SOLVING BURGERS' EQUATION USING OPTIMAL RATIONAL APPROXIMATIONS *

TERRY HAUT, GREGORY BEYLKIN AND LUCAS MONZÓN

ABSTRACT. We solve viscous Burger's equation using a fast and accurate algorithm— referred to here as the reduction algorithm— for computing near optimal rational approximations.

Given a proper rational function with n poles, the reduction algorithm computes (for a desired L^∞-approximation error) a rational approximation of the same form, but with a (near) optimally small number $m \ll n$ of poles. Although it is well-known that (nonlinear) optimal rational approximations are much more efficient than linear representations of functions via a fixed basis (e.g. wavelets), their use in numerical computations has been limited by a lack of efficient, robust, and accurate algorithms. The reduction algorithm presented here computes reliably (near) optimal rational approximations with high accuracy (e.g., $\approx 10^{-14}$) and a complexity that is essentially linear in the number n of original poles. A key tool is a recently developed algorithm for computing small con-eigenvalues of Cauchy matrices with high *relative* accuracy, an impossible task for standard algorithms without extended precision.

Using the reduction algorithm, we develop a numerical calculus for rational representations of functions. Indeed, while operations such as multiplication and convolution increase the number of poles in the representation, we use the reduction algorithm to maintain an optimally small number of poles.

To demonstrate the efficiency, robustness, and accuracy of our approach, we solve Burgers' equation with small viscosity ν. It is well known that its solutions exhibit moving transition regions of width $\mathcal{O}(\nu)$, so that this equation provides a stringent test for adaptive PDE solvers. We show that optimal rational approximations capture the solutions with high accuracy using a small number of poles. In particular, we solve the equation with local accuracy $\epsilon = 10^{-9}$ for viscosity as small as $\nu = 10^{-5}$.

1. INTRODUCTION

We solve viscous Burgers' equation using a fast and accurate algorithm for constructing rational approximations with (near) optimally small L^∞ error. When the viscosity ν is small, solutions of Burgers' equation develop sharp (moving) transition regions of width $\mathcal{O}(\nu)$, which presents a challenge for standard numerical methods. Although solving viscous Burgers' equation is primarily of academic interest, it allows us to demonstrate the efficiency, accuracy, and robustness of using optimal rational approximations for numerical computations. Our ultimate goal is to develop nonlinear

Key words and phrases. optimal rational approximations, high relative accuracy algorithms, con-eigenvalue problem, Burgers' equation, nonlinear approximations.

This research was partially supported by NSF grant DMS-100995 and DOE/ORNL grant 4000038129.

*Published in *Applied and Computational Harmonic Analysis, 34 (2013), pp. 83–-95.*

approximation methods for solving partial differential and integral equations in higher dimensions, where the ability to construct near optimal rational (or exponential) approximations to functions of one variable is a key component.

Since the seminal result in [21], it has been known that functions with singularities may be efficiently approximated in the L^∞ norm using proper rational functions. Indeed, the number of poles required to approximate a function with singularities is directly related to the sparsity of the function's wavelet coefficients (see [16, Theorem 11.1]). However, in contrast to more traditional L^2-type methods (using e.g., wavelet bases as in [2]), the use of such optimal L^∞-type approximations in numerical analysis has been limited due to a lack of efficient and robust algorithms.

Given a proper rational function f, we present an algorithm—which we refer to as the reduction algorithm—to compute, for a fixed number of poles, a rational approximation g to f with a (near) optimal L^∞ error. We use the reduction algorithm to develop a numerical calculus based on rational functions. Although operations such as multiplication and convolution increase the number of poles in the representation, we use the reduction algorithm afterwards to keep the number of poles optimally small for a specified accuracy. A salient feature of this approach is that optimal rational approximations efficiently represent functions with singularities or sharp transitions, and that positions of the poles are directly associated with the locations of singularities [5].

Our reduction algorithm relies on theory developed by Adamyan, Arov, and Krein [1] (referred below as AAK) for constructing optimal approximations in the L^∞-norm using meromorphic functions with a specified number of poles in the unit disk. In particular, let f denote a real valued (periodic) rational function with n pairs of complex poles γ_j, $1/\overline{\gamma_j}$ ($|\gamma_j| < 1$) and coefficients α_j, $\overline{\alpha_j}$. Then it turns out (see Appendix Section 4.1) that a (near) optimal rational approximation g, containing exactly m poles in the unit disk, may be obtained from the mth con-eigenvector u_m of the associated $n \times n$ Cauchy matrix $C_{ij} = \sqrt{\alpha_i}\sqrt{\overline{\alpha_j}}/\left(1 - \gamma_i\overline{\gamma_j}\right)$. Moreover, the approximation error satisfies $\|f - g\|_\infty \approx \lambda_m$, where λ_m is the mth con-eigenvalue of C, and the m poles of the approximation are roots of a rational function determined by the components of the con-eigenvector u_m. An analogous formulation also exists for obtaining (near) optimal approximations via decaying exponentials [4, 6], as well as rational functions defined on the real line. We formulate the con-eigenvalue problem in Section 2, and refer to [15, Section 4.6] for its general discussion. See also [22] for a clear discussion of the AAK theory.

Let us observe that in order to employ the reduction algorithm, two seemingly ill-advised numerical tasks must be performed — namely, accurately computing small con-eigenvalues (and con-eigenvectors) of Cauchy matrices, and computing all the roots in the unit disk of certain rational functions. One of the main points of this paper is to provide algorithms that solve both problems efficiently, reliably, and with high accuracy. A key tool in this regard is an algorithm developed in [**?**] to compute even the tiniest con-eigenvalues of positive-definite Cauchy matrices C with high *relative* accuracy, which is impossible using standard methods (see [13], [12], and [11] for the background on algorithms for achieving high relative accuracy). Also, of particular importance, is the robustness of the root-finding method, since it must be employed

many times. For example, in the context of solving Burgers' equation with viscosity $\nu = 10^{-5}$ and approximation tolerance $\epsilon = 10^{-9}$, on the order of a million applications of the reduction algorithm are performed.

For functions with n poles resulting from intermediate computations, the reduction algorithm requires only $\mathcal{O}(m^2 n)$ operations to find an optimal approximation with m poles. In our numerical experiments with the reduction algorithm, we find that an approximation error of $\epsilon \approx 10^{-14}$ may be reliably obtained within double precision arithmetic, even when the number of poles n is large and their spatial distribution is highly clustered.

There is a significant literature devoted to applications of the AAK approach in control theory (cf. [23]), signal processing (cf. [8]), and numerical analysis (cf. [25, 27, 29, 5]), to mention just a select few. The reformulation of the AAK theory given here could be related to the approaches taken in [28], [20], and [10]. However, as far as we know, all of the AAK-type algorithms discussed in the literature require $O(n^3)$ operations when applied to a rational function with n poles, and may require extended precision arithmetic if high accuracy of the result is desired. In contrast, our reduction algorithm requires only $\mathcal{O}(m^2 n)$ operations to find an optimal approximation with m poles and achieves high accuracy ($\epsilon \approx 10^{-14}$) using only double precision arithmetic.

We show in this paper that solutions of Burgers' equation with viscosity ν require only $\mathcal{O}(\log \nu^{-1}) + \mathcal{O}(\log \epsilon^{-1})$ poles for its rational approximation with an L^∞ error of size ϵ. Burgers' equation has been traditionally used to test the limits of new numerical methods since the solution develops sharp transition regions that need to be captured adaptively. Conceptually, the two closest adaptive methods are those in [24] and [2]. While in [2] adaptivity is achieved by adding wavelet scales when needed, the algorithm in [24] achieves spectral accuracy by adaptively positioning the necessary number of interpolating nodes within the transition region.

