NEAR OPTIMAL RATIONAL APPROXIMATIONS OF LARGE DATA SETS

ANIL DAMLE, GREGORY BEYLKIN, TERRY HAUT AND LUCAS MONZÓN

ABSTRACT. We introduce a new computationally efficient algorithm for constructing near optimal rational approximations of large (one-dimensional) data sets. In contrast to wavelet-type approximations, these new approximations are effectively shift invariant. We note that the complexity of current algorithms for computing near optimal rational approximations prevents their use for large data sets.

In order to obtain a near optimal rational approximation of a large data set, we first construct its intermediate B-spline representation. Then, by using a new rational approximation of B-splines, we arrive at a suboptimal rational approximation of the data set. We then use a recently developed fast and accurate reduction algorithm for obtaining a near optimal rational approximation from a suboptimal one. Our approach requires first splitting the data into large segments, which may later be merged together, if needed. We also describe a fast algorithm for evaluating these rational approximations. In particular, this allows us to interpolate the original data to any grid.

One of the practical applications of our algorithm is the compression of audio signals. To demonstrate the potential competitiveness of our approach, we construct a near optimal rational approximation of a piano recording.

1. INTRODUCTION

In this paper we develop an algorithm for constructing near optimal rational representations of functions using as input a large number of equally spaced samples. Examples of such data sets include, among others, digitized versions of musical recordings and continuous seismic records. Optimal or near optimal rational approximations provide both a method for data compression, as well as a useful representation for further data analysis. We observe that rational approximations are more efficient than wavelet decompositions. In fact, the ability of wavelets to compress signals may be justified via optimal rational approximations, see e.g., [11, Chapter 11]. Furthermore,

Key words and phrases. optimal rational approximations, nonlinear approximations, signal compression, fast algorithms, spline interpolation.

This research was partially supported by NSF grant DMS-1009951, DOE/ORNL grant 4000038129 and NSF MCTP grant DMS-0602284. A.D. is currently supported by NSF Fellowship DGE-1147470.

Appl. Comput. Harmon. Anal., v. 35, no. 2, pp.251-263,
http://dx.doi.org/10.1016/j.acha.2012.08.011.

in contrast to wavelet decompositions, rational functions are closed under translations and, thus, optimal rational approximations are shift invariant. Indeed, shifting an optimal rational approximation yields the optimal approximation of the shifted function or data.

Our rational representations are optimal in the sense that, for a given accuracy of approximation, the number of poles is minimal. We say that the approximation is "near optimal" if, instead of the desired accuracy ϵ, our algorithms yield accuracy ϵ', where ϵ' is slightly smaller than ϵ. In such case the number of poles may not be minimal in the strict sense (we note that we have an *a posteriori* check to identify such situation, if needed). We use the term "suboptimal", if we know that the number of poles definitely exceeds the optimal number (for a given accuracy).

For functions given analytically or for functions described by a relatively small number of samples, there are several methods for obtaining their near optimal rational approximations [5, 6, 7]. For a large data set these methods are impractical due to their computational complexity. On the other hand, computing a wavelet decomposition of a large data set does not present a difficulty since its computational cost is linear in the number of samples; we use these facts in our approach.

We first compute a B-spline representation of the data, which provides a simple and efficient method for a transition to a suboptimal rational representation. For this purpose, we construct a new rational approximation of B-splines, where the poles are arranged on a rectangular grid aligned with the location of spline knots. We then split the data into large segments, and compute suboptimal rational approximations for each segment. Finally, we compute a near optimal rational approximation using a recently developed, fast and accurate algorithm in [10].

Although the example provided here is compression of audio recordings, the algorithm may be used to compress and analyze other types of signals, e.g., signals obtained by continuous, global seismic monitoring. In particular, we view compression via near optimal rational approximations as the first step in signal analysis since the poles carry frequency and time information. As shown in [6], poles of near optimal rational approximations concentrate near the singularities of functions. For signals, this corresponds to locations of rapid change, such as occurring when a piano key is struck or at wave arrivals in seismic recordings. The location of the poles also carries information about local frequency content of the signal in a manner similar to wavelets, i.e., the logarithmic distance of these locations from the real axis corresponds to wavelet scales.

We start in Section 2 by providing the background information on the key existing algorithms that facilitate the development of our new approach. Next, in Section 3, we construct a rational approximation (with special properties) of a B-spline to be used in intermediate computations. Then, in Section 4, we describe in detail the algorithm for constructing near optimal

rational approximations of large data sets. We present an overview of the algorithm and examine the specifics of each step. Section 5 contains numerical examples that validate the performance of the algorithm. Finally, Section 6 contains concluding remarks.

2. Preliminary Considerations

2.1. **An algorithm for finding a near optimal rational approximation.** We start by describing the method in [5] to obtain a near optimal rational approximation from samples of the Fourier transform of the function. For functions with a fast decaying Fourier transform this method is closely connected to the theory developed by Adamjan, Arov and Krein (AAK) [1, 2, 3].

Given samples $\hat{f}\left(a\frac{k}{2N}\right)$, $k = 0, 1, \ldots, 2N$, (that sufficiently oversample $\hat{f}(\xi)$ on the interval $\xi \in [0, a]$), we seek a representation of $\hat{f}(\xi)$ of the form

$$\left| \hat{f}(\xi) - \sum_{j=1}^{M} w_j e^{-\eta_j \xi} \right| \leq \epsilon, \tag{1}$$

where ϵ is the desired accuracy. The algorithm proceeds as follows:

- Construct a $N+1 \times N+1$ Hankel matrix of the form $H_{kl} = \widehat{f}\left(a\frac{k+l}{2N}\right)$, $k, l = 0, \ldots, N$.
- Find a singular vector $\mathbf{u} = (u_0, \ldots, u_N)$ that solves the con-eigenvalue problem $\mathbf{H}\mathbf{u} = \sigma\bar{\mathbf{u}}$, with σ selected according to the target accuracy ϵ. We may find $\mathbf{u}$ by solving the eigenvalue problem for

