

A Fast Algorithm for Chebyshev, Fourier, and Sinc Interpolation onto an Irregular Grid

JOHN P. BOYD

*Department of Atmospheric, Oceanic, & Space Sciences and Laboratory for Scientific Computation,
University of Michigan, 2455 Hayward Avenue, Ann Arbor, Michigan 48109*

Received November 19, 1990; revised November 8, 1991

A Chebyshev or Fourier series may be evaluated on the standard collocation grid by the fast Fourier transform (FFT). Unfortunately, the FFT does not apply when one needs to sum a spectral series at N points which are spaced *irregularly*. The cost becomes $O(N^2)$ operations instead of the FFTs $O(N \log N)$. This sort of “off-grid” interpolation is needed by codes which dynamically readjust the grid every few time steps to resolve a shock wave or other narrow features. It is even more crucial to semi-Lagrangian spectral algorithms for solving convection–diffusion and Navier–Stokes problems because off-grid interpolation must be performed several times per time step. In this work, we describe an alternative algorithm. The first step is to pad the set of spectral coefficients $\{a_n\}$ with zeros and then take an FFT of length $3N$ to interpolate the Chebyshev series to a very fine grid. The second step is to apply either the M th order Euler sum acceleration or $(2M+1)$ -point Lagrangian interpolation to approximate the sum of the series on the irregular grid. We show that both methods yield full precision with $M \ll N$, allowing an order of magnitude reduction in cost with no loss of accuracy. © 1992 Academic Press, Inc.

1. INTRODUCTION

The problem we pose is that of efficiently evaluating the interpolatory sum

$$f(x) \approx \sum_{j=1}^{\infty} f(x_j) C_j(x) \tag{1.1}$$

at a set of N points $\{x_i\}$ which do not coincide with the interpolation points $\{x_j\}$. The cardinal functions $C_j(x)$ are linear combinations of the underlying Fourier or Chebyshev polynomials which have the property

$$C_j(x_i) = \delta_{ij}, \tag{1.2}$$

that is, the j th cardinal function vanishes at all of the interpolation points except for the j th point, where it is equal to one. We shall give the explicit form of both the cardinal functions and the corresponding interpolation points below for each of the three species of spectral basis sets we discuss: sinc functions, Fourier series, and Chebyshev polynomials.

We shall not explain or justify (1.1) or give a detailed account of interpolatory and pseudospectral algorithms because these are reviewed in the books by Boyd [1] and Canuto *et al.* [2].

One motive for this work is that dynamically-adaptive pseudospectral codes require an “off-grid” interpolation every time the grid is restructured to improve numerical resolution of narrow features like shock waves and fronts. As explained in the articles by Bayliss and Matkowsky [3], Bayliss *et al.* [4], Guillard and Peyret [5], and Augenbaum [6, 7], such schemes compute measures of the local smoothness of the solution at a time level $t = t^n$. Then, the methods perform a change of variable from the original coordinate x to a new coordinate y via a mapping function

$$x = g(y), \tag{1.3}$$

where $g(y)$ is chosen so that the canonical N -point grid in y —an *evenly* spaced grid for Fourier and sinc series—corresponds to an *uneven* resolution in x with grid points clustered around the shock or front. Before the solution can be advanced to the next time level using the new coordinate y , however, the solution must be interpolated from the original grid to the irregular grid whose points $\{x_i\}$ are the images under the mapping of the canonical grid in the new coordinate x , i.e.,

$$x_i = g(y_i), \quad i = 1, \dots, N. \tag{1.4}$$

Another application is semi-Lagrangian time-marching algorithms for solving convection–diffusion problems and the Navier–Stokes equations. These schemes have shown great promise in resolving shock and narrow frontal zones because the physics of shock formation is explicitly included in these algorithms via the method of characteristics [19–23]. A crucial step is the fixed point iteration which looks upstream to determine the location of the fluid particle at the previous time level:

$$\xi_i^{k+1} = x_i - \Delta\tau u^{m-1}(\xi_i^k), \tag{1.5}$$

where k is the iteration counter, x_i is a point on the usual spectral grid (evenly spaced for Fourier and sinc basis sets), Δt is the time step, $u^{m-1}(x)$ is the velocity at the previous time level, and ξ_i^k is the approximation to the characteristic variable $\xi(x)$ at the i th grid point after k fixed point iterations. In typical applications, at least three fixed point iterations are required per time step. Because the values of the characteristic coordinate $\xi(x)$ do not coincide with the points of the regular grid, each iteration requires an off-grid interpolation to compute the values of the velocity $u(\xi(x))$ at the “upstream” points.

The $O(N^2)$ cost of each off-grid interpolation has discouraged the use of spectral semi-Lagrangian schemes. The European Centre for Medium-Range Weather Forecasting has modified its model, the most accurate forecasting code in the world, to use semi-Lagrangian advection. However, the semi-Lagrangian step is performed via finite differences even though the rest of the model employs a spherical harmonic spectral algorithm. Cost is part of the reason that the ECMWF model has become a “mixed metaphor”; the availability of good monotone-preserving finite difference schemes was another reason.

Nevertheless, Suli and Ware [19] have obtained promising results from a semi-Lagrangian spectral code. To break the high cost of off-grid interpolation, they use a Chebyshev polynomial interpolation of $\exp(ik(\xi - [x]))$, where $[x]$ denotes the point on the evenly spaced Fourier grid which is closest to ξ . They report speedups ranging from a factor of seven to a factor of fifty, depending upon the machine. (We shall return to their algorithm in Section 7).

The goal of this article is to describe an alternative fast algorithm for performing this crucial step. As noted in the abstract, the cost of evaluating the N terms in (1.1) at each of N points through direct summation is $O(N^2)$ operations. In addition, because the irregular grid is *different* at each regridding or semi-Lagrangian iteration, the cardinal functions themselves must be evaluated anew at each regridding or iteration (instead of being computed once and for all as a preprocessing step)—also an $O(N^2)$ cost. Since all the other operations of a pseudospectral algorithm can be performed by the fast Fourier transform (FTT) in $O(N \log N)$ operations, the gloomy conclusion is that in the asymptotic limit $N \Rightarrow \infty$, this “off-grid” interpolation is the *rate-determining step* for the entire calculation!

The secret of our fast algorithm is to apply a sum-acceleration method to (1.1). As we shall explain in Section 2, the cardinal series converges so slowly—the terms decrease as $O(1/j)$ —that *truncation* of the series fails because it is too inaccurate. When we multiply the terms of (1.1) by the weights of a sum-acceleration scheme, however, the rate of convergence is vastly improved to the point that only a few terms of the (weighted) series give us the full desired accuracy.

There is, however, a modest complication: the rate of con-

vergence of sum acceleration methods is highly non-uniform in wavenumber. That is, truncating the weighted sums after M terms gives very high accuracy for low wavenumbers but very poor accuracy for wavenumbers near the aliasing limit. We shall explain what we mean by “high” and “low” wavenumbers and the “aliasing limit” in later sections. The important point is that because of this non-uniformity, we can greatly improve the effectiveness of sum acceleration by applying the fast Fourier transform to interpolate from the original grid to a regular grid with finer spacing *before* applying the acceleration.

Thus, we apply one N -point FFT to compute the Chebyshev or Fourier coefficients of $f(x)$, $\{a_n\}$, pad these coefficients with zeros to define a vector of length $N_f > N$, and then apply an N_f -point FFT to calculate $f(x)$ on a grid with a higher density of points than the original grid. Then and only then do we apply sum acceleration. The cost of both FFTs is $O(N \log N)$; the reward is that we can obtain the same accuracy with a much smaller M , that is, a much smaller number of terms in the (weighted) series (1.1) than would be needed on the original grid.

We shall compare two species of sum accelerations: (i) the Euler acceleration and (ii) “economization,” which is polynomial interpolation. The Euler transform is the most widely used and reliable technique for accelerating *slowly converging, alternating* series [8–10]. We shall describe this method in more detail in Section 4.

The other technique is to approximate the pseudospectral solution, which is a high degree polynomial, by a set of overlapping polynomials of much lower degree which are good *local* approximations. By analogy to the standard numerical tactic of approximating a high order Taylor series by a truncated Chebyshev series of much lower order (“economization” or “telescoping” of a power series” [11, 12]), we shall dub this technique “economization.” However, rather than computing the “economized” polynomial as a truncated Chebyshev series, as Lanczos did in his original application, we shall “economize” via a low degree interpolating polynomial.

Lagrangian polynomial interpolation is not usually thought of as a “sum acceleration” but rather as an all-purpose approximation scheme. However, Boyd [10] has shown that centered finite differences, usually derived via polynomial interpolation, may equally well be regarded as the result of applying a “regular, positive, Toeplitz” sum acceleration scheme to the corresponding pseudospectral derivative series. Similarly here, the practical use of Lagrange interpolation is the same as that of the Euler acceleration: to replace the original, slowly converging series by an alternative approximation that needs only $(2M + 1)$ values of $f(x_j)$.

For simplicity, we shall discuss only one-dimensional interpolation here. This restriction entails no loss of generality, because multidimensional spectral calculations

are performed using tensor products of one-dimensional basis sets [1, 2], and multidimensional transforms are normally performed via "partial summation" which reduces them to a sequence of one-dimensional interpolations [1, Chap. 9].