We compare the performance of our algorithms with that in [24], where authors use sub-optimal rational approximations based on conformally mapped Chebyshev grid points and barycentric interpolation. It appears that (for a comparable approximation error and viscosity) using optimal rational approximations to represent solutions of Burgers' equation results in significantly fewer poles. We also note that (as far as we know) our method successfully solves viscous Burgers' equation with the smallest viscosity reported in the literature, thus demonstrating the efficiency, accuracy and robustness of the reduction algorithm. Since standard methods for discretizing PDEs (e.g., collocation, projection, etc.) do not readily fit within the framework of our nonlinear numerical calculus, we also present a discretization scheme that may be of independent interest.

In Section 2, we describe the reduction algorithm and its connection to solving a coneigenvalue problem. In Section 3, we discuss the main algorithm for solving Burgers' equation, and present our numerical results.

2. Reduction algorithm for rational functions

In this section, we summarize the algorithm for obtaining a (near) optimal approximation of a periodic rational function by another periodic rational function with a

smaller number of poles. As mentioned earlier, our reduction algorithm is based on a theorem of Adamyan, Arov, and Krein ([1]), which concerns the approximation of a periodic function f, essentially bounded on the unit circle $\partial\mathbb{D}$, by a meromorphic function $r(z)$ ($z = e^{2\pi ix}$) containing a specified number of poles in the unit disk. We limit our presentation to rational functions f taking real values on $\partial\mathbb{D}$. This case turns out to be particularly important, as it allows us to develop a practical algorithm based on approximating the Fourier series coefficients of f with positive index. More general functions f may be dealt with by using the techniques in [5]. We note that the AAK theory may also be formulated for functions defined on the real line (cf. [22]).

2.1. **Overview of key algorithmic steps.** Following the same steps as in [4, Section 6] (see also Section 4.1), if the original function is rational, the (infinite) Hankel system derived from AAK theory may be reduced to a finite con-eigenproblem. Specifically, consider a rational function f of the form

$$f(z) = \sum_{i=1}^{n} \frac{\alpha_i}{z - \gamma_i} + \sum_{i=1}^{n} \frac{\overline{\alpha_i} z}{1 - \overline{\gamma_i} z} + f_0, \tag{2.1}$$

where f_0 is real, the residues α_j and poles γ_j are complex, and $0 < |\gamma_j| < 1$. Note that f is real-valued on the unit circle and that the Fourier series coefficients $\hat{f}_k$ of $f(e^{2\pi ix})$ satisfy

$$\hat{f}_k = \sum_{i=1}^{n} \alpha_i \gamma_i^{k-1}, \ k \geq 1,$$

with $\hat{f}_{-k} = \overline{\hat{f}_k}$ and $\hat{f}_0 = f_0$. We now describe an algorithm to find a rational approximation $g(e^{2\pi ix})$ to $f(e^{2\pi ix})$, of the same functional form (2.1), with a specified error in the L^∞-norm and a (near) optimal number of poles. Given a target accuracy ϵ, the steps for computing the rational approximant g are as follows:

Step 1: Compute a con-eigenvalue $\lambda_m \approx \epsilon$ and corresponding con-eigenvector u of the positive-definite Cauchy-like matrix C,

$$Cu = \lambda_m \overline{u}, \ \ u = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix}, \ \ C_{ij} = \frac{\alpha_i^{1/2} \overline{\alpha}_j^{1/2}}{1 - \gamma_i \overline{\gamma_j}}, \ \ i, j = 1, \ldots, n. \tag{2.2}$$

Here the con-eigenvalues $\lambda_0 \geq \lambda_1 \geq \ldots \lambda_{n-1} > 0$ are labeled in non-increasing order. In contrast to standard methods, our algorithm exploits the structure of C to compute its con-eigenvalues (and associated con-eigenvectors) with high *relative* accuracy, and in order $\mathcal{O}\left(n \log\left(\epsilon^{-1}\right)^2\right)$ operations (see Section 2.2).

Step 2: Find the m zeros η_j inside the unit disk of the proper rational function $v(z)$,

$$v(z) = \frac{1}{\lambda_m} \sum_{i=1}^{n} \frac{\sqrt{\overline{\alpha_i}}\, \overline{u_i}}{1 - \overline{\gamma_i} z}. \tag{2.3}$$

The fact that there are exactly m zeros in the unit disk, corresponding to the index m of the con-eigenvalue λ_m, is a consequence of the AAK theory. As shown in Section 4.1 (see equations (4.8)), the key to the high accuracy of evaluating the function $v(z)$ is the relationship

$$v(\gamma_i) = \overline{u_i}/\sqrt{\overline{\alpha_i}}, \quad i = 1, \ldots, n, \tag{2.4}$$

which, together with the n poles $1/\overline{\gamma_i}$, uniquely determines $v(z)$.

Step 3: Find the coefficients β_i of $g(z)$ by solving the $m \times m$ linear system,

$$\sum_{i=1}^{m} \frac{1}{1 - \eta_i \overline{\eta_j}} \beta_i = \sum_{i=1}^{n} \frac{\alpha_i}{1 - \gamma_i \overline{\eta_j}}, \quad j = 1, \ldots, m. \tag{2.5}$$

Denoting $\|f - g\|_\infty = \sup_{x \in [0,1]} |f(e^{-2\pi i x}) - g(e^{-2\pi i x})|$, the resulting rational approximation $g(e^{2\pi i x})$ satisfies $\|f - g\|_\infty \approx \epsilon$ and, thus, is close to the best L^∞-error achievable by rational functions with no more than m poles in the unit disk (see also [25] for a discussion of optimal rational approximations).

Remark 1. In Step 3, we solve for the coefficients β_i in $\mathcal{O}(m^2)$ operations by exploiting the structure of Cauchy matrices (see [11, 7]). We note that such a solver may require quadruple precision if the overall desired approximation error ϵ is smaller than $\approx 10^{-10}$. However, since $m = \log(\epsilon^{-1})$ is small, Step 3 for finding coefficients β_i does not impact the overall speed of the algorithm even if performed in quadruple precision.

Remark 2. In applications where the function $f(e^{2\pi i x})$ has singularities or sharp transitions, the poles γ_j in the rational representation of $f(e^{2\pi i x})$ may be located very close to the unit circle (and/or to each other). In such cases, it is advantageous to maintain the poles in the form $\gamma_j = \exp(-\tau_j)$, since they are well separated on a logarithmic scale. Importantly, the reduction algorithm computes the new poles $\eta_j = \exp(-\zeta_j)$ with nearly full precision in the exponents ζ_j, i.e., the ratio $\left|\hat{\zeta}_j - \zeta_j\right| / |\zeta_j|$ is close to machine precision even when $|\zeta_j| \ll 1$ (see [?]). However, to achieve high accuracy in the numerical examples of this paper, it was not necessary to maintain the poles in exponential form.

Remark 3. It may be shown (to be published elsewhere) that the con-eigenvalues λ_m of the positive-definite Cauchy matrix $C_{ij} = \alpha_i^{1/2} \overline{\alpha}_j^{1/2} / (1 - \gamma_i \overline{\gamma_j})$ in (2.2) satisfy the inequality

$$\max\{\lambda_{2m}, \lambda_{2m-1}\} \leq n^2 \frac{|\alpha_m|}{1 - |\gamma_m|^2} \Pi_{k=1}^{m-1} |f_{\gamma_k}(\gamma_m)|^2,$$

where f_{γ_k} denote the Moebius transformations

$$f_b(z) = \frac{z - b}{1 - \overline{b} z}$$

and the parameters α_m and γ_m are appropriately sorted. Since the transformation f_b maps the unit disk into itself if $|b| < 1$, the con-eigenvalues decay as $\lambda_m \sim r^m$

$(r < 1)$. This estimate shows that, for accuracy ϵ, we may reasonably expect $\mathcal{O}(\log \epsilon^{-1})$ terms in our approximation. In fact, we have observed this behavior in our numerical experiments.