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{0} & \mathbf{H} \\ \mathbf{H}^* & \mathbf{0} \end{bmatrix}, \tag{2}$$

  which yields the singular values of $\mathbf{H}$, $\sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_M \geq \cdots \geq \sigma_N$. We choose the singular value σ_M, so that $\sigma_M/\sigma_0 \approx \epsilon$. Typically the singular values decay exponentially fast so that $M \ll N$.
- Compute M appropriate roots (see [5, 6]) of the polynomial $u(z) = \sum_{n=0}^{N} u_n z^n$ and denote them γ_j. Given values γ_j, the exponents η_j in (1) are computed as

$$\eta_j = 2N \log \gamma_j, \tag{3}$$

  where we use the principle value of the logarithm.
- Compute the weights w_j in (1) by solving the least-squares Vandermonde system

$$\sum_{j=1}^{M} w_j \gamma_j^k = \widehat{f}\left(a\frac{k}{2N}\right), \quad 0 \leq k \leq 2N. \tag{4}$$

If no a priori information is available, we may use all N roots of $u(z)$, and then determine the M necessary roots by selecting those with corresponding weights of magnitude greater than the target accuracy.

Following [6], from (1) we obtain the rational representation

$$f(x) = -2\mathcal{R}e\left(\sum_{j=1}^{M}\frac{w_j}{2\pi i x-\eta_j}\right) = -2\mathcal{R}e\left(\sum_{j=1}^{M}\frac{u_j+iv_j}{x-t_j-is_j}\right) \tag{5}$$
$$= -2\sum_{j=1}^{M}\frac{u_j\,(x-t_j)-v_js_j}{(x-t_j)^2+s_j^2},$$

where t_j, s_j, u_j, v_j are real values such that

$$\frac{\eta_j}{2\pi i} = t_j + is_j \quad \text{and} \quad \frac{w_j}{2\pi i} = u_j + iv_j.$$

As illustrated in [6], the positions of the poles $t_j \pm is_j$ carry information about the location of singularities of the function f. Furthermore, the representation for any translate of $f(x)$ in (5) is readily obtained by simply shifting the poles.

2.2. **Algorithm for reduction of a suboptimal rational approximation.** An effective and accurate algorithm for reducing the number of poles of a rational function while maintaining some target accuracy is given in [10]. The formulation of the problem may also be found in [5] and is based on results in [3]. Although we present this algorithm for rational trigonometric functions, a similar algorithm exists for functions defined on the real line [6].

We start with a real valued rational function $f(z)$,

$$f(z) = \sum_{j=1}^{M_0}\frac{d_j}{z-\mu_j} + \sum_{j=1}^{M_0}\frac{\overline{d_j}z}{1-\overline{\mu_j}z} + d_0, \tag{6}$$

with $d_0 \in \mathbb{R}$, $d_j, \mu_j \in \mathbb{C}$ and $0 < |\mu_j| < 1$. Our goal is to find a rational function $r(z)$ of the form

$$r(z) = \sum_{j=1}^{M}\frac{r_j}{z-\eta_j} + \sum_{j=1}^{M}\frac{\overline{r_j}z}{1-\overline{\eta_j}z} + d_0,$$

with fewer poles than $f(z)$ such that

$$\left|r(e^{2\pi ix}) - f(e^{2\pi ix})\right| < \epsilon \quad \forall x \in [0,1).$$

The steps of the algorithm in [10] are as follows.

- Consider the Cauchy matrix $C_{kj}(\mu_k, d_j)$,

$$C_{kj} = \frac{\sqrt{d_k}\sqrt{\overline{d}_j},}{1-\mu_k\overline{\mu}_j}, \quad k,j = 1,\ldots,M_0.$$

We use the algorithm in [10] to solve the con-eigenproblem

$$Cu = \sigma_M\bar{u}$$

for a con-eigenvalue σ_M and con-eigenvector $u = (u_1, u_2, \cdots, u_{M_0})^t$. The con-eigenvalues are ordered $\sigma_0 \geq \sigma_1 \geq \ldots \geq \sigma_{M_0-1}$ and $\sigma_M/\sigma_0 \approx$

ϵ. In contrast to standard algorithms, the con-eigenvalues (and con-eigenvectors) are computed with high relative accuracy in $\mathcal{O}\left(M^2 M_0\right)$ operations.

- Find all the roots inside the unit disk of the function

$$v(z) = \frac{1}{\sigma_M} \sum_{j=1}^{M_0} \frac{\sqrt{\overline{d_j}}\,\overline{u_j}}{1 - \overline{\mu_j} z}.$$

  Note that there are exactly M roots ν_l of $v(z)$ inside the unit disk based on results from [3].
- Finally solve for the residuals r_l of $r(z)$ by solving the $M \times M$ linear system

$$\sum_{j=1}^{M} \frac{r_j}{1 - \nu_j \overline{\nu_k}} = \sum_{j=1}^{M_0} \frac{d_j}{1 - \mu_j \overline{\nu_k}}.$$

Using this algorithm, we obtain $\|f - r\| \approx \sigma_M$, which provides a near optimal representation of $f(z)$ using only M pairs of conjugate-reciprocal poles, ν_l and $\overline{\nu_l}^{-1}$. The computational complexity of this algorithm is $\mathcal{O}\left(M^2 M_0\right)$, where M is the number of resulting poles and M_0 is the original number of poles. Since typically $M \ll M_0$, this algorithm is effectively linear in its practical use.

2.3. **Spline representations.** We use an intermediate representation via B-splines as the first step towards computing the (near) optimal rational approximation. Although theoretically we may use scaling functions of any wavelet-type basis, the choice of B-splines reduces the computational cost of this intermediate step.