2. CARDINAL FUNCTIONS

2.1. Sinc Basis

The sinc basis is efficient [1] for solving problems on the infinite interval, $x \in [-\infty, \infty]$. The optimum grid is evenly spaced with some constant grid spacing h :

$$x_j = hj, \quad j = 0, \pm 1, \pm 2, \dots \quad (2.1)$$

The cardinal function series is

$$f(x) = \sum_{j=-\infty}^{\infty} f(x_j) C^{\text{sinc}}_j(x), \quad (2.2)$$

where

$$C^{\text{sinc}}_j(x) \equiv \text{sinc}\left(\frac{x-x_j}{h}\right) \quad (2.3)$$

$$\text{sinc}(x) \equiv \frac{\sin(\pi x)}{\pi x}. \quad (2.4)$$

Note also that for the sinc basis, the cardinal functions are all translated-and-rescaled copies of the single, universal function defined by (2.4) and illustrated in Fig. 1.

In practical applications, we usually limit this method to functions which decay exponentially as $|x| \Rightarrow \infty$ and

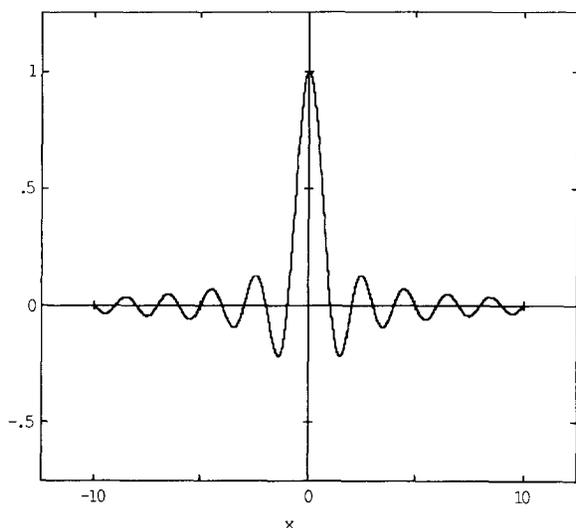


FIG. 1. The Whittaker cardinal (sinc) function.

truncate the infinite grid to those points, where $f(x)$ is non-negligible:

$$x_j = 0, \pm h, \pm 2h, \dots, \pm Nh, \quad \text{where } |f(x)| \ll 1 \\ \forall |x| > Nh. \quad (2.5)$$

Unfortunately, N is usually rather large in practical applications. We can see the possibilities for a more drastic truncation of the cardinal series by employing a trigonometric identity and the above definitions to rewrite the sum without error as

$$f(x) = \frac{\sin(\pi x/h)}{\pi} \sum_{j=-\infty}^{\infty} \frac{(-1)^j f(x_j)}{x/h - j}. \quad (2.6)$$

Unfortunately, however, the $1/(x/h - j)$ factor decreases only as the inverse linear power of j . Thus, if we truncate the infinite pseudospectral grid by discarding all points such that $|j| \geq M$, where $M \ll N$, we will make an unacceptably large error of $O(1/M)$.

However, there are other options. Equation (2.6) shows that the cardinal series is an *almost alternating* series. That is to say, the cardinal sum is strictly alternating if $f(x)$ is a one-signed function. If $f(x)$ has zeros but varies slowly enough so that the spectral method with grid spacing h gives good accuracy, then the series will be "almost alternating" in the sense that the j th term will be opposite in sign from the $(j+1)$ th term for most j . Alternating and almost alternating series are the *raison d'être* of the Euler sum acceleration: the ideal target.

2.2. Fourier Basis

The optimum grid for trigonometric interpolation is also evenly spaced. Without loss of generality, we assume the periodicity interval is 2π :

$$x_j = hj, \quad j = 0, \pm 1, \pm 2, \dots, N \quad (2.7)$$

$$h = \pi/N. \quad (2.8)$$

The cardinal function series is

$$f(x) = \sum_{j=-(N-1)}^N f(x_j) C^{\text{Fourier}}_j(x). \quad (2.9)$$

We omit $x = -Nh$ because, thanks to the periodicity, $f(\pi) = f(-\pi)$. Indeed, in many applications, x is a polar angle (in cylindrical coordinates) or longitude (in spherical coordinates) so that $x = -\pi$ and $x = \pi$ are physically the same point; the lower limit on the sum avoids double-

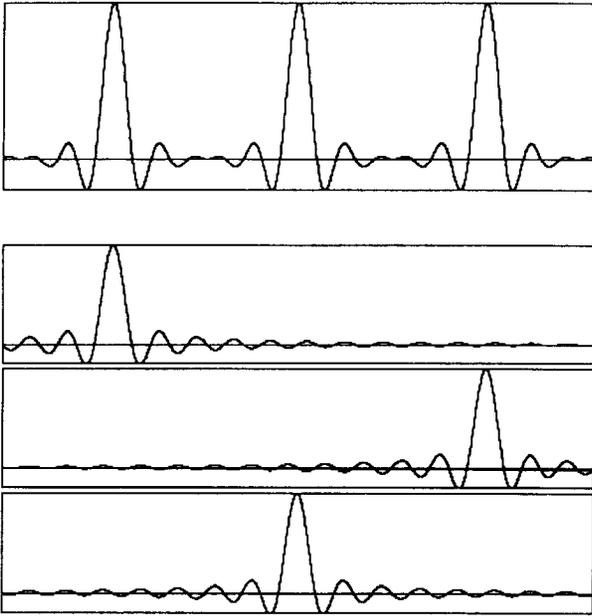


FIG. 2. The Fourier cardinal function for $N = 5$ (top panel). Below it are three of the infinite number of superposed copies of the sinc functions which sum to the Fourier cardinal function.

counting of the value of $f(x)$ at this point. The cardinal functions are

$$C^{\text{Fourier}}_j(x) = C^{\text{Fourier}}_0(x - jh) \tag{2.10}$$

$$C^{\text{Fourier}}_0(x) = \frac{1}{2N} \sin(Nx) \cot(x/2) \tag{2.11}$$

$$C^{\text{Fourier}}_0(x) = \sum_{m=-\infty}^{\infty} \text{sinc}\left(\frac{x - 2\pi m}{h}\right). \tag{2.12}$$

The second definition shows the intimate relationship between the sinc basis (for the infinite interval) and the Fourier basis (for the periodic interval). If we place identical copies of $\text{sinc}(x/h)$ at intervals of 2π over the whole infinite interval, we obtain a function which by construction is periodic with a period of 2π ; this function is a trigonometric polynomial which is the Fourier cardinal function as shown in Fig. 2. It is hardly surprising that Euler acceleration and polynomial economization work as well (and with identical convergence rates) for the Fourier interpolation as for the sinc series.

2.3. Chebyshev Basis

The Chebyshev-Lobatto grid is

$$x_j = \cos(\pi j/N), \quad j = 0, \dots, N. \tag{2.13}$$

The cardinal series is

$$f(x) = \sum_{j=0}^N f(x_j) C^{\text{Cheb}}_j(x) \tag{2.14}$$

$$C^{\text{Cheb}}_j(x) = (-1)^{j+1} (1 - x^2) \frac{dT_N(x)}{dx} \frac{1}{(p_j N^2 [x - x_j])}, \tag{2.15}$$

$j = 0, \dots, N,$

where $p_j = 1$, except for $p_0 = p_N = 2$. Unlike the sinc and Fourier cardinal functions, the Chebyshev cardinal functions are not the translates of a single master function, but rather all have different shapes.

Nevertheless, there is a close relationship between the Chebyshev and Fourier cardinal functions [1]:

$$C_j^{\text{Cheb}}(x) \equiv \frac{1}{p_j} \{ C^{\text{Fourier}}_j(\arccos[x]) + C^{\text{Fourier}}_{-j}(\arccos[x]) \}. \tag{2.16}$$

This is a consequence of the identity

$$T_n(x) = \cos(nt) \quad \text{for all } n$$

if $x = \cos(t) \Leftrightarrow t = \arccos(x)$. (2.17)

Thus, a Chebyshev series is really a Fourier cosine series in disguise, where the “disguise” is the change of variable $x = \cos(t)$. Fourier cosine series define functions which are *symmetric* about $x = 0$, i.e., $f(x) = f(-x)$ for all x . Consequently, the Fourier cardinal functions must be added in pairs, as in (2.16), to give the cardinal functions for the Fourier cosine series. After the change of variable implemented by the inverse cosine functions in (2.16), these cosine cardinal functions become the Chebyshev cardinal functions. Figure 3 schematically summarizes these relationships.

We have summarized the close connections between the sinc, Fourier, and Chebyshev cardinal functions because these interrelationships explain why the off-grid interpolation problem is almost identical for all these different species of basis functions. (And, therefore, why we treat all three cases in a single article.)