Let us now briefly discuss the algorithmic aspects behind efficiency and accuracy of solving steps 1-3 above.

2.2. **Accurate computation of con-eigenvalues/eigenvectors.** For Step 1, we use a recent algorithm developed and analyzed in [?] for computing con-eigenvalues of Cauchy matrices with high relative accuracy, which we briefly describe in this section.

It is well-known that standard eigenvalue algorithms compute an approximate con-eigenvalue $\widehat{\lambda_m}$ with an error no better than $\left|\lambda_m - \widehat{\lambda_m}\right| / |\lambda_1| = \mathcal{O}(\delta)$, and an approximate unit con-eigenvector $\widehat{u}_m$ with an error no better than

$$\|u_m - \widehat{u}_m\|_2 = \mathcal{O}(\delta) / \text{absgap}_m, \quad \text{absgap}_m \equiv \min_{l \neq m} |\lambda_m - \lambda_l| / |\lambda_1|,$$

where δ denotes the machine round off. This implies that a computed con-eigenvalue smaller than $|\lambda_1|\,\delta$ will generally have few or no correct digits. Another undesirable feature of using standard con-eigenvalue methods to solve Step 1 is the $\mathcal{O}(n^3)$ complexity for finding the $m \ll n$ poles of $g(z)$, where n is the original number of poles of $f(z)$.

In contrast, the con-eigenvalue algorithm introduced in [?] computes even the smallest con-eigenvalues (and corresponding con-eigenvectors) accurately, i.e., the computed con-eigenvalue $\widehat{\lambda_m}$ satisfies $\left|\lambda_m - \widehat{\lambda_m}\right| / |\lambda_m| = \mathcal{O}(\delta)$, and the computed unit con-eigenvector $\widehat{u}_m$ satisfies

$$\|u_m - \widehat{u}_m\|_2 = \mathcal{O}(\delta) / \text{relgap}_m \quad \text{relgap}_m \equiv \min_{l \neq m} |\lambda_m - \lambda_l| / (\lambda_l + \lambda_m).$$

Thus, the computed con-eigenvalues and con-eigenvectors are accurate if the *relative* distance between the con-eigenvalues is not too small (which is the case for matrices considered here). Importantly, the mth con-eigenvalue (and con-eigenvector) is computed in $\mathcal{O}(m^2 n)$ operations. We note that, under mild assumptions, the con-eigenvalues of positive definite Cauchy matrices decay exponentially fast. It then follows that, for a given desired accuracy ϵ, $\|f(e^{2\pi i x}) - g(e^{2\pi i x})\|_\infty \approx \epsilon$, the number of poles m in the approximant $g(z)$ is $\mathcal{O}(\log \epsilon^{-1})$. Therefore, the complexity of our algorithm is $\mathcal{O}\left((\log \epsilon^{-1})^2 n\right)$, i.e., is essentially linear in the number of original poles n and, thus, its speed is controlled by the number of poles of the final optimal approximation. Moreover, in contrast to the usual perturbation theory for general matrices, small perturbations of the poles γ_m and residues α_m (determining the Cauchy matrix $C = C(\alpha, \gamma)$) lead to correspondingly small perturbations in the con-eigenvalues and con-eigenvectors [?].

2.3. **Finding poles for near optimal approximation.** There are two numerical difficulties associated with the root-finding algorithm in Step 2 of Section 2.1. First, the roots of polynomial or rational functions may be notoriously ill-conditioned with respect to their defining parameters. In particular, using the explicit formula (2.3) to compute values of $v(z)$ typically results in a loss of roughly $\log_{10}(\lambda_m^{-1})$ digits. Indeed,

using (2.4) to rewrite (2.3) as

$$\sum_{i=1}^{n} \frac{\overline{\alpha_i}\, v(\gamma_i)}{1-\overline{\gamma_i} z} = \lambda_m v(z),$$

we see that the sum must suffer cancellation of about $\log_{10}(\lambda_m^{-1})$ digits if $v(\gamma_i)$ and $v(z)$ are of comparable size (note that λ_m controls the approximation error and, thus, is necessarily small).

The second difficulty associated with the root-finding step is that root-finding methods based on standard iterative procedures such as Newton's method are often too sensitive to the initial guess and, for that reason, may not locate all the roots reliably. Our PDE solver (see Section 3) requires roughly a million applications of the reduction algorithm and, thus, it is imperative that the root-finding algorithm is both efficient and reliable in locating all m roots of $v(z)$ (recall that the index m of the con-eigenvalue λ_m corresponds to the number of roots in the unit disk). Indeed, due to optimality of the rational approximation, missing even one root leads to an unacceptably large error in the corresponding approximation.

The root-finding algorithm presented below makes use of two key observations. First, the values $v(\gamma_i) = \overline{u_i}/\sqrt{\overline{\alpha_i}}$ of $v(z)$ may be computed in Step 1 with high accuracy from the con-eigenvalue components u_i. Noticing that the n values $v(\gamma_i)$ and poles $\overline{\gamma_i}^{-1}$ uniquely determine $v(z)$, we compute $v(z)$ via rational interpolation with the values $v(\gamma_i)$ and poles $\overline{\gamma_i}^{-1}$ rather than using formula (2.3). Heuristically, the reason this approach works well is that the roots of $v(z)$ are typically close to the poles γ_i (since the roots yield the poles of a near optimal approximation), and it is natural to expect that having many accurate values $v(\gamma_i)$ of $v(z)$ close to the roots allows us to compute them with high accuracy. The second key observation is that the roots of $v(z)$ coincide with the eigenvalues of a rank-one-plus-diagonal matrix, and this matrix may be applied (along with its shifted inverse) in $\mathcal{O}(n)$ operations. This yields an efficient and robust way to locate all roots of $v(z)$ within the unit disk.

The basic strategy behind the root-finding algorithm is as follows. First, we use Newton's method on the rational interpolant computed from the values $v(\gamma_i)$ and poles $\overline{\gamma_i}^{-1}$. Since we have good initial guesses for Newton's method, this procedure typically locates most of the roots of $v(z)$. To compute any roots that Newton's method misses (recall that we know from Step 1 the total number m of roots in the unit disk), we use an efficient version of shifted inverse iteration on the diagonal-plus-rank-one matrix whose eigenvalues coincide with the roots of $v(z)$. Because the eigenvalues of this matrix are often ill-conditioned, some of these eigenvalues may be only evaluated with a few accurate digits. However, using Newton's method on the rational interpolant allows us to refine the missing roots to nearly full precision.

