for rational L_∞-approximation. SIAM Journal on Numerical Analysis **9**(3), 493–504 (1972)

[17] Ralston, A.: Rational Chebyshev approximation by Remes' algorithms. Numer. Math. **7**(4), 322–330 (1965)