We can also see the similarities directly. If we substitute the polynomial definition of the Chebyshev cardinal function into the cardinal series for $f(x)$ and extract the common factors, we obtain

$$f(x) = -\frac{(1 - x^2)}{N^2} \frac{dT_N}{dx}(x) \sum_{j=0}^N \frac{(-1)^j f(x_j)}{p_j(x - x_j)}. \tag{2.18}$$

The summation (2.18) is *identical* in form to that for the sinc series. To be sure, the limits of the sum are finite, the first

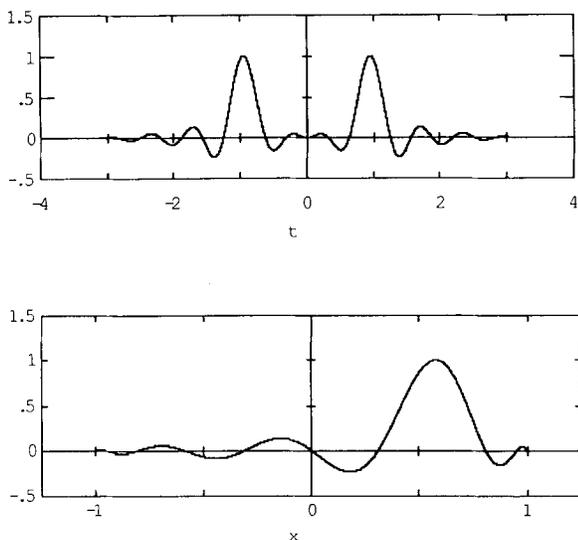


FIG. 3. Top panel: the Fourier cosine cardinal function $j=3$ for $N=10$. Bottom: the corresponding Chebyshev cardinal function. The two functions are identical except for the change of variable $x = \cos(t)$, where t is the argument of the cosine cardinal function and x is the argument of the Chebyshev polynomial. The graph of the Chebyshev cardinal function has only a single maximum versus the double maxima of the cosine cardinal function because the interval $x \in [-1, 1]$ is the image of the right half of the interval in the upper graph.

and last terms are weighted by $\frac{1}{2}$ by the factor p_j , and the grid points $\{x_j\}$ are unevenly spaced in contrast to the even spacing of the sinc and Fourier grids. Nevertheless, the summation is *slowly converging* and *alternating*, and these are the similarities that both permit and demand sum acceleration methods for cardinal series.

3. ALIASING AND ACCURACY

The Fourier cardinal series is a trigonometric polynomial of degree N . In order to develop a deeper understanding of off-grid interpolation, we must ask: how does the cardinal series approximate $\exp(ikx)$ for various k ?

Similarly, the sinc series is a so-called “band-limited” function, that is, one which may be represented exactly as a Fourier transform with finite limits of integration. To understand sinc series is to understand how rapidly the interpolation series converges for $\exp(ikx)$ for each wavenumber k within the “band.”

For both types of series, the included wavenumbers satisfy the inequality

$$|k| \leq \pi/h \equiv k_{\text{limit}}. \quad (3.1)$$

We may dub π/h “the aliasing limit,” k_{limit} , because all higher wavenumbers are “aliased” to lower wavenumbers on a grid with grid spacing h [1, 2].

The cardinal series define only polynomials or band-limited functions, so for purposes of “off-grid” interpolation, we are interested only in wavenumbers *within* the aliasing limit. Nevertheless, the cardinal series are only imperfect solutions to differential equations because these truncated approximations omit the higher wavenumbers that are present in the exact solutions. Consequently, it is senseless to strive for *exact* evaluation of the cardinal series; a more reasonable goal is to interpolate with an error no worse than that inherent in the cardinal series with grid spacing h .

This observation is important because sum acceleration algorithms are intrinsically inexact in the sense that only the full series, which we are desperately trying to avoid summing, can furnish an exact off-grid interpolation. We shall see in Sections 4 and 5, however, that the error of both the Euler and polynomial sum accelerations can be made arbitrarily small by carrying these processes to sufficiently high order.

One complication, however, is that the accuracy of these approximations is highly *non-uniform* in *wavenumber*. That is to say, low wavenumbers are approximated very accurately, whereas wavenumbers whose absolute value is close to the aliasing limit are approximated poorly unless the order M of the Euler series or polynomial approximation is very large. It is for this reason that we recommend using the fast Fourier transform where applicable to first interpolate the cardinal series to a finer grid. When we halve the grid spacing, for example, the cardinal series in $(h/2)$ contains only wavenumbers up to *half* the aliasing limit for a grid of spacing $(h/2)$. It follows that on this fine grid, we need to use only rather small M to accurately approximate all the wavenumbers in the range $k \in [-k_{\text{limit}}/2, k_{\text{limit}}/2]$.

In the next two sections, we create a theory for both Euler acceleration and polynomial economization. All the statements for the sinc and Fourier basis sets apply to Chebyshev polynomials, too, except that “wavenumber” must be redefined as “polynomial degree”; the Chebyshev “aliasing limit” is a polynomial degree of N .

4. THE EULER SUM ACCELERATION

4.1. Justification: Conformal Mapping

The Euler transformation is the “old reliable” of sum acceleration algorithms for *alternating* series. It is a *regular* summation, which means that it is guaranteed to converge to the same result as direct summation of any finite or convergent infinite series to which it is applied. Because it is very *general*, the Euler transformation will yield a new series which will converge exponentially fast when applied to almost any alternating series that is converging or diverging algebraically (that is, with terms growing or decaying as a power of n or more slowly varying functions).

Conformal mapping provides one justification of the Euler acceleration. A slowly converging series of the form

$$S = \sum_{j=0}^{\infty} a_j \tag{4.1}$$

can always be identified as the special case $z = 1$ of the power series

$$S(z) = \sum_{j=0}^{\infty} a_j z^j. \tag{4.2}$$

The reward for making a simple problem more complicated is that we can now apply all the machinery for analytic continuation of a power series to (4.2).

If the power series coefficients a_j are algebraic rather than exponential functions of j , then $S(z)$ is a function whose radius of convergence is one, implying that $S(z)$ has one or more poles or branch points on the unit circle in the complex z -plane. (The formal definition of "algebraic" is that the coefficients satisfy the bound

$$|a_j| \leq \text{const } e^{\epsilon j} \tag{4.3}$$

for arbitrarily small ϵ and all j while it is *not* possible to bound all the coefficients by

$$|a_j| \leq \text{const } e^{-\epsilon j}; \tag{4.4}$$

the informal definition is that the series for off-grid interpolation decay so that $a_j \propto 1/j^k$ for some constant k .)

If the series is *strictly* alternating, i.e., $\text{sgn}(a_j) = -\text{sgn}(a_{j+1})$, then the singularity is at $z = -1$, since this is where the terms of the power series all add in phase. This singularity forces the convergence of the power series to be algebraic everywhere on the unit circle. However, our real goal, $S(1)$, is the value of $S(z)$ at the point *opposite* the singularity; at $z = 1$, $S(z)$ is non-singular and smooth.

The Euler transform is the conformal change-of-variable

$$\zeta = \frac{2z}{1+z} \Leftrightarrow z = \frac{\zeta}{2-\zeta}. \tag{4.5}$$

The singular point $z = -1$ is equivalent to $\zeta = \infty$, so the offending singularity is mapped to infinity. The right half of (4.5) shows that the mapping itself is singular at $\zeta = 2$, so the power series for $S(z[\zeta])$ will have a disk of convergence of radius 2 in the complex ζ -plane. This implies that the coefficients of the transformed series will decrease roughly as $(\frac{1}{2})^j$. This in turn implies that at $z = 1$ ($\Leftrightarrow \zeta = 1$), the transformed series will converge at this rate: every term reduces the error in $S(1)$ by a factor of 2 and every 10 terms reduces the error by a factor of over 1000!

4.2. The Algorithm

The reason that this conformal mapping is practical is that as long as the mapping function $\zeta(z)$ is linear in z as $|z| \rightarrow 0$, as is true of (4.5), then the first M terms of the power series in ζ depend only upon the first M terms of the power series for the mapping function and the first M terms of the series (4.2).

The M th Euler transform is (for $z = 1$)

$$E_M \equiv \sum_{j=0}^M w_{Mj} a_j, \tag{4.6}$$

where

$$w_{Mj} = \sum_{r=j}^M \frac{M!}{2^{Mr} r! (M-r)!}. \tag{4.7}$$

The weights up to and including $M = 8$ are listed in Table I.

One great virtue of the algorithm is simplicity. The conformal mapping formalism, which is so essential to *justify* the Euler transform, is *invisible* in the *algorithm*. To compute E_4 , for example, one merely multiplies the first terms in the series by the five numerical weights in the fourth row of Table I.

