Resampling Images in Fourier Domain

GARY M. BERNSTEIN\(^1\) AND DANIEL GRUEN\(^{2,3}\)

Received 2014 January 09; accepted 2014 February 04; published 2014 February 28

ABSTRACT. When simulating sky images, one often takes a galaxy image \(F(x)\) defined by a set of pixelized samples and an interpolation kernel, and then wants to produce a new sampled image representing this galaxy as it would appear with a different point-spread function, a rotation, shearing, or magnification, and/or a different pixel scale. These operations are sometimes only possible, or most efficiently executed, as resamplings of the Fourier transform \(\hat{F}(u)\) of the image onto a \(u\)-space grid that differs from the one produced by a discrete Fourier transform (DFT) of the samples. In some applications, it is essential that the resampled image be accurate to better than one part in \(10^3\), so in this paper, we first use standard Fourier techniques to show that Fourier-domain interpolation with a wrapped sinc function yields the exact value of \(\hat{F}(u)\) in terms of the input samples and kernel. This operation scales with image dimension as \(N^4\) and can be prohibitively slow, so we next investigate the errors accrued from approximating the sinc function with a compact kernel. We show that these approximations produce a multiplicative error plus a pair of ghost images (in each dimension) in the simulated image. Standard Lanczos or cubic interpolators, when applied in Fourier domain, produce unacceptable artifacts. We find that errors less than one part in \(10^3\) can be obtained by (1) fourfold zero-padding of the original image before executing the \(x \rightarrow u\) DFT, followed by (2) resampling to the desired \(u\)-grid using a six-point, piecewise-quintic interpolant that we design expressly to minimize the ghosts, then (3) executing the DFT back to \(x\)-domain.

1. INTRODUCTION

In real images, one obtains a finite, pixelized (i.e., sampled) rendition of objects, for example, the point-spread function (PSF) from stellar images, or galaxy images. Many forms of subsequent analysis require continuum representations of the objects or their Fourier transforms, for example, to resample the objects onto a new grid, or to predict the appearance of the object after rotation, distortion, or convolution with a new PSF. Our principal concern is the validation of weak gravitational lensing (WL) measurement methods: the intrinsic appearance of a galaxy is slightly magnified or sheared by the gravitational deflection of intervening (dark) matter, then the image is convolved with the PSF of the atmosphere and optics. WL software must estimate the lensing effects given a sampled, noisy version of a galaxy ensemble, and must do so to better than part-per-thousand accuracy to recover all of the information available about dark matter and dark energy (Huterer et al. 2006; Amara & Réfrégier 2008). Simulated sky images used to test these methods must therefore have fidelity at least this good in rendering the sheared, convolved versions of realistic galaxies (e.g., Kitching et al. 2012; Mandelbaum et al. 2013).

The PSF of a sky exposure is usually estimated empirically, often by fitting stellar images with a model consisting of an \(N \times N\) grid of pixel values \(a_{ij}\) and a specified interpolation function \(K_x\), such that the continuum representation is

\[
F(x, y) = \sum_{i,j=-N/2}^{N/2-1} a_{ij} K(x - i, y - j),
\]

e.g., as done by the PSFEx software (Bertin 2011). To recover the intrinsic shape of a galaxy that has been observed through this PSF, the observed shape must be corrected for the PSF, an operation that is most straightforwardly done in the Fourier domain (Bernstein 2010), and requires part-per-thousand accuracy in the PSF representation (Huterer et al. 2006; Amara & Réfrégier 2008). One question, therefore, is how to compute the Fourier transform

\[
\hat{F}(u, v) = \int dx dy F(x, y) e^{-2\pi i (ux + vy)}
\]

of a PSF defined by interpolation on a grid of samples. A simple discrete Fourier transform (DFT) is insufficient, as it represents a periodic PSF, and does not include the effects of the interpolation function.

A second question arises when simulating the effect of weak gravitational lensing on real galaxies. This requires taking pixelized images of the real galaxies, then calculating their appearance after application of a lensing shear, convolution with a new
PSF, and sampling on a new pixel grid (e.g., Mandelbaum et al. 2011). Rotation, distortion, or an incommensurate repixelization require interpolation in either real or Fourier space, and the application of convolution favors a Fourier-domain solution. If we need to deconvolve the original image for its PSF, a Fourier-domain solution is strongly favored. Hence we ask: what schemes for interpolation of the Fourier representation of the galaxy \( F(u, v) \) (between DFT samples) are needed to produce a simulated sheared, resampled galaxy at part-per-thousand level? In answering the first question we will find an exact method, but it requires sinc interpolation that can take \( O(N^4) \) operations. We will search for approximate solutions that instead require \( O(K^2 N^2) \) operations for some small kernel size \( K \), but still attain the desired precision in rendering.

We are motivated by the WL science, but there are of course many cases in astrometry and beyond of an accurate method of interpolating images in Fourier domain from a discrete set of points to arbitrary \( u \) values.

Our conventions for Fourier transforms and associated functions are in the Appendix.