Let us now describe this algorithm in greater detail. As noted above, $v(z)$ is uniquely determined from its n values $v(\gamma_i) = u_i/\sqrt{\alpha_i}$, accurately computed from Step 1, and its n poles $1/\overline{\gamma_i}$. This allows us to compute an approximation $\widetilde{v}(z)$ to $v(z)$ via continued fractions,

$$\widetilde{v}(z) = \frac{a_1}{1 + a_2 (z-\gamma_1) / (1 + a_3 (z-\gamma_2) / (1 + \cdots))}, \tag{2.6}$$

where the coefficients a_j are determined from the interpolation conditions $\widetilde{v}(\gamma_i) = v(\gamma_i)$, and may, in general, be computed in $\mathcal{O}(n^2)$ operations. Importantly, the poles γ_i are often clustered "around" the roots of $v(z)$ (this is the case in our PDE application), and it is sufficient to use local rational interpolation within a given cluster to find roots. This reduces the complexity to essentially $\mathcal{O}(m)$ operations, where m is the number of roots in the unit disk. Once the coefficients a_j are determined, the values of $\widetilde{v}(z)$ and $\widetilde{v}'(z)$ may be computed in $\mathcal{O}(n)$ operations using recursion formulas [9] (the complexity reduces to $\mathcal{O}(m)$ if it is done locally as described above). As indicated previously, this method yields very accurate results when the poles γ_i are highly clustered (which is the case in our PDE application). Indeed, the roots of $v(z)$ coincide with the poles of a (near) optimal rational approximation, so that a given root is often located close to some particular cluster $\gamma_{i_1}, \gamma_{i_2}, \ldots, \gamma_{i_k}$ of original poles. Since Step 1 computes the values $v(\gamma_{i_k})$ of $v(z)$ with high accuracy, such pole clustering actually contributes to a high degree of numerical stability. As a technical point, computing the coefficients a_j in 2.6 requires arranging the nodes $\gamma_1, \ldots, \gamma_n$ in increasing order of magnitude in order to achieve high accuracy.

We also note that, as an alternative to using continued fractions, the roots of $v(z)$ may also be accurately computed using Lagrange interpolation (and the known poles $\overline{\gamma_i}^{-1}$ of $v(z)$),

$$v(z) = \frac{\Pi_{i=1}^n (z-\gamma_i)}{\Pi_{i=1}^n (1-z\overline{\gamma_i})} \sum_{j=1}^n \frac{s_j}{(z-\gamma_j)}, \quad s_j = \frac{\Pi_i (1-\gamma_j\overline{\gamma_i})}{\Pi_{i\neq j}(\gamma_j-\gamma_i)} v(\gamma_j). \tag{2.7}$$

Computing the barycentric weights s_j, in general, requires $\mathcal{O}(n^2)$ operations, and evaluation of $v(z)$ and $v'(z)$ (once the weights s_j are computed) requires $\mathcal{O}(n)$ operations. Constructing rational interpolants from appropriately grouped pole clusters γ_i again allows us to reduce the complexity to $\mathcal{O}(m)$ operations. We note that computing the coefficients s_j requires in this case arranging the nodes $\gamma_1, \ldots, \gamma_n$ in decreasing order of magnitude in order to achieve high accuracy.

As mentioned previously, we compute roots that the above procedure misses by using the fact that the roots of (2.3) coincide with the eigenvalues of the diagonal-plus-rank-one matrix (cf. [26] and [19]),

$$A = D + ab^{\mathrm{T}}, \tag{2.8}$$

where the diagonal matrix D and the vectors a and b satisfy

$$D_{ii} = \overline{\gamma_i}^{-1}, \quad a_i = \frac{\overline{\gamma_i}^{-1}\sqrt{\alpha_i}u_i}{\sum_{j=1}^n \overline{\gamma_j}^{-1}\sqrt{\alpha_j}u_j}, \quad b_i = \overline{\gamma_i}^{-1}, \quad i = 1, \ldots, n.$$

Using the Sherman-Morrison formula, the matrix $(A-\lambda I)^{-1}$ may be efficiently applied in $\mathcal{O}(n)$ operations and, therefore, simultaneous inverse iteration may be used to compute all m eigenvalues of A inside the unit disk (and, hence, all m roots of $v(z)$ in the unit disk). To illustrate this procedure, assume that $m-1$ roots $\beta_1, \ldots, \beta_{m-1}$ have been found using the version of Newton's method described above, and we would like to compute the missing root β_m. To do so, we first use the Sherman-Morrison formula, combined with one step of inverse iteration, to compute eigenvectors $q_1, \ldots, q_{m-1}$ of A corresponding

to the known eigenvalues $\beta_1, \ldots, \beta_{m-1}$, one by one. We then orthogonalize these $m-1$ vectors using the stabilized Gram-Schmidt procedure, thus yielding a basis $\widehat{q}_1, \ldots, \widehat{q}_{m-1}$ for the invariant subspace $\operatorname{span}\{q_1, \ldots, q_{m-1}\} = \operatorname{span}\{\widehat{q}_1, \ldots, \widehat{q}_{m-1}\}$. Finally, we use simultaneous inverse iteration applied to $\widehat{q}_1, \ldots, \widehat{q}_{m-1}, q$, where q is chosen randomly. Notice that each step of this process requires orthogonalizing $q^{(k+1)} = \left(A - \lambda^{(k)} I\right)^{-1} q^{(k)}$ against $\widehat{q}_1, \ldots, \widehat{q}_{m-1}$, where $\lambda^{(k)}$ is the guess for β_m after k steps. The matrix-vector product $\left(A - \lambda^{(k)} I\right)^{-1} q^{(k)}$ may be computed in $\mathcal{O}(n)$ operations from the Sherman-Morrison formula. Therefore, each step of this iterative process requires $\mathcal{O}(mn)$ operations, and an initial $\mathcal{O}(m^2 n)$ operations to orthogonalize $q_1, \ldots, q_{m-1}$.

Remark 4. In applications where the poles γ_i are not clustered, we have observed that the roots of $v(z)$ are computed with nearly full precision using Lagrange interpolation (2.7). In contrast, using continued fractions as in (2.6) may not always yield accurate roots if the poles are not clustered.

Remark 5. In both Newton's method and the inverse iteration method, we used the original poles, γ_i, as starting guesses. However, the starting guess does not play a significant role in inverse iteration since it is globally convergent.

3. Solving (1+1) dimensional nonlinear partial differential equations using optimal rational approximations

We now describe a method for solving Burgers' equation,

$$u_t - uu_x = \nu u_{xx}, \quad u(x,0) = u_0(x), \quad u(0,t) = u(1,t), \tag{3.1}$$

using the reduction algorithm of Section 2. We demonstrate that using optimal rational approximations allows us to compute solutions that are accurate over a very large range in Fourier space and, thus, resolves the spatial singularities with high accuracy.

The main idea of our time-stepping scheme is to represent the solution in space as a proper rational function. The discretization of (3.1) requires only a few basic operations on such rational functions, and preserves their rational form. These operations naturally increase the number of poles in the representation and, thus, we employ the reduction algorithm at each stage of the process to keep the number of poles as small as possible. Our results show that the solution of (3.1) may be obtained using rational functions with a small number of poles and with a uniform error, even within the rapid transition region developed in the process of evolution.

We first describe how, starting from $u(x,0) = u_0(x)$, we compute $u(x,t)$ for a given timestep t. By recasting (3.1) in semigroup form (see Section 4.2), an appropriate temporal and spatial discretization of (3.1) leads to the nonlinear system of equations,

$$u_l(x) = \sum_{p=1}^{M_x} \lambda_p^l u_0\left(x - \phi_p^l\right) + \sum_{j=1}^{M_t} \sum_{p=1}^{M_x} \lambda_{p,j}^l u_j^2\left(x - \psi_p^l\right), \tag{3.2}$$

where $u_l(x) \approx u(x, \tau_l)$, $1 \le l \le M_t$, and $\{\tau_l\}$ are the M_t Gauss-Legendre quadrature nodes on the time interval $(0,t)$. The real-valued quantities ϕ_p^l, ψ_p^l, λ_p^l, $\lambda_{p,j}^l$ in (3.2)

depend on the timestep t, the number M_t of quadrature nodes in time, and the number of quadrature nodes M_x used in space to discretize the convolution kernels. From the rapid decay of the periodic heat kernel,

$$K_\nu(x,t) = \frac{1}{\sqrt{4\pi\nu t}} \sum_{k\in\mathbb{Z}} e^{-(x+k)^2/(4\nu t)},$$

where ν is the viscosity parameter in (3.1), it follows that ϕ_p^l and ψ_p^l are localized to a $\mathcal{O}\left(\sqrt{\nu t}\right)$ neighborhood of $x = 0$ (see Section 4.2 for details).