We recall the definition of the m^{th} degree B-spline as

$$\beta_m(x) = \beta_{m-1}(x) * \beta_0(x),$$

with

$$\beta_0(x) = \begin{cases} 1, & |x| \leq \frac{1}{2} \\ 0, & \text{otherwise,} \end{cases}$$

(see e.g., [8]). For convenience, we only use B-splines of odd degree. It is easy to show that, in this case, β_m is a piecewise polynomial of degree m with knots on the integers and supported on $[-(m+1)/2, (m+1)/2]$. To represent periodic functions, we use periodized versions of B-splines. Let us introduce the 1-periodic function

$$a_m(\omega) = \sum_{j \in \mathbb{Z}} |\widehat{\beta}_m(\omega + j)|^2 = \sum_{l=-\frac{m-1}{2}}^{\frac{m-1}{2}} \beta_m(l) e^{-2\pi i l \omega}.$$

Given a uniformly sampled 1-periodic function f, we seek the coefficients α_j such that

$$f\left(\frac{k}{2N}\right) = \sum_{j=0}^{2N} \alpha_j \beta_m(k-j), \quad k = 0, \ldots, 2N. \tag{7}$$

The algorithm in [4, 12] rapidly computes the coefficients α_j in (7) using the Fast Fourier Transform (FFT). It performs the following steps:

- Set $f_k = f(\frac{k}{2N})$ and compute, for $k = 0, \ldots, 2N$,
$$\widehat{f}_k = \sum_{n=0}^{2N} f_n e^{\frac{-2\pi i}{2N+1} kn}$$
using the FFT.
- Compute, for $k = 0, \ldots, 2N$,
$$\widehat{\alpha}_k = \frac{\widehat{f}_k}{a_m(\frac{k}{2N+1})}.$$
- The B-spline coefficients are now obtained via the FFT as
$$\alpha_j = \frac{1}{2N+1} \sum_{n=0}^{2N} \widehat{\alpha}_n e^{\frac{2\pi i}{2N+1} jn}, \quad j = 0, \ldots, 2N$$

This algorithm requires $\mathcal{O}(N \log N)$ operations. The details may be found in the appendix in [12].

3. Rational representation of B-splines

In this section we construct rational approximations of B-splines. In our construction we force the real parts of the poles to be integers $l \in \mathbb{Z}$, so that the poles are aligned with the knots of the B-spline. As we explain below, this reduces the cost of intermediate computations.

Specifically, we are looking for a suboptimal rational approximation of the form (5), with poles $l \pm i\tau_k$, so that

$$\left| \beta_m(x) + 2 \sum_{l=-\frac{m+1}{2}}^{\frac{m+1}{2}} \sum_{k=1}^{R} \frac{u_{k,l}(x-l) - v_{k,l}\tau_k}{(x-l)^2 + \tau_k^2} \right| \leq \epsilon, \tag{8}$$

where the number of rows of poles, R, will be chosen later. We note that the constraint on the real part of the poles arranges them on a rectangular grid (see Figure 2).

We start by computing a near optimal rational approximation of a B-spline following the approach in [6]. For a given m, we evaluate $\widehat{\beta}$ at a sufficient number of samples; specifically for $m = 7$ we have

$$h_n = \widehat{\beta}_m\left(\frac{n}{32}\right), \quad n = 0, 1, \ldots, 800, \tag{9}$$

where

$$\widehat{\beta}_m(\xi) = \left(\frac{\sin \pi\xi}{\pi\xi}\right)^{m+1}, \tag{10}$$

and use the algorithm in Section 2.1 to construct a near optimal rational approximation.

An example of a near optimal rational approximation of a B-spline of degree $m = 3$ may be found in [6]. As observed in that paper, the poles concentrate towards the locations of the knots of the B-spline since its third derivative is discontinuous at these points. In our application we would like to use a higher degree B-spline to lessen the impact of discontinuities and obtain fewer poles. In Figure 1 we present the results for a near optimal approximation of a 7th degree B-spline using the same approach as in [6]. Since the poles, $t_j \pm is_j$, appear in complex conjugate pairs, in Figure 1 we display (on a $\log_{10}$ scale) only those with negative imaginary part.

We then seek a suboptimal rational representation of $\beta(x)$ with poles in the locations indicated in (8) and use the near optimal approximation to select the parameters τ_k in (8). Taking into account that the poles closer to the real line are responsible for the high frequency content of the representation, whereas those furthest away capture the lower frequency content, we limit the range for the imaginary parts of our suboptimal poles by using the corresponding maximum, s^+, and minimum, s^-, of the near optimal poles. We select three rows of poles, i.e., $R = 3$ in (8), by choosing imaginary parts $\tau_1 = s^+$, $\tau_3 = s^-$, and

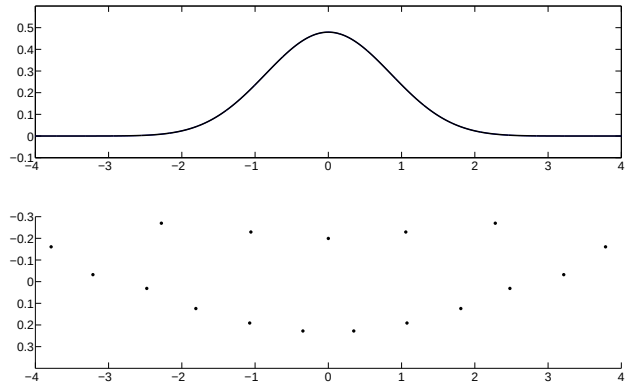
$$\tau_2 = e^{\frac{1}{2}(\log \tau_1 + \log \tau_3)}.$$

The real part for all of these poles are at locations l, where $l = -\frac{m+1}{2}, \ldots, \frac{m+1}{2}$ (recall that m is odd). The choice of three rows of poles is based on the degree of the B-spline and our accuracy requirements (see Figure 2(b)) and may be different in other applications.