One mechanical question remains: how do we match the terms in the cardinal series with those of (4.1)? Inspection of (2.6) shows that for a given x , the factor $1/(x/h - j)$ is largest for $j = n$, where n is the index of the grid point nearest to x . The denominator factors decrease both to the right and the left of x at roughly the same rate. It follows that in order to group the cardinal functions so that the terms in the rearranged series decrease systematically, thereby making it possible for the Euler acceleration to efficiently extrapolate the behavior of the series to its eventual sum, we should identify $f(x_n) C_n(x) \Rightarrow a_0$ and $[f(x_{n+j}) C_{n+j}(x) + f(x_{n-j}) C_{n-j}(x)] \Rightarrow a_j$. The M th Euler transform of the cardinal series becomes

$$E_M(x) \equiv w_{M0} f(x_n) C_n(x) + \sum_{j=1}^M w_{Mj} \times \{f(x_{n+j}) C_{n+j}(x) + f(x_{n-j}) C_{n-j}(x)\}. \tag{4.8}$$

TABLE I

Euler Weights w_{Mj}

M	$j=0$	1	2	3	4	5	6	7	8
1	1	$\frac{1}{2}$							
2	1	$\frac{3}{4}$	$\frac{1}{4}$						
3	1	$\frac{7}{8}$	$\frac{1}{2}$	$\frac{1}{8}$					
4	1	$\frac{15}{16}$	$\frac{11}{16}$	$\frac{5}{16}$	$\frac{1}{16}$				
5	1	$\frac{31}{32}$	$\frac{13}{16}$	$\frac{1}{2}$	$\frac{3}{16}$	$\frac{1}{32}$			
6	1	$\frac{63}{64}$	$\frac{57}{64}$	$\frac{21}{32}$	$\frac{11}{32}$	$\frac{7}{64}$	$\frac{1}{64}$		
7	1	$\frac{127}{128}$	$\frac{115}{128}$	$\frac{99}{128}$	$\frac{1}{2}$	$\frac{29}{128}$	$\frac{1}{16}$	$\frac{1}{128}$	
8	1	$\frac{255}{256}$	$\frac{247}{256}$	$\frac{219}{256}$	$\frac{163}{256}$	$\frac{37}{256}$	$\frac{1}{256}$	$\frac{9}{256}$	$\frac{1}{256}$

It is important to note that the M th-order Euler transform requires $(2M + 1)$ grid point values of $f(x)$.

There are a couple of technical complications. First, when x is near the edges of the interval $[-\pi, \pi]$ (Fourier basis) or the edges of the truncated interval $[x_{-N}, x_N]$, (4.8) will demand values for $f(x)$ and the cardinal functions beyond the edges of the canonical interval. For the sinc basis, values of $f(x)$ outside the truncated grid are negligible, so these missing $f(x_j)$ may be simply ignored. In the Fourier case, we can invoke the periodicity of the solution to supply the missing values:

$$f(x_j) = f(x_j - 2\pi) = f(x_{j-2N}) \quad \text{if } x_j > \pi \quad (4.9)$$

and similarly for grid point values $< -\pi$.

The second complication is that in Chebyshev polynomial applications we found that the convergence of the Euler transform was sharply degraded as soon as the sum in (4.8) reached the edges of the Chebyshev grid. The most effective way to accelerate Chebyshev cardinal series is to make the change of variable

$$t = \arccos(x) \quad (4.10)$$

and then approximate $f(\cos[t])$ by a series of Fourier cardinal functions in t . The Chebyshev interval $x \in [-1, 1]$ maps into only half of the Fourier interval, $t \in [0, \pi]$, but we can supply missing values of $f(\cos[t])$ for negative t by invoking symmetry: $f(\cos[-t]) = f(\cos[t])$ for all t . This approach is equivalent to approximating the Chebyshev cardinal function in the weighted sum (4.8) by

$$C_j^{\text{Cheb}}(x) \approx C_j^{\text{Fourier}}(\arccos[x]) \quad (4.11)$$

and dropping the restriction that C_j^{Cheb} is defined only for $j \geq 0$. The other term in (2.16), $C_{-j}^{\text{Fourier}}(\arccos[x])$, is neglected to avoid "edge-of-the-grid shock." However, the contributions of the missing cardinal functions are all correctly included because symmetry with respect to $t = 0$ is invoked so that the Euler sum uses $(2M + 1)$ grid point values even at the edge of the grid on $t \in [0, \pi]$.

4.3. Convergence Theory

The sinc series for $\exp(ikx)$ is

$$\begin{aligned} e^{ikx} &= \sum_{l=-\infty}^{\infty} e^{ikhl} \operatorname{sinc}\left(\frac{x-lh}{h}\right) \\ &= e^{ikhN} \operatorname{sinc}(X/h) \left\{ 1 + X \sum_{j=1}^{\infty} (-1)^j \right. \\ &\quad \left. \times \left[e^{ikhj} \frac{1}{X-jh} + e^{-ikhj} \frac{1}{X+jh} \right] \right\}, \quad (4.12) \end{aligned}$$

where we have introduced $X \equiv x - nh$; n is the index of the grid point nearest x . Observing that by definition, $|X| \leq h/2$, we see that in the limit of vanishing wavenumber, the series (4.12) is strictly alternating. The conformal mapping argument implies that the Euler accelerated series must converge as $(\frac{1}{2})^M$.

For finite k , however, the $\exp(ikhj)$ factors superimpose a slow oscillation on top of the alternation of signs. How does the Euler acceleration work for nonalternating series?

To answer these questions, it is helpful to introduce

$$\kappa \equiv k/k_{\text{limit}} \quad (4.13)$$

as a grid-independent measure of how close or far the wavenumber is from the aliasing limit. The crucial observation is that as $j \Rightarrow \infty$,

$$\frac{(-1)^j e^{i\pi\kappa j}}{X - jh} = \frac{e^{i\pi(\kappa+1)j}}{h(X/h - j)} \rightarrow -\frac{1}{h} \frac{e^{i\pi(\kappa+1)j}}{j}. \quad (4.14)$$

This implies that the terms in the sum over the first expression in braces in (4.12) are asymptotically proportional to the sum of the power series (for $z = 1$) of the function¹

$$\ln\{1 + ze^{i\pi\kappa}\} = -\sum_{j=1}^{\infty} e^{i\pi(\kappa+1)j} \frac{z^j}{j}. \quad (4.15)$$

Since the convergence of a series is controlled by its asymptotic behavior, we may expect that the behavior of the logarithmic function (4.15) under the Euler transform will faithfully mirror that of the cardinal series (4.12). Indeed, without the approximation (4.14), the series in (4.15) is just a special case of the hypergeometric equation, which is known to have a logarithmic singularity at the same point as (4.15) (Note that we can model the other half of (4.12), the terms from the second part of the expression in braces, by making the replacement $i \Rightarrow -i$ in (4.15).)

Under the conformal mapping $z = \zeta/(2 - \zeta)$, (4.15) becomes

$$\begin{aligned} &\ln\left\{1 + \frac{\zeta}{2 - \zeta} e^{i\pi\kappa}\right\} \\ &= \ln\left\{1 - \zeta \left[\frac{1 - e^{i\pi\kappa}}{2}\right]\right\} - \ln\left\{1 - \frac{\zeta}{2}\right\} \quad (4.16) \\ &\ln\{1 + z(\zeta) e^{i\pi\kappa}\} \\ &= \sum_{j=1}^{\infty} \left\{ \left(\frac{1}{2}\right)^j - \left(\frac{1}{r}\right)^j e^{i\pi(\kappa-1)j/2} \right\} \frac{\zeta^j}{j}, \quad (4.17) \end{aligned}$$

¹ The generalization of (4.15) without the approximation (4.14), that is, with $1/j$ in the denominator of (4.15) replaced by $1/(j - X/h)$, is the hypergeometric function: $(1/v) F(1, v; 1 + v; \exp(i\pi(\kappa + 1))z)$ in the standard notation [13], where $v = -X/h$. Because this hypergeometric function has no simple form, we have analyzed (4.15), but it is known that this function, like that defined by (4.15), has a branch point where $z = \exp(-i\pi(\kappa + 1))$.

where

$$\frac{1 - e^{i\pi\kappa}}{2} = \frac{1}{r} e^{i\pi(\kappa-1)/2} \tag{4.18}$$

$$r = \sqrt{2/(1 - \cos(\pi\kappa))}. \tag{4.19}$$

If we define a measure of the rate of convergence via

$$\gamma = \max(\omega) \quad \text{such that} \quad |a_j| \leq \text{const} \left(\frac{1}{\omega}\right)^j \quad \text{for all } j \tag{4.20}$$

then (4.17) shows that

$$\gamma = \min(2, r(\kappa)) = \begin{cases} 2, & \kappa \leq \frac{1}{3} \\ \sqrt{2/(1 - \cos(\pi\kappa))}, & \kappa > \frac{1}{3}. \end{cases} \tag{4.21}$$

In words, whenever $\kappa \leq \frac{1}{3}$ —that is, whenever the wavenumber k is smaller than $\frac{1}{3}$ of the aliasing limit—each term in the Euler-transformed series is smaller than its predecessor by a factor of 2. Figure 4 shows how γ varies with κ —very rapidly decreasing for $\kappa > \frac{1}{3}$. When $\kappa = \frac{1}{2}$, i.e., a wavenumber that is half the highest unaliased wavenumber, $\gamma = \sqrt{2}$. This means that one must carry the Euler-weighted sum to double the degree in order to obtain the same accuracy for this κ as for $\kappa < \frac{1}{3}$ (see Fig. 5).

Although this analysis is heuristic rather than rigorous, we shall see it is nicely confirmed by the numerical results of Section 6. The same analysis applies with only minor modifications to the Fourier and Chebyshev cardinal series; we omit the details because the graphs below will justify our claims.