We assume that the initial function $u(x,0) = u_0(x)$ is given as a periodic rational function of the form

$$u_0(x) = \sum_{j=1}^{M_0} \frac{\alpha_j}{e^{-2\pi i x} - \gamma_j} + \sum_{i=1}^{M_0} \frac{\overline{\alpha_j}}{e^{2\pi i x} - \overline{\gamma_j}} + \alpha_0,$$

and that this representation is nearly optimal. We then solve the system of equations (3.2) by approximating each function u_l using the reduction algorithm. We obtain, via fixed point iteration applied to (3.2) and the reduction algorithm, rational functions $u_l(x)$ of the form,

$$u_l(x) = \sum_{j=1}^{M_l} \frac{\alpha_{j,l}}{e^{-2\pi i x} - \gamma_{j,l}} + \sum_{j=1}^{M_l} \frac{\overline{\alpha_{j,l}}}{e^{2\pi i x} - \overline{\gamma_{j,l}}} + \alpha_0, \tag{3.3}$$

which solve (3.2) to a specified level of precision, and have a (near) optimal number of poles.

More specifically, given $u_j^{(m)} \approx u_j(x)$, $1 \le j \le M_t$, at iteration m, we use (3.2) to define the next iterates $u_l^{(m+1)}(x)$ for $l = 1, \ldots, M_t$,

$$\begin{aligned} u_l^{(m+1)}(x) &= \sum_{p=1}^{M_x} \lambda_p^l u_0\left(x - \phi_p^l\right) \\ &+ \sum_{p=1}^{M_x} \lambda_{p,j}^l \sum_{j=1}^{l-l} \left\{ \left(u_j^{(m+1)}\left(x - \psi_p^l\right)\right)^2 + \sum_{j=l+1}^{M_t} \lambda_{p,j}^l \left(u_j^{(m)}\left(x - \psi_p^l\right)\right)^2 \right\}. \end{aligned} \tag{3.4}$$

Note that, in computing $u_l^{(m+1)}(x)$ for $l > 1$, we use the functions $u_j^{(m+1)}(x)$, $1 \le j < l$ already available to us. We take $u_j^{(1)}(x) = u_0(x)$, $1 \le j \le M_t$, as an initial guess for $u_j(x)$.

Although this initial form for $u_l^{(m+1)}(x)$ is also rational, it is not of the form (3.3), since it contains poles of multiplicity two. However, it follows from equation (3.4) and the distribution of the parameters ϕ_p^l and ψ_p^l in (4.12), that the poles of $u_l^{(m+1)}(x)$ are tightly clustered in $\mathcal{O}\left(\sqrt{\nu t}\right)$ neighborhoods about the poles γ_m of the initial function $u_0(x)$. We may therefore obtain a very accurate sub-optimal representation of $u_l^{(m+1)}(x)$ of the required form (3.3) by computing $(q, q+1)$ Pade approximants of the rational functions in (3.4) associated with each cluster of poles, where the Pade expansions are

centered about $1/\overline{\gamma_m}$. In our numerical experiments, $(q, q+1)$ Pade approximations of order $1 \leq q \leq 4$ typically yield an L^∞ error smaller than 10^{-14}. Note that obtaining a proper rational approximation of $u_l^{(m+1)}(x)$ in this manner requires solving M_0 small (e.g., 3×3) linear systems, and yields a sub-optimal approximation with about three times the optimal number of poles. We then use the reduction algorithm, outlined in Section 2, to obtain an optimal rational representation of $u_l^{(m+1)}(x)$. This process is repeated until the desired level of precision is obtained.

3.1. **Examples.** As a first example, we solve equation (3.1) with viscosity $\nu = 10^{-3}$, and with initial condition $u_0(x) = \sin(2\pi x)$. For the time discretization, we use a timestep equal to 10^{-3} and $M_t = 3$ quadrature nodes τ_l in $(0, t)$ (see equation (3.2)). This yields a local error of less than 10^{-11}. For the spatial part, we apply the reduction algorithm by selecting the smallest con-eigenvalue value greater than $\epsilon = 10^{-12}$, which ensures a uniform L^∞-error of about 10^{-12}. In our application of Pade approximation, we obtain a spatial error in the L^∞-norm no larger than 10^{-11}.

We take 400 timesteps, which ensures that we evolve (3.1) past the point at which the solution begins to decrease. To assess the error, we independently obtain the solution to (3.1) by using the Hopf-Cole transformation to reduce Burger's equation to the heat equation. We then solve the heat equation in extended precision arithmetic (the Hopf-Cole transformation is highly ill-conditioned) to obtain a solution that we use as a gauge for assessing accuracy. We verify that the L^∞-norm of the difference between the two solutions remains less than 1.6×10^{-9}.

Figure 3.1 shows the computed solutions $u(x, t)$, which have 5, 9, 14, and 13 complex-conjugate pairs of poles at times $t = 0.02$, $t = 0.11$, $t = 0.21$, and $t = 0.41$. We also show the error of the computed solution at times $t = 0.11$, $t = 0.21$, and $t = 0.41$.

As a second example, we solve Burgers' equation (3.1) with viscosity $\nu = 10^{-5}$ and the initial condition $u_0(x) = \sin(2\pi x) + 1/2 \sin(4\pi x)$. In our temporal discretization, we used a timestep equal to 10^{-5} and $M_t = 3$ quadrature nodes. For the spatial part, we apply the reduction algorithm with an approximation error of $\epsilon = 10^{-9}$, which ensures a uniform L^∞-error of $\approx 10^{-9}$. In our application of Pade approximation, we obtain a spatial error in the L^∞-norm no larger than 10^{-11}. Although we were unable to independently verify the accuracy of the computed solutions for such a small viscosity ν (for the lack of alternative methods of reasonable complexity), we note that the iteration scheme in (3.4) converged (in the L^∞-norm) to within an error no larger than 7.5×10^{-9} at every timestep.

Figure 3.2 shows the computed solutions $u(x, t_0 j)$, with $t_0 = 10^{-5}$ and time steps t_j, $j = 10^2$, 10^4, 2×10^4, 3×10^4, 5×10^4. We see that the solution $u(x, t)$ develops two moving sharp transition regions, which approach each other and eventually merge into a single one about $x \approx 1/2$. The rational representations of $u(x, t_j)$ have 4, 11, 33, 29, and 19 complex-conjugate pairs of poles, respectively. Figure 3.3 demonstrates that the transition region of $u(x, t)$ occur within intervals of width $\approx 10^{-5}$. Finally, Figure 3.4 illustrates the poles $\gamma_i(t)$ in the representation,

$$u(x,t) = \sum_{j=1}^{M_0} \frac{\alpha_j(t)}{e^{-2\pi i x} - \gamma_j(t)} + \sum_{i=1}^{M_0} \frac{\overline{\alpha_j(t)}}{e^{2\pi i x} - \overline{\gamma_j(t)}} + \alpha_0,$$