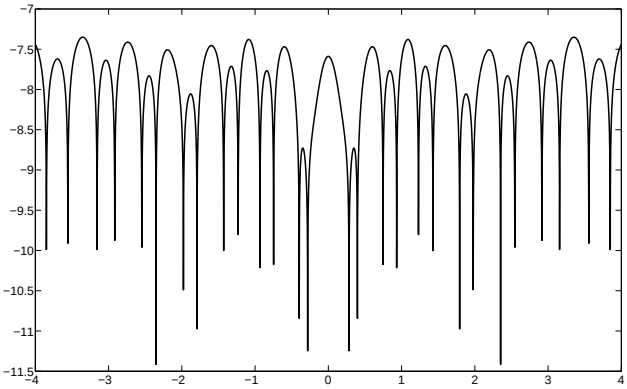
Once the location of poles is fixed, the weights in (8) are obtained by solving a linear system of equations. Unlike in the case of the near optimal approximation, the computation of weights in the Fourier domain leads to a severely ill-conditioned Vandermonde system (4). Instead, we directly discretize the representation for $\beta(x)$ in (8) and compute the weights by minimizing the l_∞-norm of the residual. We note that while a B-spline has a compact support, its rational approximation does not and, therefore, we must control the error to within the desired accuracy outside the B-spline support as well. This property of the approximation is particularly important for the merging algorithm in Section 4.

To discretize $\beta(x)$ in (8), we choose a set of points $\{x_n\}_{n=0}^{N_s}$,

$$x_n = \begin{cases} -50 + \frac{46n}{499}, & n = 0, \ldots, 499, \\ -4 + \frac{8(n-499)}{232}, & n = 500, \ldots, 730, \\ 4 + \frac{46(n-731)}{499}, & n = 731, \ldots, 1230, \end{cases}$$



(a)



(b)

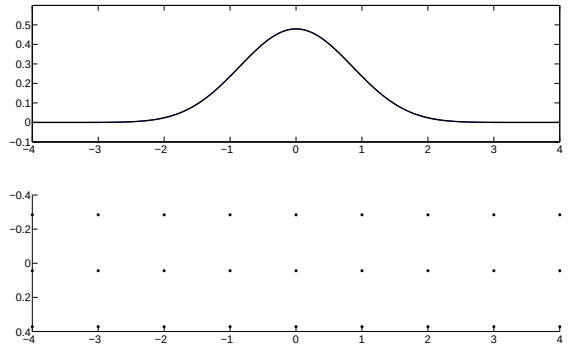
FIGURE 1. (a) Near optimal rational representation of a B-spline of degree 7 and its 16 poles (displayed on a $\log_{10}$ scale for their imaginary part). (b) Associated error on a $\log_{10}$ scale.

which generates a dense grid within the support of the B-spline and a relatively sparse grid outside. We then consider the overdetermined linear system

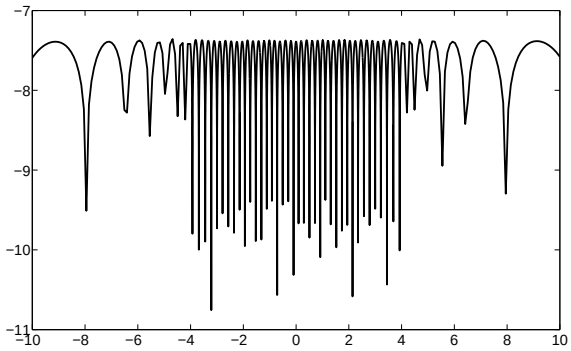
$$\beta_m(x_n) = -2 \sum_{l=-\frac{m+1}{2}}^{\frac{m+1}{2}} \sum_{k=1}^{R} \left(\frac{u_{k,l}(x_n - l)}{(x_n - l)^2 + \tau_k^2} - \frac{v_{k,l}\tau_k}{(x_n - l)^2 + \tau_k^2} \right), \quad n = 0, \ldots, N_s,$$

and solve for the real coefficients $u_{k,l}$ and $v_{k,l}$. This linear system is of size $(N_s + 1) \times 2 \cdot (m + 2)$ (where in our case $N_s = 1230$, $m = 7$ and $R = 3$) and we solve the problem by minimizing the l_∞-norm of the residual using

the optimization package CVX [9]. The resulting absolute error is shown in Figure 2(b).



(a)



(b)

FIGURE 2. (a) Rational representation in (8) of the B-spline of degree 7 and its 27 poles arranged on a rectangular grid (displayed using a $\log_{10}$ scale for their imaginary part). (b) Associated approximation error on a $\log_{10}$ scale. Outside $[-10, 10]$ the error is smaller than within this interval.

The purpose of this suboptimal representation of the B-spline is to convert a B-spline decomposition of the original signal to a suboptimal rational representation. The special choice of pole locations implies that the number of poles of the resulting suboptimal representation exceeds the number of B-spline coefficients in (7) only by a factor of 3. In fact, our choice of B-splines as a basis was motivated by this moderate increase in the number of terms in comparison to other wavelet-type decompositions. The choice of the 7th degree spline is dictated by the target accuracy for our final signal approximation. For greater accuracy, higher order B-splines should be used and their suboptimal approximation may be obtained by the same procedure.

4. Near optimal rational approximations

We now briefly describe the key steps of our algorithm for computing near optimal rational representations of large data sets. We assume that the signal is negligible at its start and end and has a large number of samples so that it is impractical to use the algorithm in Section 2.1 directly. Instead, our algorithm involves the following steps:

1: In order to apply the algorithm described in Section 2.3, we use a partition of unity to subdivide the signal into overlapping sections so that we may treat each section as a periodic function. The size of these sections should be appropriate for an efficient use of the FFT but otherwise is arbitrary. We compute the B-spline coefficients for each section. We then combine the B-spline coefficients from each section to get the B-spline coefficients for the entire signal.

2: We group the computed B-spline coefficients from Step **1** into consecutive segments (which are unrelated to the subdivision used in Step **1**). The size of these segments should be appropriate to guarantee efficiency of the reduction algorithm in Section 2.2. By using the B-spline approximation constructed in Section 3, we obtain a suboptimal rational representation of each segment.

3: On each segment we use the reduction algorithm in Section 2.2 to obtain a near optimal rational approximation for that segment.

4: We now merge the rational representations of adjacent segments. As we explain below, only adjacent segments interact with each other so that this step does not have to be done globally. Furthermore, only poles near the boundary between segments need to be merged and then reduced. This step may be considered optional since the overlap of the functions associated with adjacent segments is small in comparison to the length of each segment, so that the potential reduction of the number of poles is a small percentage of their total number.