For the sinc series, there is unfortunately no good way to reduce the maximum value of k to well below the aliasing limit, k_{limit} . However, if the grid spacing h is fine enough so

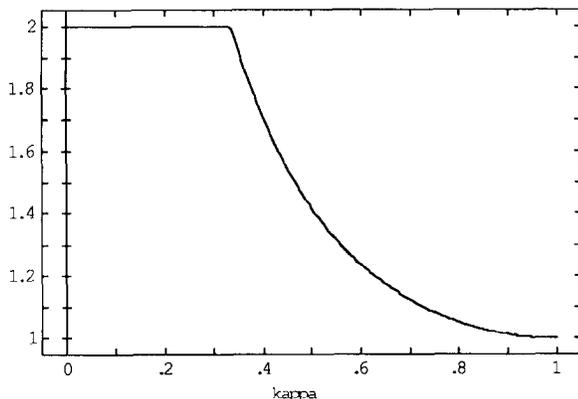


FIG. 4. The convergence factor $\gamma(\kappa)$, defined by (4.21).

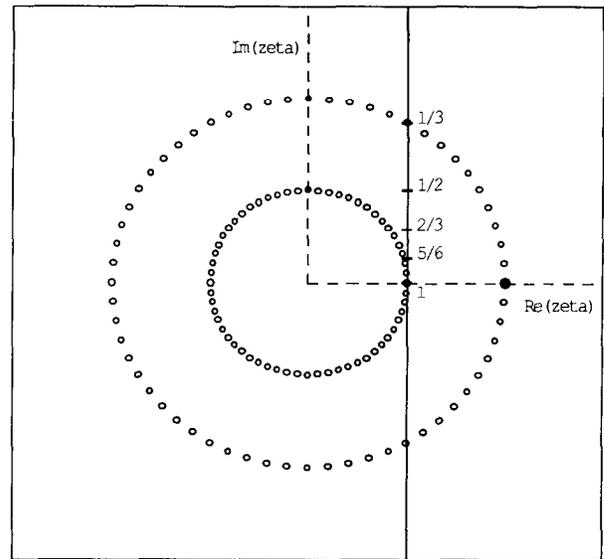


FIG. 5. The complex ζ -plane. The solid circle at $\zeta = 2$ is the singularity induced by the Euler mapping itself; for $\kappa \leq \frac{1}{3}$, the power series in ζ converges within the disk of radius 2 which is indicated by the larger of the two dotted circles. The solid vertical line, $\text{Re}(\zeta) = 1$, is the image of the unit circle in the z -plane; all the logarithmic singularities of the function defined by (4.12) lie on this line for all values of κ . The tick marks on this line denote the images of the singularities for various values of κ , which are indicated by the numerals next to each tick mark. For $\kappa > \frac{1}{3}$, the singularities lie nearer to the real axis than the mapping singularity. In this range of κ , the disk of convergence of the ζ power series is controlled by the logarithmic singularities of the function defined by (4.12). In the limit $\kappa \Rightarrow 1$, the branch point is at $z = 1$, which maps into $\zeta = 1$, and the ζ power series converges only within the unit disk marked by the smaller of the two dotted circles.

that the solution is well resolved, then the amplitude of wavenumbers near the aliasing limit must be small—otherwise wavenumbers slightly greater than the aliasing limit (and therefore unresolved) would necessitate a finer grid. It follows that we can tolerate a poor interpolation for wavenumbers near k_{limit} because the amplitude of these wavenumbers is very small. We shall show through numerical examples in Section 6 that it is possible to realize savings by using the Euler transform or polynomial acceleration for sinc series.

When the basis set may be interpolated to a finer grid via the fast Fourier transform, we recommend this option. Tripling the resolution implies that the maximum wavenumber on the fine grid will be only $\frac{1}{3}$ of the aliasing limit on the original grid. It follows that the convergence factor $\gamma = 2$ for all wavenumbers included in the cardinal series. Thus, every 10 terms in the Euler-weighted series will reduce the error by more than a factor of 1000.

5. POLYNOMIAL ECONOMIZATION

Lagrangian polynomial interpolation is a familiar approximation scheme described in many texts such as [1,

11, 12]. It is well known that for finite difference formulas, which are derived by differentiating the interpolating polynomial, centered difference formulas are much more accurate than one-sided approximations of the same order. Similarly, our M th order polynomial acceleration shall use Lagrangian interpolants which employ $(2M + 1)$ points centered as well as possible on the target value of x . Our scheme uses the $(M + 1)$ points on one side of x (including the grid point nearest x) plus M points on the opposite side of x . Thus, the M th order interpolant employs the same number of points, $(2M + 1)$, as the M th order Euler approximation.

Defining x_n to be the grid point nearest x , as in the previous section, the Lagrangian approximation is

$$P_M(x) = f(x_n) L_0(x) + \sum_{j=1}^M \{f(x_{n+j}) L_{n+j}(x) + f(x_{n-j}) L_{n-j}(x)\}, \quad (5.1)$$

where the $L_j(x)$ are the cardinal functions for $(2M + 1)$ -point Lagrangian interpolation:

$$L_j(x) \equiv \prod_{\substack{k=-M \\ k \neq j}}^M \frac{(x - x_{n+k})}{(x_{n+j} - x_{n+k})}, \quad L_j(x_{n+k}) = \delta_{jk}. \quad (5.2)$$

Equation (5.1) is exactly the same form as the M th-order Euler accelerant of the previous section, Eq. (4.8): a weighted, centered sum of $(2M + 1)$ grid point values of $f(x)$. The only difference is the replacement

$$w_{Mj} C_j(x) \Rightarrow L_j(x) \quad (5.3)$$

which corresponds to a different *weighting* of the grid point values.

As for the Euler acceleration, grid values beyond the edge of the grid are treated differently for each of the three basis sets as follows: The condition for the sinc basis,

$$f(x_j) = 0 \quad \text{for all } |j| > N \quad [\text{Sinc}], \quad (5.4)$$

follows from the constraint that the sinc grid must be sufficiently wide so that all $f(x)$ outside the grid are negligibly small.

The Fourier conditions

$$f(x_j) = f(x_{j-2N}), \quad j > N$$

and [Fourier] (5.5)

$$f(x_j) = f(x_{j+2N}), \quad j < -N$$

follow from the periodicity of $f(x)$. (The Fourier expansion is inappropriate unless $f(x)$ is periodic.)

For the Chebyshev basis, there are two ways to perform the polynomial interpolation. First, we can interpolate $f(\cos[t])$ in the “trigonometric argument” t . In this case,

$$\begin{aligned} f(\cos[t_{-j}]) &= f(\cos[t_j]) \\ \text{and} \quad t_j &= j \frac{\pi}{N}, \quad (5.6) \\ f(\cos[t_{N+j}]) &= f(\cos[t_{N-j}]) \end{aligned}$$

Alternatively, we can perform polynomial interpolation in the “Chebyshev argument” x . In this case, we reduce the number of grid points used to define the interpolating polynomial (and lower its degree) near the edges of the interval $[-1, 1]$ when there are fewer than M points to the left or to the right of x . This does not lead to large errors near ± 1 because the Chebyshev grid points are clustered near these endpoints; one can show that the polynomial interpolation in x has an error which is roughly uniform in x despite the variable degree [1].

Figure 6 compares the errors in these two interpolation schemes for Chebyshev cardinal series. The grid has 97 points, but the function being interpolated is a polynomial of degree 24. Consequently, interpolation in x should be exact for all $M \geq 12$. (Recall that by definition the degree of the interpolation is double the economization order M .) The x -interpolation error does have a pronounced minimum at $M = 12$, but for larger M , the error degrades

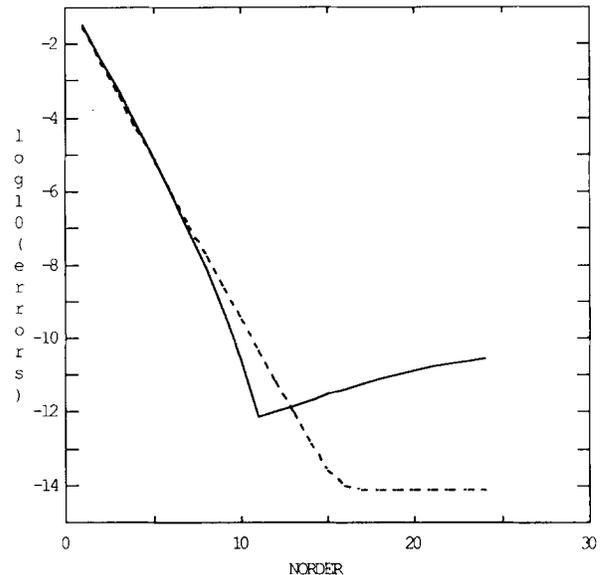


FIG. 6. Chebyshev general Fourier $T - N/4$, $N = 96$. Log_{10} of maximum pointwise error as a function of the order M of the polynomial interpolation scheme. (Note that the degree of the interpolation points is equal to $(2M)$ except possibly near the ends of the interval.) The function $f(x)$ is a polynomial of degree 24 on a grid with $N = 96$, i.e., $f(x)$ contains components up to $\frac{1}{4}$ of the aliasing limit for this grid. Solid: interpolation in x (“Chebyshev argument”). Dashed: interpolation in t (“trigonometric argument”), where $x = \cos(t)$.

because the interpolation on the very unevenly spaced Chebyshev grid is corrupted by roundoff error. In contrast, interpolation in the trigonometric argument is insensitive to roundoff error and levels off at $O(10^{-14})$, a factor of 100 smaller than the minimum for the x -interpolation. For $M < 8$, the two interpolation schemes give errors which are indistinguishable. Even so, the interpolation in t is preferable.