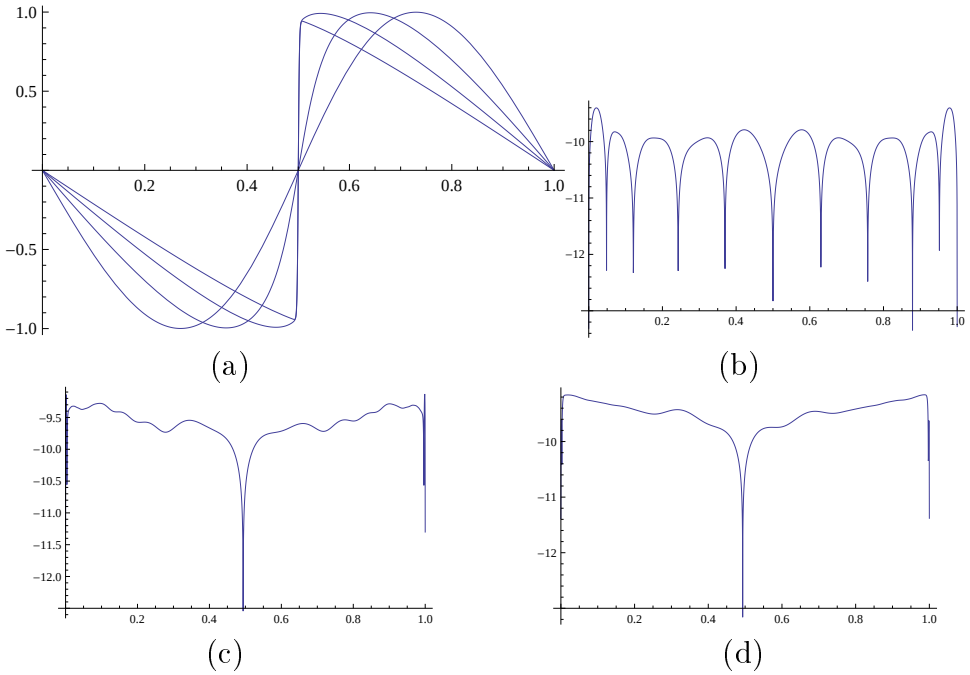

FIGURE 3.1. (a) Computed solution $u(x,t)$ at $t = 0.02$, $t = 0.11$, $t = 0.21$, $t = 0.41$ and its absolute error (on a logarithmic scale) for (b) $t = 0.11$, (c) $t = 0.21$, and (d) $t = 0.41$.

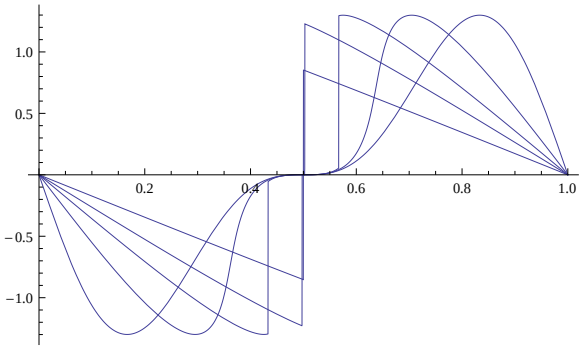

FIGURE 3.2. Plots of $u(x,t)$, for $t_j = 10^{-3}$, .1, .2, .3, and .5.

for $t = .2, .274, .3, .4$. As expected, the poles cluster about transition regions, and move (adaptively) as the two wavefronts approach each other.

4. Appendix

4.1. **Review of AAK theory.** In order to formulate the basic AAK theorem on the unit disk, let us denote by H^∞ the Hardy space of bounded analytic functions and by H^∞_N the set of functions

$$H^\infty_N = \left\{ \frac{g(z)}{(z-\eta_1)\cdots(z-\eta_k)},\ |\eta_j| < 1,\ k \le N,\ and\ g \in H^\infty \right\}.$$

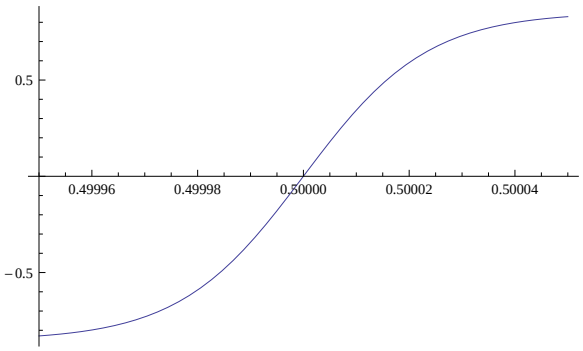

FIGURE 3.3. Solution $u(x,t)$ at time $t = .4$, localized about the transition region $(1/2 - 10^{-5}, 1/2 + 10^{-5})$. Note the absence of any Gibbs-type phenomena.

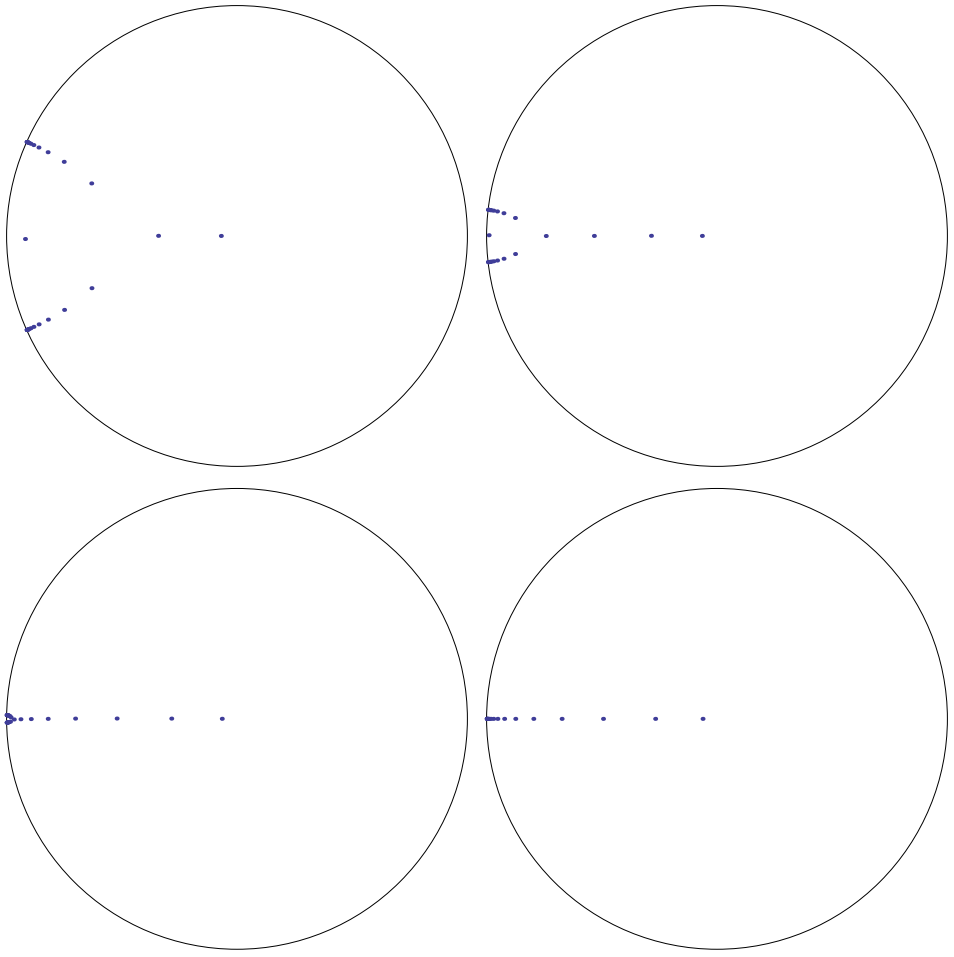

FIGURE 3.4. Location of poles (within the unit disk) in the representation of $u(x,t)$, for $t = .2$, $.275$, $.3$, and $.4$.