Once a near optimal rational representation has been constructed, we need a fast algorithm (see below) for its evaluation to generate samples. We note that besides recovering the original signal, rational representations also allow us to interpolate the original signal to an arbitrary grid. This property of the representation is useful in many applications; for example, a higher sampling rate improves the quality of sound reproduced by speakers.

4.1. **Steps of the algorithm.** We now describe each step in some detail.

1: We use a partition of unity as our windows, and we note that there may be significant overlap between adjacent windows. The only requirement for the windows is a smoothly decaying transition region as to avoid introducing additional frequency content into the signal, and sufficient decay as to obtain partitions that are appropriate for the use of the FFT. We then use the algorithm in Section 2.3 to

compute the B-spline coefficients for each section of the signal. Once the B-spline coefficients for each section are found, by adding components from different sections, we obtain the B-spline coefficients for the entire data set. The cost of this step is $\mathcal{O}(N_{signal} \log N_{part})$, where N_{signal} is the total number of samples and N_{part} is the number of samples in each section (assuming they are of the same length). As a result, we obtain a representation

$$f(x) = \sum_{j=0}^{N_{signal}-1} \alpha_j \beta_m(x-j). \tag{11}$$

2: Given the signal in the form (11), we split the sum into segments of length P for further processing,

$$f_p(x) = \sum_{j=pP}^{(p+1)P-1} \alpha_j \beta_m(x-j) = \sum_{j=0}^{P-1} \alpha_{j+pP} \beta_m(x-j-pP), \tag{12}$$

where

$$p = 0, \ldots, \left\lfloor \frac{N_{signal}-1}{P} \right\rfloor. \tag{13}$$

In our construction we allow incomplete segments.
For each segment we replace the B-splines by their suboptimal rational representation constructed in Section 3 and obtain the suboptimal representation

$$\widetilde{f}_p(x) = -2 \sum_{j=0}^{P-1} \left(\alpha_{j+pP} \sum_{l=-\frac{m+1}{2}}^{\frac{m+1}{2}} \sum_{n=1}^{R} \frac{u_{n,l}(x-j-pP-l) - v_{n,l}\tau_n}{(x-j-pP-l)^2 + \tau_n^2} \right). \tag{14}$$

Thus, the suboptimal approximation in each segment requires $(P+m+1)\cdot R$ poles (with our choice of seventh degree B-splines, $R=3$, see Section 3).

3: For each p in (13), we apply the reduction algorithm described in Section 2.2 to the suboptimal approximation $\widetilde{f}_p(x)$. We obtain a near optimal representation with M_p^{opt} poles,

$$\tilde{f}_p^{opt}(x) = -2\mathcal{R}e \sum_{k=1}^{M_p^{opt}} \frac{w_k^p}{2\pi i x - \eta_k^p}, \qquad \left\| \widetilde{f}_p^{opt}(x) - f_p(x) \right\| < \epsilon.$$

4: In order to merge the near optimal approximations from adjacent segments, we may use the reduction algorithm once again. We note that, for our purposes, we need to merge only poles near the boundary between adjacent segments keeping unchanged the poles far away from the boundary region. To accomplish this, we consider the function (we wish to reduce) generated by the poles and their corresponding residues. By requesting a slightly higher accuracy across

the support of both segments, we preserve the overall accuracy of the merged approximation. In our experiments, we reduce the set of poles located at most 64 units (measured by step size of the original signal) from the midpoint of the overlapping region between adjacent segments. This selection is made to assure that the positions of the nodes possibly affected by the splitting of the data into segments are adjusted by the reduction algorithm. The slightly higher accuracy (of one half extra digit) assures that the untouched poles are not impacted by this merge, and hence we do not need to recompute their weights. We obviously do not obtain the optimal approximation over the support of the two segments, but we claim that the approximation is near optimal, both in terms of the number of poles and their locations.

Given the optimal representations $\widetilde{f}_p^{opt}(x)$ for p in (13), we merge the adjacent representations and denote the entire merged representation as

$$\widetilde{f}(x) = \sum_p \widetilde{f}_p^{merged}(x).$$

We also note that the observed reduction in the number of poles within two adjacent segments is minimal and, therefore, this step may be considered optional in practice.

4.2. **Fast algorithm for evaluation of rational representations.** A Fast Multipole type-method provides a fast algorithm for the evaluation of rational functions. An efficient approach (see [13, 5]) is based on approximating samples of rational functions of the form $1/(x - t_j \pm i\, s_j)$ by decaying exponentials. Specifically, we need to evaluate the function f in (5) at values $x_1 < x_2 < \cdots < x_K$. Denoting $f_k = -f(x_k)$, we have

$$f_k = \sum_{j=1}^{M} \left(\frac{u_j + iv_j}{x_k - t_j - is_j} + \frac{u_j - iv_j}{x_k - t_j + is_j} \right), \quad k = 1, \ldots K, \tag{15}$$

where both M and K are large. We observe that, for the applications of oscillatory signal compression, the parameters s_j describing the distance of poles from the real axis are bounded, $|s_j| \leq s$, where s is small in comparison with the range of t_j, $j = 1, \ldots, M$. We split the summation in (15) into three

parts,

$$
\begin{aligned}
f_k = f_k^+ + f_k^- + f_k^{local} &= \sum_{x_k - t_j \geq \alpha s} \left(\frac{u_j + iv_j}{x_k - t_j - is_j} + \frac{u_j - iv_j}{x_k - t_j + is_j} \right) \\
&+ \sum_{t_j - x_k \geq \alpha s} \left(\frac{u_j + iv_j}{x_k - t_j - is_j} + \frac{u_j - iv_j}{x_k - t_j + is_j} \right) \qquad (16) \\
&+ \sum_{|x_k - t_j| < \alpha s} \left(\frac{u_j + iv_j}{x_k - t_j - is_j} + \frac{u_j - iv_j}{x_k - t_j + is_j} \right),
\end{aligned}
$$

and evaluate f_k^+, f_k^- and f_k^{local} separately, where that of f_k^{local} proceeds directly. The condition on the factor α is decribed below ($\alpha = 5$ is a typical choice). It remains to describe an algorithm for evaluating f_k^+ since f_k^- is computed in a similar manner.