5.2. Convergence Theory

Polynomial interpolation on an evenly spaced grid can be expressed as the Stirling series [14]

$$\begin{aligned}
 f(x) = & f(x_n) + \sum_{j=0}^{M-1} (x/h)(x^2/h^2 - M^2) \xi_j(x) \\
 & \times ([(j-1)!]^2 / (2j+1)!) \mu \delta^{2j+1} f(x_n) \\
 & + \sum_{j=1}^M (x^2/h^2) \xi_j(x) \\
 & \times ([(j-1)!]^2 / (2j)!) \delta^{2j} f(x_n),
 \end{aligned} \tag{5.7}$$

where

$$\xi_j(x) = (-1)^j \prod_{p=1}^j \left(1 - \frac{[x-x_n]^2}{p^2 h^2} \right) \tag{5.8}$$

$$\begin{aligned}
 \delta^2 f(x_j) &= f(x_{j+1}) - 2f(x_j) + f(x_{j-1}) \\
 \mu \delta f(x_j) &= f(x_{j+1}) - f(x_{j-1}).
 \end{aligned} \tag{5.9}$$

The exponential $\exp(ikx)$ is an eigenfunction of the second difference operator:

$$\begin{aligned}
 \delta^2 e^{ikx} &= -4 \sin^2(kh/2) e^{ikx} \\
 &= -4 \sin^2(\pi\kappa/2) e^{ikx}.
 \end{aligned} \tag{5.10}$$

Using the asymptotic relations

$$\xi_j(x) \sim (-1)^j \operatorname{sinc} \left(\frac{x-x_n}{h} \right) \text{ as } j \rightarrow \infty \tag{5.11}$$

$$\frac{[(j-1)!]^2}{(2j)!} \sim \sqrt{\pi} \frac{1}{4^j j^{3/2}}, \tag{5.12}$$

we find that the Stirling series for $\exp(ikx)$ gives back $\exp(ikx)$ multiplied by an infinite series whose real part consists of terms which are asymptotically proportional to those of the series

$$\Psi(\kappa) \equiv \sum_{j=1}^{\infty} \frac{1}{j^{3/2}} [\sin^2(\pi\kappa/2)]^j. \tag{5.13}$$

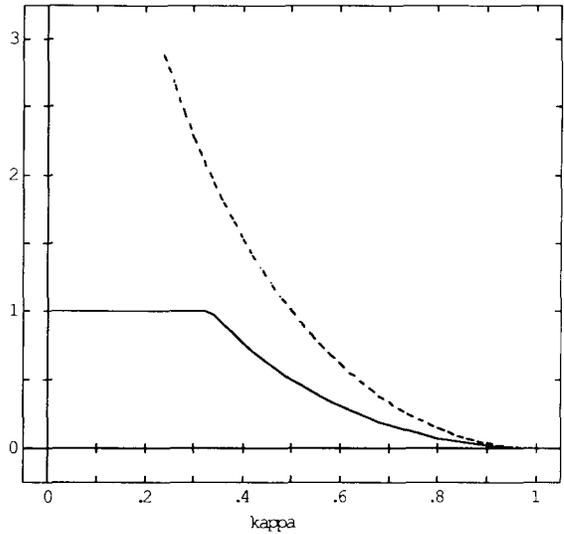


FIG. 7. \log_2 of the convergence factor $\gamma(\kappa)$ for polynomial interpolation (upper curve, dashed) and for Euler acceleration (lower curve, solid).

If we use the measure of convergence defined by (4.20), i.e., the terms of the Stirling series (and the error of M th-order polynomial economization) are decreasing as $O([1/\gamma]^M)$, then (5.13) implies

$$\gamma = 1/\sin^2 \left(\frac{\pi\kappa}{2} \right). \tag{5.14}$$

Figure 7 shows $\gamma(\kappa)$ for the polynomial acceleration and for comparison, for the Euler summation, too. For all values of κ , the polynomial acceleration clearly has a much faster rate of convergence. With such obvious apparent superiority, why discuss the Euler acceleration at all?

The answer is that the weights for the Euler method are independent of j , being functions only of the order of the method. Consequently, these weights can be computed at the beginning of the interpolation and then reused. The cost of the Euler acceleration is linear in the order M .

In contrast, the Lagrangian cardinal functions $L_j(x)$ must be recomputed for each grid point x ; the cost of the polynomial acceleration is proportional to the square of M [14]. Thus, the two algorithms are competitive in terms of cost.

6. NUMERICAL RESULTS

6.1. Simplifications

The first simplification of our numerical results is that the convergence theories of the previous sections are independent of x for all x that do not coincide with a grid point. Simple numerical experiments have confirmed that this is true empirically as well as theoretically. The absolute errors are largest when x is halfway between two grid points, but the rate of convergence is the same for all off-grid x .

Therefore, all the tables and graphs below display, for each M , the maximum absolute error found over the set of all points halfway between collocation points.

The second simplification is that the convergence theories are also *independent of the basis set*. For the expansion of the real part of $\exp(ikx)$, i.e., $\cos(kx)$, the Fourier and Chebyshev results are in fact *identical*, which is obvious once one recalls that Chebyshev polynomials are merely an assumed name for a Fourier cosine series. The Fourier and sinc results are not identical, but numerical experiments showed only slight differences: for a given k and method, the sinc and Fourier errors differ only in the third decimal place, even for small M for most combinations of M and k . It follows that there is no point in comparing results from different basis sets.

The third simplification is that the errors for $\sin(kx)$ ($= \text{Im}[\exp(ikx)]$) are almost identical to those for $\cos(kx)$. Consequently, the graphs and tables below are for the acceleration of the cardinal series of $\cos(kx)$ or for the equivalent Chebyshev polynomial.

Thus, the interesting issues reduce to: How well are the theoretical predictions for the error as a function of the order M confirmed by numerical experiment? How does the Euler method compare with polynomial interpolation as a function of κ , that is, as a function of how close the wavenumber is to the aliasing limit?

6.2. Experiments

Table II addresses the first issue: how accurate is the error theory for polynomial interpolation? The prediction is

$$E^{\text{Polynomial}}_M / E^{\text{Polynomial}}_{M+1} \sim \gamma(\kappa) = \frac{1}{\sin^2(\pi\kappa/2)}, \quad [M \rightarrow \infty], \quad (6.1)$$

where

$$E^{\text{polynomial}}_M \equiv \max_{\text{all } x} |P_M(x) - f(x)|, \quad P_M(x) \equiv (2M + 1) - \text{point Lagrangian interpolant of } f(x). \quad (6.2)$$

We see that for all three values of κ ($\equiv k/k_{\text{limit}}$), the error ratio converges smoothly and almost monotonically from above to the prediction of (6.1).

To reiterate that the convergence theory is almost independent of the basis set, we used a different basis set for each pair of columns in Table II. The Chebyshev errors differed from the Fourier only because of roundoff error; the sinc errors differ only slightly from the Chebyshev and Fourier errors.

TABLE II

The Maximum Pointwise Errors, $E^{\text{polynomial}}_m$, for the Polynomial Interpolation of $\cos(kx)$ for Three Different Values of κ

M	$\kappa = 1/8$		$\kappa = 1/4$		$\kappa = 1/2$	
	E^{poly}_M	Ratio	E^{poly}_M	Ratio	E^{poly}_M	Ratio
1	3.75E-3	—	0.0291	—	0.207	—
2	1.07E-4	35.0	3.24E-3	8.98	0.0821	2.52
3	3.41E-6	31.4	3.98E-4	8.14	0.0352	2.33
4	1.14E-7	29.9	5.12E-5	7.77	0.0157	2.24
5	3.90E-9	29.2	6.77E-6	7.56	7.16E-3	2.19
6	1.36E-10	28.7	9.11E-7	7.43	3.31E-3	2.16
7	4.81E-12	28.3	1.24E-7	7.35	1.55E-3	2.14
8	1.73E-13	27.8	1.70E-8	7.29	7.30E-4	2.12
9			2.36E-9	7.20	3.46E-4	2.11
10			3.29E-10	7.17	1.65E-4	2.10
11			4.60E-11	7.15	7.90E-5	2.09
12			6.46E-12	7.12	3.80E-5	2.08
13			9.12E-13	7.08	1.83E-5	2.08
14			1.31E-13	6.96	8.84E-6	2.07
15					4.28E-6	2.07
16					2.08E-6	2.06
17					1.01E-6	2.06
18					4.91E-7	2.06
19					2.40E-7	2.05
20					1.17E-7	2.05
21					5.71E-8	2.05
22					2.79E-8	2.05
23					1.37E-8	2.04
24					6.70E-9	2.04
Predicted ratio		26.3		6.83		2.00

Note. The first pair of columns is for the sinc basis with $\kappa = \frac{1}{8}$, the second is for Fourier series with $\kappa = \frac{1}{4}$, and the third pair is for Chebyshev polynomials with $\kappa = \frac{1}{2}$. The M th ratio is defined to be the ratio $E^{\text{polynomial}}_{M-1} / E^{\text{polynomial}}_M$. The values of k and N , the number of points on the grid, are not listed because they are irrelevant; only the *relative* magnitude of k relative to the largest alias-free wavenumber for that N , k_{limit} , is relevant as measured by $\kappa \equiv k/k_{\text{limit}}$. (For the Chebyshev calculations, the function being interpolated is not $\cos(kx)$ but rather $T_{\kappa N}(x)$, where $(N+1)$ is the number of grid points.)