Suppose $f \in L^\infty$ has the Fourier series

$$f(z) = \sum_{n=-\infty}^{\infty} f_n z^{-n},$$

and consider the associated infinite Hankel matrix H_f

$$H_f = \begin{pmatrix} f_1 & f_2 & f_3 & \cdots \\ f_2 & f_3 & f_4 & \cdots \\ f_3 & f_4 & f_5 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

with singular values σ_n considered in decreasing order. From the singular value problem for the N^{th} singular value

$$\begin{aligned} H_f v &= \sigma_N w, \\ H_f^* w &= \sigma_N v, \end{aligned} \tag{4.1}$$

where $v = (v_j)_{j\geq 1}$ and $w = (w_j)_{j\geq 1}$, we define the functions

$$v(z) = \sum_{j=1}^{\infty} v_j z^{j-1}, \quad w(z) = \sum_{j=1}^{\infty} w_j z^{-j},$$

and

$$r(z) = f(z) - \sigma_N \frac{w(z)}{v(z)}. \tag{4.2}$$

For this particular case, the AAK theorem asserts that $r \in H_N^\infty$ and

$$\|f - r\|_\infty = \inf_{g \in H_N^\infty} \|f - g\|_\infty = \sigma_N.$$

An important special case is when $f(z)$ has the form (2.1), that is,

$$f(z) = \sum_{m=1}^{M} \frac{\alpha_m z^{-1}}{1 - \gamma_m z^{-1}} + \sum_{m=1}^{M} \frac{\overline{\alpha_m} z}{1 - \overline{\gamma_m} z} + f_0, \tag{4.3}$$

where α_m and γ_m are complex and $0 < |\gamma_m| < 1$. We now show that the infinite singular value problem (4.1) may be reduced to the finite con-eigenvalue problem (2.2).

First, note that equation (4.1) may be written as

$$\sum_{j=1}^{\infty} f_{i+j-1} v_j = \sigma w_i, \quad i = 1, 2, \ldots \tag{4.4}$$

$$\sum_{j=1}^{\infty} \overline{f_{i+j-1}} w_j = \sigma v_i, \quad i = 1, 2, \ldots \tag{4.5}$$

Using that the Fourier coefficients of (4.3) are of the form

$$f_n = \sum_{m=1}^{M} \alpha_m \gamma_m^{n-1}, \quad n \geq 1,$$

we calculate from (4.4)

$$\begin{aligned}\sum_{j=1}^{\infty}\left(\sum_{m=1}^{M}\alpha_m\gamma_m^{i+j-2}\right)v_j &= \sum_{m=1}^{M}\alpha_m\gamma_m^{i-1}\sum_{j=1}^{\infty}\gamma_m^{j-1}v_j \\ &= \sum_{m=1}^{M}\alpha_m\gamma_m^{i-1}v\left(\gamma_m\right)=\sigma w_i.\end{aligned}$$

Now multiplying both sides of the last equation by z^{i-1} and summing, we obtain

$$\sum_{m=1}^{M}\frac{\alpha_m}{1-\gamma_m z}v\left(\gamma_m\right)=\sigma z^{-1}w(z^{-1}). \tag{4.6}$$

Similarly, from (4.5), we have

$$\begin{aligned}\sum_{j=1}^{\infty}\left(\sum_{m=1}^{M}\overline{\alpha_m\gamma_m}^{i+j-2}\right)w_j &= \sum_{m=1}^{M}\overline{\alpha_m\gamma_m}^{i-1}\sum_{j=1}^{\infty}\overline{\gamma_m}^{j-1}w_j \\ &= \sum_{m=1}^{M}\overline{\alpha_m\gamma_m}^{i-1}\left(\overline{\gamma_m}^{-1}w\left(\overline{\gamma_m}^{-1}\right)\right)=\sigma v_i.\end{aligned}$$

Finally, multiplying by z^{i-1} and summing, we arrive at

$$\sum_{m=1}^{M}\frac{\overline{\alpha_m}}{1-\overline{\gamma_m}z}\overline{\gamma_m}^{-1}w\left(\overline{\gamma_m}^{-1}\right)=\sigma v(z). \tag{4.7}$$

Hence, for a function f of the form (4.3), the functions v and w in (4.2) turn out to be rational and fully determined by their values at the poles of f. Taking $z=\overline{\gamma_n}$ and $z=\gamma_n$ in equations (4.6) and (4.7), respectively, we obtain

$$\begin{aligned}\sum_{m=1}^{M}\frac{\alpha_m}{1-\gamma_m\overline{\gamma_n}}v\left(\gamma_m\right) &= \sigma\overline{\gamma_m}^{-1}w\left(\overline{\gamma_m}^{-1}\right), \\ \sum_{m=1}^{M}\frac{\overline{\alpha_m}}{1-\overline{\gamma_m}\gamma_n}\overline{\gamma_m}^{-1}w\left(\overline{\gamma_m}^{-1}\right) &= \sigma v\left(\gamma_n\right). \end{aligned}\tag{4.8}$$

We symmetrize the above equations by multiplying the first equation by $\overline{\alpha_n}^{1/2}$ and the second equation by $\alpha_n^{1/2}$ to get

$$\begin{aligned}\sum_{m=1}^{M}\frac{\alpha_m^{\frac{1}{2}}\overline{\alpha_n}^{\frac{1}{2}}}{1-\gamma_m\overline{\gamma_n}}\alpha_m^{\frac{1}{2}}v\left(\gamma_m\right) &= \sigma\overline{\alpha_n}^{1/2}\overline{\gamma_m}^{-1}w\left(\overline{\gamma_m}^{-1}\right), \\ \sum_{m=1}^{M}\frac{\overline{\alpha_m}^{\frac{1}{2}}\alpha_n^{\frac{1}{2}}}{1-\overline{\gamma_m}\gamma_n}\overline{\alpha_m}^{1/2}\overline{\gamma_m}^{-1}w\left(\overline{\gamma_m}^{-1}\right) &= \sigma\alpha_m^{\frac{1}{2}}v\left(\gamma_m\right).\end{aligned}$$

Let us define the vectors p and q with entries $p_m = \alpha_m^{\frac{1}{2}} v(\gamma_m)$, $q_m = \overline{\alpha_n}^{1/2}\overline{\gamma_m}^{-1} w(\overline{\gamma_m}^{-1})$, and the positive definite matrix C with entries

$$C_{mn} = \frac{\alpha_m^{\frac{1}{2}}\overline{\alpha_n}^{\frac{1}{2}}}{1-\gamma_m\overline{\gamma_n}}.$$

Then the above equations are equivalent to

$$\begin{aligned} C\,p &= \sigma q, \\ C\,\overline{q} &= \sigma\overline{p}, \end{aligned}$$

which may be reduced to a con-eigenvalue problem for $\sigma > 0$, see [15, Section 4.6]. One simple way to see this and obtain an equation of the form (2.2) is by defining $x = p + \overline{q}$. If $x = 0$, then $iq = \overline{ip}$ and hence

$$C(ip) = \sigma\overline{ip}.$$

If $x \neq 0$, we have

$$Cx = \sigma\overline{x}$$

and, in both cases, we obtain a con-eigenvalue problem for the matrix C.