We have

$$
\begin{aligned}
f_k^+ &= \sum_{x_k - t_j \geq \alpha s} \left(\frac{u_j + iv_j}{x_k - t_j - is_j} + \frac{u_j - iv_j}{x_k - t_j + is_j} \right) \\
&= 2 \sum_{x_k - t_j \geq \alpha s} \int_{-\infty}^{\infty} e^{-e^y (x_k - t_j) + y} \left(u_j \cos(e^y s_j) - v_j \sin(e^y s_j) \right) dy. \qquad (17)
\end{aligned}
$$

The effective range of integration in (17) is finite due to the exponential ($y \to -\infty$) and super-exponential ($y \to \infty$) decay of the integrand. Our choice of the factor α prevents an excessive oscillatory behavior of the integrand within that range. In order to obtain an exponential approximation of the form

$$
f_k^+ = \sum_{t_j \leq x_k} \sum_{l=1}^{L} \lambda_{l,j} e^{-\mu_l (x_k - t_j)}, \quad \alpha s \leq x_k - t_j \leq T, \quad \mathcal{R}e(\mu_l) > 0, \qquad (18)
$$

(where T is sufficiently large to accommodate a given segment of the signal), we may now proceed as in [5, 7]. Indeed, we discretize the integral in (17) to any desired precision and use an appropriate algorithm to reduce the number of terms.

In (18) we may switch the order of summation and, as a result, construct a recursion (see [13, 5]). Denoting

$$
q_{k,l} = \sum_{t_j \leq x_k} \lambda_{l,j} e^{-\mu_l (x_k - t_j)},
$$

we obtain

$$
\begin{aligned}
q_{k+1,l} &= \sum_{t_j \leq x_{k+1}} \lambda_{l,j} e^{-\mu_l (x_{k+1} - t_j)} \\
&= e^{-\mu_l (x_{k+1} - x_k)} q_{k,l} + \sum_{x_k < t_j \leq x_{k+1}} \lambda_{l,j} e^{-\mu_l (x_{k+1} - t_j)}.
\end{aligned}
$$

This recursion leads to an $\mathcal{O}(L \cdot K) + \mathcal{O}(L \cdot M)$ algorithm for computing f_k^+.

5. Numerical Examples

We have computed several approximations using the algorithm from Section 4. Since one of the potential applications for this method is a compression scheme, we illustrate our algorithm using a large data set from a high quality audio recording. Generally, audio recordings are done with 16 bits per sample. This means that the maximum accuracy possible is 2^{-15}, provided that the maximum amplitude of the signal is 1. This section contains an example of finding a near optimal approximation for a single segment, $f_p(x)$, and then demonstrates the procedure of merging the approximations of adjacent segments and, thus, constructing a rational representation of the whole data set.

5.1. **Rational representation for a single segment.** First, using the algorithm in Section 4, we compute a B-spline representation for the entire signal. We then consider the performance of our algorithm to approximate, with accuracy 6×10^{-4}, the function $f_p(x)$ in (12). Figure 3 displays the rational approximation with 44 poles, the locations of those poles, and the associated error. We display the error $\left| f_p(x) - \widetilde{f}_p(x) \right|$, where $\widetilde{f}_p(x)$ is the resulting rational approximation and note that we achieve the same accuracy vis-à-vis the original signal (in a slightly smaller interval) due to the fact that the computation of a B-spline representation is accurate to machine precision.

Although this level of compression is reasonable, we note that in order to develop a complete compression scheme additional steps should be taken to encode the parameters of the near optimal rational approximation. Furthermore, by taking into account the level of signal noise, we may reduce the number of poles in the representation. Indeed, by examining the decay of the singular values of the Cauchy matrix formed for the reduction algorithm, we can develop an approximation tailored to the level of noise in the signal [6].

5.2. **Merging of adjacent segments.** One of the key benefits of the approximation method used is that each pole of the near optimal rational representation only locally influences the reconstruction. For this reason merging the near optimal rational approximations of adjacent segments minimally alters the original pole locations for the two segments. To illustrate this, we compute near optimal rational approximations for two adjacent segments, $f_p(x)$ and $f_{p+1}(x)$, each of length 512. Figure 4 shows the near optimal approximations of the two adjacent segments along with the error. The first segment requires 30 poles and the second requires 29 poles.

Figure 5 shows the representation for the merged windows, pole locations and associated error.

The poles that were within 64 sample distances of the boundary between segments were merged. The merged approximation required 58 poles, which means that a minimal reduction has taken place with respect to the total