Table III is similar except that the Euler method is used. When $\kappa \leq \frac{1}{3}$, the error ratios converge to 2 from above, implying that for the Euler method, each increase in the order M decreases the error by at least a factor of 2.

When $\kappa > \frac{1}{3}$, the error in the Euler method decreases as a *damped oscillation*. For $\kappa = \frac{1}{2}$, the oscillation has a predicted period of 4. Table IIIb therefore lists only every fourth error. The ratios of these errors asymptotes to the predicted ratio of 4, implying an average rate of convergence of $\gamma(\frac{1}{2}) = \sqrt{2}$.

We also note an amusing (but encouraging fact): In all cases, the ratios converge to the predicted value of γ from *above*. This implies that both acceleration schemes are actually converging somewhat *faster* for small to moderate order than predicted by the asymptotic theories of Sections 4 and 5. Brute force summation of the entire cardinal series is wasteful and unnecessary.

TABLE IIIa

The Maximum Pointwise Errors, E^{Euler}_M , for the Euler Acceleration of the Fourier Cardinal Series for $\cos(kx)$ for Three Different Values of κ

M	$\kappa = 1/8$		$\kappa = 1/4$	
	E^{Euler}_M	Ratio	E^{Euler}_M	Ratio
1	0.149	—	0.156	—
2	0.0651	2.29	$6.22E-2$	2.50
3	$2.93E-2$	2.22	$2.79E-2$	2.23
4	$1.35E-2$	2.18	$1.27E-2$	2.20
5	$6.27E-3$	2.15	$6.13E-3$	2.07
6	$2.95E-3$	2.13	$2.84E-3$	2.16
7	$1.39E-3$	2.11	$1.31E-3$	2.17
8	$6.63E-4$	2.10	$6.25E-4$	2.10
9	$3.17E-4$	2.09	$2.98E-4$	2.09
10	$1.52E-4$	2.08	$1.44E-4$	2.07
11	$7.31E-5$	2.08	$6.94E-5$	2.07
12	$3.53E-5$	2.07	$3.33E-5$	2.08
13	$1.71E-5$	2.07	$1.61E-5$	2.07
14	$8.27E-6$	2.06	$7.79E-6$	2.06
15	$4.02E-6$	2.06	$3.78E-6$	2.06
16	$1.95E-6$	2.06	$1.84E-6$	2.05
17	$9.52E-7$	2.05	$8.97E-7$	2.05
18	$4.64E-7$	2.05	$4.37E-7$	2.05
19	$2.27E-7$	2.05	$2.13E-7$	2.05
20	$1.11E-7$	2.05	$1.04E-7$	2.05
Predicted ratio		2.00		2.00

Note. The first pair of columns is for $\kappa = \frac{1}{8}$ and the second is for $\kappa = \frac{1}{4}$. The M th ratio is defined to be $E^{\text{Euler}}_{M-1}/E^{\text{Euler}}_M$.

TABLE IIIb

Same as IIIa but for $\kappa = \frac{1}{2}$

M	E^{Euler}_M	Ratio
4	$2.05E-2$	—
8	$4.81E-3$	4.25
12	$7.32E-4$	6.57
16	$1.47E-4$	4.99
20	$2.94E-5$	4.99
24	$6.20E-6$	4.74
28	$1.34E-6$	4.64
32	$2.94E-7$	4.55
36	$6.56E-8$	4.48
40	$1.48E-8$	4.43
Predicted ratio		4.00

Note. Because the Euler error oscillates (with a period of 4 in M) for this value of κ , we list only every fourth error; similarly, the ratios are defined to be $E^{\text{Euler}}_{M-4}/E^{\text{Euler}}_M$.

Figures 8 and 9 illustrate the rate of convergence of the Euler and polynomial accelerations. For comparison, we have also graphed the errors in the partial sums of the unaccelerated cardinal series. These are hopelessly inferior to both acceleration methods.

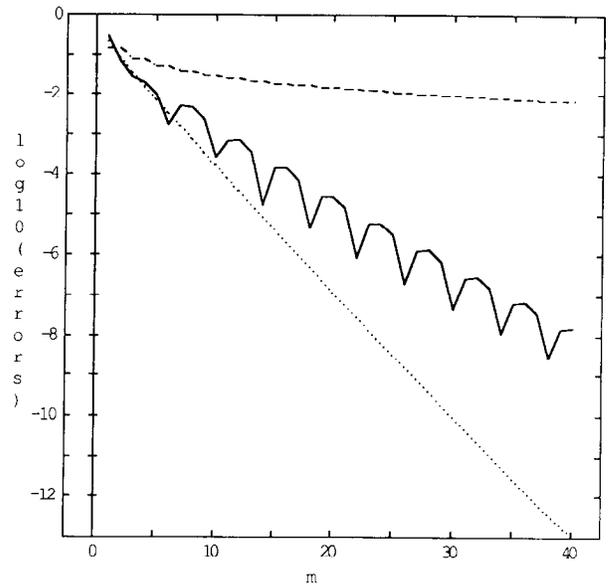


FIG. 8. Log_{10} of the error versus order M : solid (Euler summation), dotted (acceleration-by-interpolation), and dashed (truncation of the cardinal series). Fourier interpolation of $\cos(Nx/2)$ on a grid with $N = 96$ points (although results are almost independent of N).

For both values of κ graphed (and indeed for all κ), the acceleration-by-polynomial-interpolation clearly has a much faster rate of convergence with M than the Euler transform. However, the cost of polynomial interpolation grows as $O(M^2)$. In contrast, the weights of the Euler acceleration are independent of x —they are given, functions only of M , in Table I—so the cost of the Euler scheme is

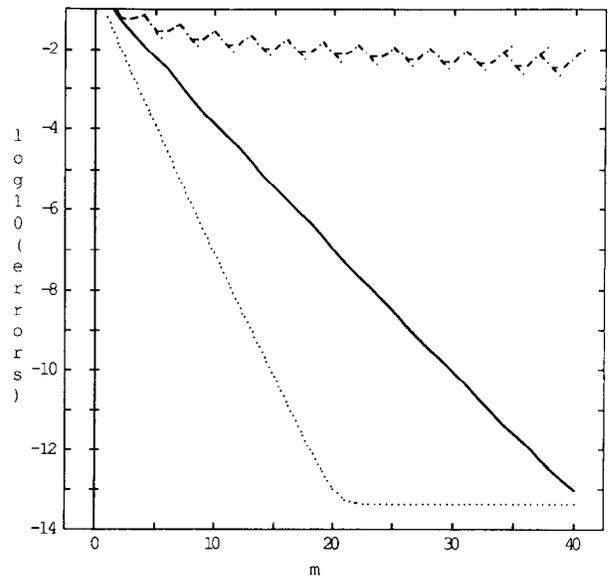


FIG. 9. Log_{10} of the error versus order M : solid (Euler summation), dotted (acceleration-by-interpolation), and dashed (truncation of the cardinal series). This is the same as Fig. 8 except that the function interpolated is $\cos(Nx/3)$, $N = 96$.

linear in the order M . Thus, the two acceleration schemes are competitive in *cost* in spite of the fact that the polynomial acceleration is a clear winner in the convergence rate.

We content ourselves with the vague statement “is competitive” because the actual, practical efficiency of the two competitors depends on the compiler, hardware, and available fast Fourier transform routine. An assembly language, fully vectorized FFT code makes it practical to take a transform which is a higher multiple of N so as to minimize κ . For small κ , the polynomial scheme converges so rapidly that $M < 10$ is sufficient to achieve accuracy to many decimal places. For such small M , the $O(M^2)$ cost of polynomial interpolation is not a burden, and the polynomial acceleration is better. On the other hand, if the FFT is inefficient, then one should minimize the length (and cost) of the FFT by choosing larger κ . This requires higher order for both acceleration schemes to achieve a given error, and larger M , in turn, makes the Euler scheme look better.

So, we shall limit ourselves to describing these alternatives without making a final choice between them. Numerical analysis is (mostly) about factors of 10; factors of two are for accountants and compiler writers.

For the sinc basis, of course, there is no FFT. Boyd [15] has shown that fast multipole methods provide an $O(N \log_2 N)$ transform for sinc series. Unfortunately, the proportionality constant is so large in comparison to the FFT that the best choice for sinc functions is to refrain from the pad-and-transform step which is useful for other basis sets. How effective are sum accelerations when $f(x)$ contains a mixture of spectral components all the way up to the aliasing limit?