4.2. **Discretization of Burgers' equation.** We rewrite the equation (3.1) in semi-group form (see, e.g., [14, 17, 18, 3])

$$u(t) = e^{\nu t L}u(0) + \int_0^t e^{\nu(t-\tau)L} N(u(\tau))d\tau, \tag{4.9}$$

where $u(t)$ denotes the function $u(\cdot, t)$. The operator L, $Lu(x) = u_{xx}$, represents the linear part of (3.1) while the operator N, $N(u) = 1/2\,(u^2)_x$, represents the nonlinear part. The action of the operator $e^{\nu t L}$ on a function f is given by

$$\left(e^{\nu t L} f\right)(x) = \int_{-\frac{1}{2}}^{\frac{1}{2}} K_\nu(y,t) f(x-y)dy, \text{ with } K_\nu(y,t) = \frac{1}{\sqrt{4\pi\nu t}} \sum_{k\in\mathbb{Z}} e^{-(y+k)^2/(4\nu t)}.$$

To discretize equation (4.9) in time, we use the approximation

$$N(u(\tau)) \approx \sum_{j=1}^{M_t} R_j(\tau) N\left(u\left(\tau_j\right)\right), \ \tau \in [0,t]$$

where $\{\tau_j\}_{j=1}^{M_t}$ denote the Gauss-Legendre nodes on the interval $(0,t)$, and $R_j(\tau)$ denote the Legendre interpolating polynomials for these nodes, i.e.,

$$R_j(\tau_m) = \delta_{jm}, \quad \text{for } j,m = 1,\ldots,M_t.$$

Taking $t = \tau_l$ in (4.9), we obtain the semi-discrete system of equations

$$u_l = e^{\nu\tau_l L}u_0 + \sum_{j=1}^{M_t}\left(\int_0^{\tau_l} e^{\nu(\tau_l-\tau)L} R_j(\tau)d\tau\right) N(u_j), \quad 1 \le l \le M_t, \tag{4.10}$$

where $u_l = u_l(x)$ denote the computed values of u at time $t = \tau_l$ and $u_0 = u(x,0)$.

For the spatial discretization, using $N(u) = 1/2\,(u^2)_x$ and integrating by parts, equation (4.10) may be written as

$$u_l(x) = \int_{-\frac{1}{2}}^{\frac{1}{2}} K_\nu(y,\tau_l)u_0(x-y)dy + \sum_{j=1}^{M_t}\int_{-\frac{1}{2}}^{\frac{1}{2}} L_{\nu,j}(y,\tau_l)u_j^2(x-y)dy, \tag{4.11}$$

where the kernel $L_{\nu,j}(y,t)$ is given by

$$L_{\nu,j}(y,t) = -\frac{1}{2}\int_0^t \left(\partial_y K_\nu\right)(y,t-s)\,R_j(s)ds.$$

For small ν, $K_\nu(y,\tau_l)$ and $L_{\nu,j}(y,\tau_l)$ decay rapidly away from zero. Therefore, we may truncate the integrals in (4.11) to the intervals $(-\delta_l(\nu),\delta_l(\nu))$ and $(-\eta_l(\nu),\eta_l(\nu))$, and then discretize using appropriately chosen quadrature nodes ϕ_p^l and ψ_p^l and weights μ_p^l and γ_p^l,

$$\begin{aligned} u_l(x) &= \int_{-\delta_l}^{-\delta_l} K_\nu(y,\tau_l)u_0(x-y)dy + \sum_{j=1}^{M_t}\int_{-\eta_l}^{\eta_l} L_{\nu,j}(y,\tau_l)u_j^2(x-y)dy \\ &\approx \sum_{p=1}^{M_x}\lambda_p^l u_0\left(x-\phi_p^l\right) + \sum_{j=1}^{M_t}\sum_{p=1}^{M_x}\lambda_{p,j}^l\left(u_j\left(x-\psi_p^l\right)\right)^2. \end{aligned} \tag{4.12}$$

In the last equation,

$$\lambda_p^l = \mu_p^l K_\nu(\phi_p^l,\tau_l), \quad \lambda_{p,j}^l = \gamma_p^l L_{\nu,j}(\psi_p^l,\tau_l),$$

which are computed beforehand given the quadrature nodes.

Remark 6. If the viscosity ν is not small, then the kernels $K_\nu\,(y,t)$ and $L_{\nu,j}\,(y,t)$ are not sharply peaked in space, using the trapezoidal rule is sufficient to obtain a sub-optimal rational representation for $u_l\,(x)$.

References

[1] 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.

[2] G. Beylkin and J. M. Keiser. On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases. *J. Comput. Phys.*, 132(2):233–259, 1997.

[3] G. Beylkin, J. M. Keiser, and L. Vozovoi. A new class of time discretization schemes for the solution of nonlinear PDEs. *J. Comput. Phys.*, 147(2):362–387, 1998.

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

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

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

[7] T. Boros, T. Kailath, and V. Olshevsky. Pivoting and backward stability of fast algorithms for solving Cauchy linear equations. *Linear Algebra Appl.*, 343/344:63–99, 2002. Special issue on structured and infinite systems of linear equations.

[8] C.K. Chui and G. Chen. *Discrete H^∞ Optimization*. Springer-Verlag, 1997.

[9] G. Dahlquist and Å. Björck. *Numerical methods in scientific computing. Vol. I.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.
[10] Ph. Delsarte, Y. Genin, and Y. Kamp. On the role of the Nevanlinna-Pick problem in circuit and system theory. *International Journal of Circuit Theory and Applications*, 9(2):177–187, 1981.
[11] J. Demmel. Accurate singular value decompositions of structured matrices. *SIAM J. Matrix Anal. Appl.*, 21(2):562–580, 1999.
[12] J. Demmel, M. Gu, S. Eisenstat, I. Slapnicar, K. Veselic, and Z. Drmac. Computing the singular value decomposition with high relative accuracy. *LAPACK Working Note*, 119(CS-97-348), 1997.
[13] J. Demmel and K. Veselic. Jacobi's method is more accurate than QR. *SIAM. J. Matrix Anal. and Appl.*, 13(4):1204–1245, 1992.
[14] E. Hille and R. S. Phillips. *Functional Analysis and Semi-groups.* American Mathematical Society, Providence, RI, 1957.
[15] R. A. Horn and C. R. Johnson. *Matrix analysis.* Cambridge University Press, Cambridge, 1990.
[16] 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.
[17] T. Kato. *Spectral theory and differential equations (Proc. Sympos., Dundee, 1974; dedicated to Konrad Jörgens)*, chapter Quasi-linear equations of evolution, with applications to partial differential equations, pages 25–70. Lecture Notes in Math., Vol. 448. Springer, Berlin, 1975.
[18] T. Kato and G. Ponce. Commutator estimates and the Euler and Navier-Stokes equations. *Comm. Pure Appl. Math*, XLI:891–907, 1988.
[19] L. Knockaert. A simple and accurate algorithm for Barycentric rational interpolation. *IEEE Signal Process. Lett.*, 15:154–157, 2008.
[20] S.Y. Kung. Optimal Hankel-norm model reductions: Scalar systems. In *Proceedings of the Joint Automatic Control Conference*, number FA8D, 1980.
[21] D. J. Newman. Rational approximation of $|x|$. *Michigan Math. J.*, 11:11–14, 1964.
[22] J. R. Partington. *An introduction to Hankel operators*, volume 13 of *London Mathematical Society Student Texts.* Cambridge University Press, Cambridge, 1988.
[23] J. R. Partington. *Linear operators and linear systems*, volume 60 of *London Mathematical Society Student Texts.* Cambridge University Press, Cambridge, 2004. An analytical approach to control theory.
[24] T. W. Tee and L. N. Trefethen. A rational spectral collocation method with adaptively transformed Chebyshev grid points. *SIAM J. Sci. Comput.*, 28(5):1798–1811, 2006.
[25] L. N. Trefethen. Rational Chebyshev approximation on the unit disk. *Numer. Math.*, 37(2):297–320, 1981.
[26] L. N. Trefethen and D. Bau, III. *Numerical linear algebra.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.
[27] L. N. Trefethen and M. H. Gutknecht. The Carathéodory-Fejér method for real rational approximation. *SIAM J. Numer. Anal.*, 20(2):420–436, 1983.
[28] N. J. Young. The singular-value decomposition of an infinite Hankel matrix. *Linear Algebra and its Applications*, 50:639–656, 1983.
[29] A. Zakhor, R. Weisskoff, and R. Rzedzian. Optimal sampling and reconstruction of mri signals resulting from sinusoidal gradients. *IEEE Transactions on Signal Processing*, 39(9):2056–2065, 1991.

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