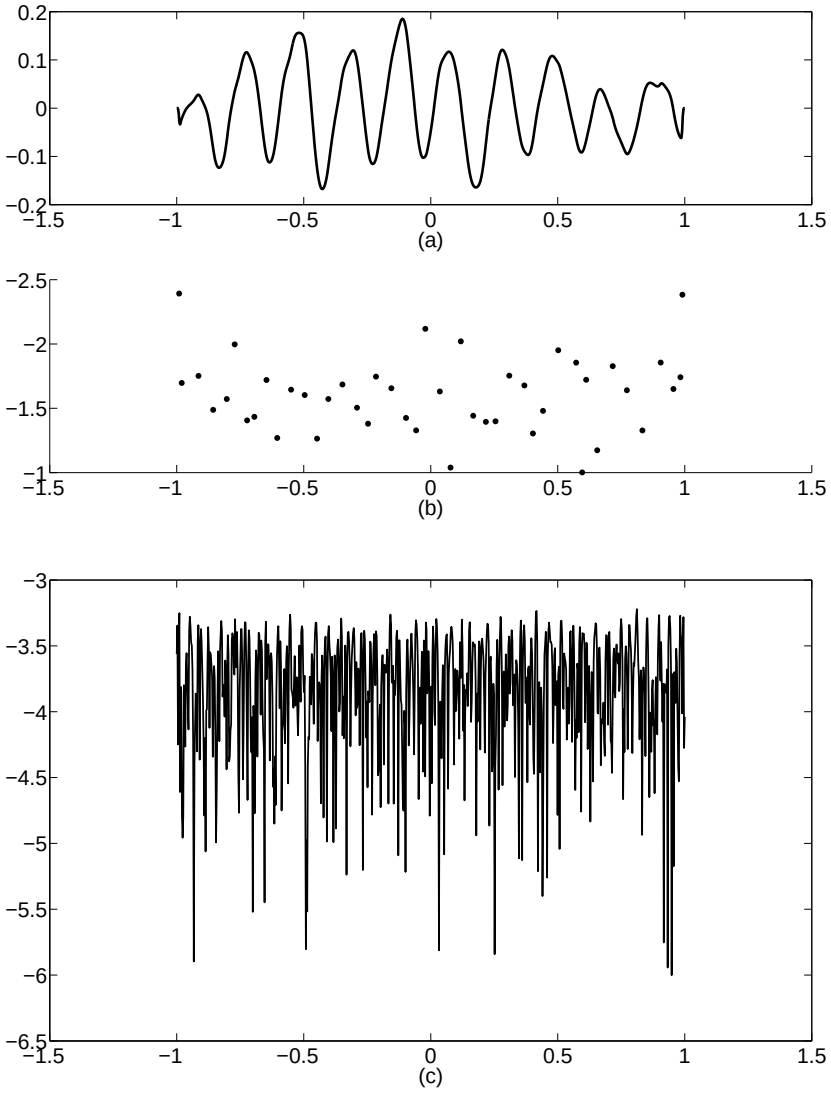


FIGURE 3. (a) Rational approximation of 768 sampled data points using 44 nodes. (b) Pole locations (displayed using a $\log_{10}$ scale for their imaginary part). (c) Associated error on a $\log_{10}$ scale.

number of poles required for the two segments. This shows that due to the compact support of the B-splines, and consequently the good localization of their rational approximations, the combined representations of the individual segments provide a near optimal representation for the combined segments. Furthermore, these results demonstrate that the step of merging adjacent representations may be considered optional.

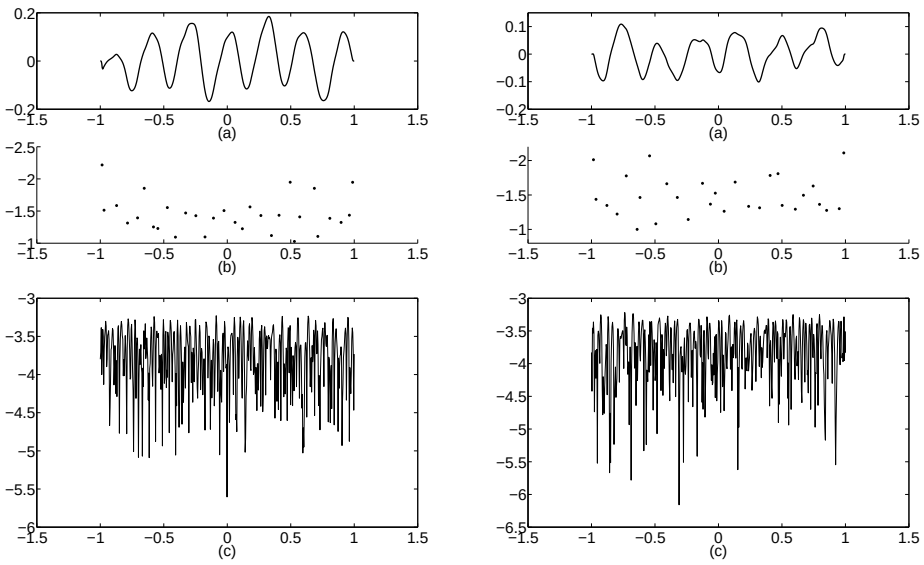


FIGURE 4. (a) Near optimal rational approximations on adjacent segments. (b) Pole locations (displayed using a $\log_{10}$ scale for their imaginary part). (c) Associated error on a $\log_{10}$ scale.

5.3. **Combined rational representations.** We now present an example of constructing a rational representation of a large signal. For this purpose we chose a portion of a piano recording containing $200,000$ samples. This recording was sampled at 44.1 kHz with 16 bits per sample. Segments of size $P = 6,250$ were used, yielding a rational representation with $18,373$ poles (the merging process was not applied to further reduce the number of poles). Counting only one pole and one weight in a conjugate pair, results in the total of 74948 real numbers to represent this signal, i.e., the compression factor ≈ 2.72 relative to the original number of samples. For a high quality compression of music, it is a good factor since no quantization or arithmetic coding has been used to increase the compression rate. The maximum absolute error is $4.83 \cdot 10^{-4}$ and there is no audible difference between the original signal and its reconstructed version. In Figure 6 we display the error of approximating the signal in this example. We note that, once the rational approximation is constructed, we can reconstruct the signal with an arbitrary sampling rate. This is a desirable property for a faithful reproduction of sound in loudspeakers.