7. COMPARISON WITH THE SULI-WARE OFF-GRID INTERPOLATION

Suli and Ware [19] developed an alternative off-grid interpolation scheme for use with their Fourier semi-Lagrangian algorithm. Their method interpolates from the Fourier series form of $f(x)$, rather than the cardinal function series, as ours do. The crucial idea is that the Fourier basis functions are *factorizable*. That is to say, we can write

$$\exp(ikx) = \exp(ikx_n) \exp(ik[x - x_n]), \quad (7.1)$$

where x_n , as earlier, denotes the grid point which is nearest to x on the regular, evenly spaced Fourier grid.

It follows that the Fourier series

$$f(x) = \sum_{j=-N}^N a_j \exp(ijx) \quad (7.2)$$

can be rewritten as

$$f(x) = \sum_{j=-N}^N a_j \exp(ijx_n) \exp(ij[x - x_n]). \quad (7.3)$$

Let us define

$$y \equiv x - x_n, \quad (7.4)$$

where $|y| < h/2$ because there is always some point on the regular grid within half a grid interval of an arbitrary point x . Let us further expand the exponential of y as a Chebyshev polynomial series:

$$\exp(ijy) \approx \sum_{m=0}^M b_m^j T_m(y/(h/2)). \quad (7.5)$$

Equation (7.3) becomes, after reversing the order of the summations,

$$f(x) \approx \sum_{m=0}^M T_m(y/(h/2)) \times \sum_{j=-N}^N a_j b_m^j \exp(ijx_n) \quad (7.6)$$

$$f(x) \approx \sum_{m=0}^M \omega_m(x_n) T_m(y/(h/2)), \quad (7.7)$$

where the coefficients of the Chebyshev series in (7.7) are now the values of the $(M + 1)$ function $\omega_m(x)$ defined by

$$\omega_m(x) \equiv \sum_{j=-N}^N a_j b_m^j \exp(ijx). \quad (7.8)$$

The beauty of their scheme is that the values of the auxiliary functions $\omega_m(x)$ are needed only at points on the regular Fourier grid, i.e., at the x_n in (7.6). It follows that at the price of $(M + 1)$ Fourier transforms, we can evaluate these auxiliary functions on the grid and the only remaining step is to sum the Chebyshev series (7.7), which can be done in $O(MN)$ operations.

The Chebyshev polynomials do not have a simple factorization property like (7.1). It follows that any generalization to the Chebyshev case is likely to be more complicated and expensive than the Fourier algorithm described here.

In their calculations, they took $M = 5$, which requires six FFTs on the standard grid. If, in our algorithm, we “pad” the grid by taking $N_f = 3N$, then (ignoring logarithmic factors), the Suli-Ware procedure requires roughly twice as much FFT work as ours. Since all operations except the FFT are proportional to N for both their method and ours, it follows that in the limit $N \Rightarrow \infty$, where the $O(N \log_2 N)$ cost of the FFTs is dominant both for the Suli-Ware and for our own, our algorithm is superior by a factor of 2.

However, for practical values of N , the factors which are proportional to N are not insignificant. It seems safe to say that the Suli-Ware algorithm is *competitive* with our

algorithms for reasonable N in the Fourier case, but has no obvious superiority.

8. SUMMARY

To interpolate a function $f(x)$ onto an irregular grid, it is unnecessary and wasteful to sum all N terms of the pseudospectral cardinal function series. Via sum acceleration, it is possible to perform the interpolation at a much more modest cost while retaining full spectral accuracy.

The M -term Euler sum acceleration has a cost of only $O(M)$ operations per point, but a relatively slow rate of convergence with the error in the acceleration decreasing as $O([\frac{1}{2}]^M)$. The polynomial economization method has a much faster rate of convergence with M than the Euler scheme, but its cost is $O(M^2)$ per point. Both methods are much cheaper than the $O(N)$ cost per point of direct summation.

The key theoretical concept is that the rate of convergence for these sum accelerations is highly *nonuniform* in wavenumber. We defined the "aliasing limit," k_{limit} , to be the largest wavenumber that can be resolved on the interpolation grid for the Fourier and sinc basis sets; the "aliasing limit" is a polynomial of degree N for N -point Chebyshev interpolation. The sum accelerations are very effective at summing the interpolant of wavenumbers k with $k \ll k_{\text{limit}}$ (that is, polynomials of degree $\ll N$ in the Chebyshev case), but are poor at interpolating wavenumbers or polynomials near the aliasing limit.

For Fourier and Chebyshev polynomials, the remedy is to perform the interpolation onto an irregular grid in two steps. The first is to apply the fast Fourier transform to interpolate $f(x)$ from the original, regular N -point grid to a regular grid of much higher resolution N_f . (For the Euler method, it is most efficient to take $N_f = 3N$, for example.) All the wavenumbers in $f(x)$ have k much smaller than the aliasing limit on the finer grid, so the sum accelerations converge rapidly.

For the sinc basis set, alas, the fast Fourier transform is not applicable. However, because the spectrum of a smooth function decays exponentially with the wavenumber, so that the amplitude of wavenumbers near the aliasing limit is very small, sum acceleration is still effective. To keep the summation error no larger in order of magnitude than the discretization error, however, we must use acceleration of higher order M than would be needed for Chebyshev or Fourier interpolation.

We can apply sum acceleration to evaluate derivatives, too. Boyd [10] develops a fast sinc differential equation solver in which sum acceleration performs most of the operations that would be done via the FFT for Chebyshev and Fourier calculations.

Rosenbaum and Boudreaux [17] and Bisseling *et al.* [18] have anticipated our philosophy by applying summation-by-parts to accelerate off-grid sinc interpolation. However, each summation-by-parts increases the rate of convergence only by one power of the sum index j and they did not use more than two such accelerations. In the limit that the truncation of the sum M is large, our schemes, which converge exponentially fast with M , are much superior to their $1/M^2$ method.

Suli and Ware [19] have developed an alternative method which also converges exponentially with M . In the limit of very large grids, their scheme is inferior to ours by a factor of 2, but for practical values of N , the difference in cost between their algorithm and ours is too close to call. Unfortunately, their off-grid interpolation is restricted to Fourier series only.

These sum acceleration/pseudospectral ideas can be extended in several ways. First, we can apply other, more powerful sum accelerations. Boyd [15] compares a wide variety of sum acceleration methods for quasi-alternating, algebraically converging series. Second, sum accelerations can also be used to solve differential equations directly. Boyd [10] is a beginning; extensions to Chebyshev and Fourier algorithms are well underway [15].

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation through DMS8716766 and OCE8812300. I thank the reviewer for his extremely careful reading and valuable suggestions.

REFERENCES

1. J. P. Boyd, *Chebyshev and Fourier Spectral Methods* (Springer-Verlag, New York, 1989).
2. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics* (Springer-Verlag, New York/Berlin, 1987).
3. A. Bayliss and B. J. Matkowsky, *J. Comput. Phys.* **71**, 147 (1987).
4. A. Bayliss, D. Gottlieb, B. J. Matkowsky, and M. Minkoff, *J. Comput. Phys.* **81**, 421 (1989).
5. H. Guillard and R. Peyret, *Comput. Meth. Appl. Mech. Eng.* **66**, 17 (1988).
6. J. Augenbaum, *Appl. Numer. Math.* **5**, 459 (1989).
7. J. Augenbaum, *Computational Acoustics-Seismo-Ocean Acoustics and Modeling*, edited by D. Lee, A. Cakmak, and R. Vichnevetsky (Elsevier-North Holland, Amsterdam, 1990), p. 19.
8. P. M. Morse and H. Feshbach, *Methods of Theoretical Physics*, Vol. 1 (Wiley, New York, 1953).
9. J. P. Boyd and D. W. Moore, *Dyn. Atmos. Oceans* **10**, 51 (1986).
10. J. P. Boyd, *Appl. Numer. Math.* **7**, 287 (1991).
11. C. Lanczos, *Applied Analysis* (Dover, New York, 1988), p. 457.
12. R. W. Hamming, *Numerical methods for Scientists and Engineers* (Dover, New York, 1986), p. 485.
13. I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series, and Products* (Academic Press, New York, 1965), p. 1075.

14. F. B. Hildebrand, *Introduction to Numerical Analysis* (Dover, New York, 1987), p. 139.
15. J. P. Boyd, *Comput. Meths. Appl. Mech. Eng.*, in press, 1993.
16. J. P. Boyd, *J. Comput. Phys.* **103**, 184 (1992).
17. J. H. Rosenbaum and G. F. Boudreaux, *Geophysics* **46**, 1667 (1990).
18. R. H. Bisseling, R. Kosloff, and D. Kosloff, *Comput. Phys. Commun.* **39**, 313 (1990).
19. E. Suli and A. Ware, *SIAM J. Numer. Anal.* **28**, 423 (1991).
20. P. J. Rasch and D. L. Williamson, *SIAM J. Sci. Stat. Comput.* **11**, 656 (1990).
21. H.-C. Kuo and R. T. Williams, *Mon. Weather Rev.* **118**, 1278 (1990).
22. P. K. Smolarkiewicz and P. J. Rasch, *J. Atmos. Sci.* **48**, 793 (1991).
23. Y. Maday, A. T. Patera, and E. M. Ronquist, *J. Sci. Comput.* **5**, 263 (1990).