6. FINAL REMARKS

We have developed, by combining several algorithms, a new approach to compute near optimal rational approximations for large data sets. The speed of these algorithms allows us to construct such approximations even for

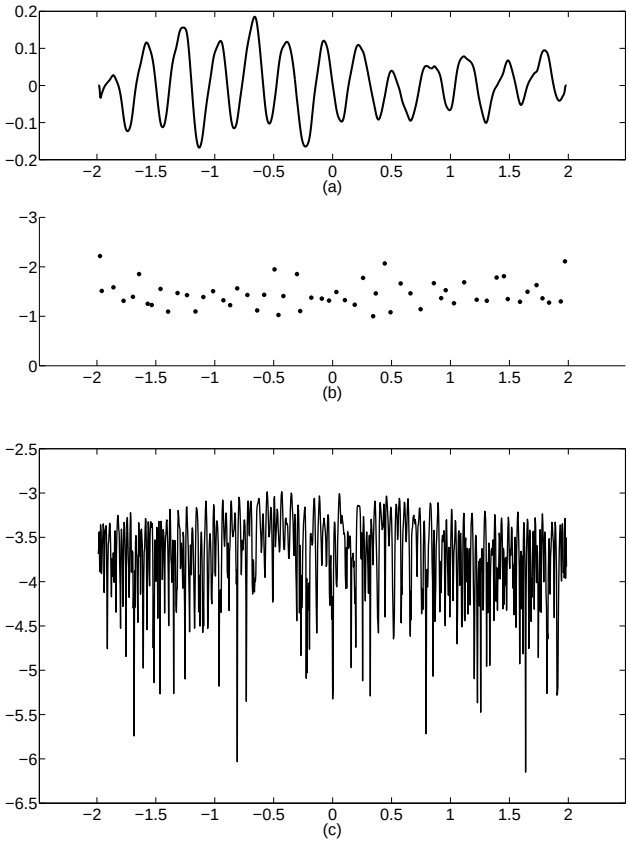


FIGURE 5. (a) Merged representation using 58 poles. (b) Pole locations (displayed using a $\log_{10}$ scale for their imaginary part). (c) Associated approximation error on a $\log_{10}$ scale.

signals that are generated by continuous monitoring, e.g., seismic monitoring. We observe that many parts of the algorithm can be trivially parallelized. In a modification of the approach, we can split the signal into sub-bands and construct rational approximations within each sub-band separately. This offers several advantages which we plan to address elsewhere.

The results also show promise for the development of a competitive algorithm for music compression. The building blocks of these representations contain information about local frequency content and are shift invariant. This property facilitates further processing of signals as the parameters are practically independent of the initial shift of the input data; this also opens up the possibility of recognizing recurring signal features at locations separated in time.

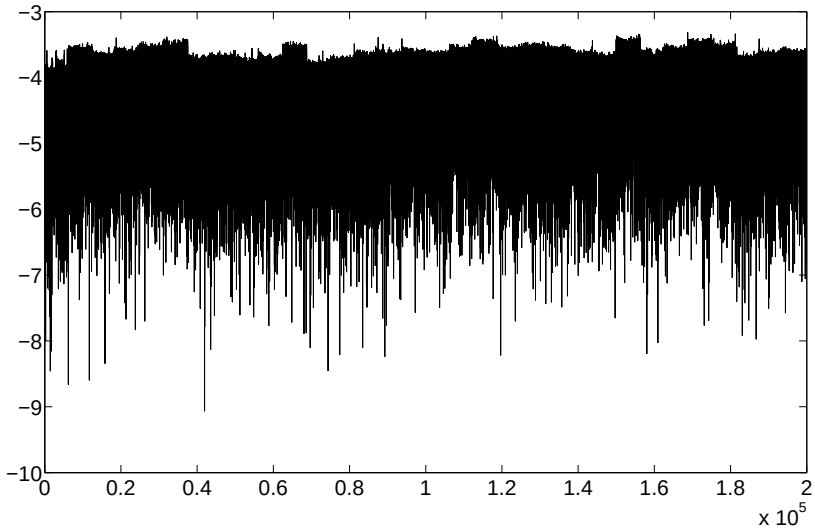


FIGURE 6. Error displayed on a $\log_{10}$ scale of signal reconstruction (a music recording) using a rational representation with $18,373$ poles. The original signal had $200,000$ samples.

REFERENCES

[1] V. M. Adamjan, D. Z. Arov, and M. G. Kreĭn. Infinite Hankel matrices and generalized Carathéodory-Fejér and I. Schur problems. *Funkcional. Anal. i Priložen.*, 2(4):1–17, 1968.

[2] V. M. Adamjan, D. Z. Arov, and M. G. Kreĭn. Infinite Hankel matrices and generalized problems of Carathéodory-Fejér and F. Riesz. *Funkcional. Anal. i Priložen.*, 2(1):1–19, 1968.

[3] V. M. Adamjan, D. Z. Arov, and M. G. Kreĭn. Analytic properties of the Schmidt pairs of a Hankel operator and the generalized Schur-Takagi problem. *Math. USSR Sbornik*, 15(1):34–75, 1971.

[4] G. Beylkin and R. Cramer. Toward multiresolution estimation and efficient representation of gravitational fields. *Celestial Mechanics and Dynamical Astronomy*, 84(1):87–104, 2002.

[5] G. Beylkin and L. Monzón. On approximation of functions by exponential sums. *Appl. Comput. Harmon. Anal.*, 19(1):17–48, 2005.

[6] G. Beylkin and L. Monzón. Nonlinear inversion of a band-limited Fourier transform. *Appl. Comput. Harmon. Anal.*, 27(3):351–366, 2009.

[7] G. Beylkin and L. Monzón. Approximation of functions by exponential sums revisited. *Appl. Comput. Harmon. Anal.*, 28(2):131–149, 2010.

[8] C. Chui. *An Introduction to Wavelets*. Academic Press, 1992.

[9] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21. http://cvxr.com/cvx, February 2011.

[10] T. S. Haut and G. Beylkin. Fast and accurate con-eigenvalue algorithm for optimal rational approximations. *arXiv:1012.3196 [math.NA]*, 2011. To appear in SIMAX.

[11] S. Jaffard, Y. Meyer, and R.D. Ryan. *Wavelets: tools for science & technology*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, revised edition, 2001.

[12] B. A. Jones, G. H. Born, and G. Beylkin. A Cubed Sphere Gravity Model for Fast Orbit Propagation. *AAS/AIAA Spaceflight Mechanics Meeting, Advances in the Astronautical Sciences*, 134:567–584, 2009.
[13] N. Yarvin and V. Rokhlin. An improved fast multipole algorithm for potential fields on the line. *SIAM J. Numer. Anal.*, 36(2):629–666 (electronic), 1999.

Department of Applied Mathematics, University of Colorado, Boulder, CO 80309-0526