2. FOURIER TRANSFORM OF AN INTERPOLATED SAMPLED IMAGE

We assume that an object is correctly modeled as an interpolation of a square, finite grid of values, so that its surface brightness \( F(x, y) \) is defined by the finite set of values \( a_{ij} \) for \(-N/2 \leq i, j < N/2\) as per equation (1). We assume \( N \) is even; an odd-valued \( N \) can be padded with another row and column of zeros. We will assume that the interpolation kernel has \( K_x(0, 0) = 1 \) and \( K_x(m, n) = 0 \) at integer \((m, n)\) other than the origin—this is the condition that the interpolation agrees with the input samples at the sampled locations. We will assume that the kernel is even, \( K_x(x, y) = K_x(-x, -y) \). The value of \( F \) is zero beyond a bounded region if \( K_x \) is. We wish to know the Fourier transform (eq. [2]).

For notational simplicity, we solve the one-dimensional case, which extends easily to two or more dimensions:

\[
F(x) \equiv \sum_{j} a_{j} K_x(x - j)
\]

\[
= (f \ast K_x)(x),
\]

\[
f(x) \equiv \sum_{j=-N/2}^{N/2-1} a_{j} \delta(x - j).
\]

The \( \ast \) indicates convolution, and \( f(x) \) is the original sampled function. From the convolution theorem, we have \( \hat{F}(u) = \hat{f}(u) \hat{K}_x(u) \). If we recognize the discrete Fourier transform (DFT) of the \( a_{j} \) as

\[
\hat{a}_k = \sum_{j=-N/2}^{N/2-1} a_{j} e^{-2\pi ijk/N}, \quad a_{j} = \frac{1}{N} \sum_{k=-N/2}^{N/2-1} \hat{a}_k e^{2\pi ijk/N},
\]

we can write

\[
\hat{f}(u) = \frac{1}{N} \sum_{j=-N/2}^{N/2-1} \sum_{k=-N/2}^{N/2-1} \hat{a}_k e^{2\pi ijk/N} e^{-2\pi iju/N}
\]

\[
= \frac{1}{N} \sum_{k=-N/2}^{N/2-1} \hat{a}_k \sum_{j=-N/2}^{N/2-1} e^{2\pi iju}, \quad v \equiv k/N - u
\]

\[
= \sum_{k=-N/2}^{N/2-1} \hat{a}_k K_u(u - k/N),
\]

\[
K_u(\nu) \equiv e^{i\pi\nu} \sin \frac{\pi N \nu}{N} \sin \frac{\pi \nu}{N} = e^{i\pi\nu}\frac{\sin N \nu}{\sin \nu}.
\]

The function \( \hat{f}(u) \) is equal to the convolution of the DFT, which exists at the points \( u_k = k/N \), with a \( u \)-domain kernel \( K_u \). We need to avoid confusing the two interpolation kernels now involved in the problem: an \( x \)-domain interpolant \( K_x \) that is chosen a priori as the definition of our function \( F(x) \); and the \( u \)-domain interpolant \( K_u \) that we derive as necessary to obtain the exact value of \( \hat{F}(u) \).

The factor \( e^{i\pi\nu} \) in \( K_u \) results from our convention of the samples starting at \( x = -N/2 \) and being asymmetrically placed about the origin. The exact expression (eq. [11]) for \( \hat{F}(u) \) is

\[
\hat{F}(u) = \hat{K}_x(u) \sum_{k=-N/2}^{N/2-1} \hat{a}_k e^{-i\pi\nu} \frac{\sin N \nu}{\sin \nu}.
\]

Note that the ratio of sines (or sincs) in \( K_u \) is equal to the result of wrapping the interpolant \( \sin(N \nu) \) at period 1. The function \( \hat{F}(u) \) extends to infinity unless the \( x \)-domain kernel has a compact transform \( \hat{K}_x(u) \). In practice one will need to truncate at some \( u_{\text{max}} \), in effect defining a band-limited kernel.

We now provide a recipe for the typical application in which one has an image of a galaxy with some input PSF, sampled at unity pixel scale to an \( N \times N \) image. One wants to output an image of exactly the same galaxy after deconvolving the input PSF, applying some affine transformation, convolving with an output PSF, and resampling onto a pixel scale \( \Delta \). Because the output DFT has to have some finite dimension \( M \times M \), we necessarily will be rendering an image that has been folded with period \( M\Delta \), hence it is necessary to choose \( M \) large enough that the folded flux is small enough to ignore. An exact answer
is available via discrete Fourier methods only when the output image is zero outside a bounded region.

The steps are:

1. Obtain the \( \tilde{a}_{ij} \) from the input \( a_{ij} \) via an \( N \)-point DFT to give \( u \)-space values on a grid of pitch \( 1/N \). The operation count is \( O(N^2 \log N) \) using fast Fourier transform (FFT) methods.

2. Select appropriate \( M \) and construct a grid of the output-sampled frequencies \( u'_{mn} \) that will have pitch \( 1/M \Delta \). The real-space affine transformation will be equivalent to a linear transformation and phase change in Fourier domain, so each \( u'_{mn} \) will have a corresponding input \( u_{mn} \).

3. Assign to each \( u_{mn} \) the value obtained from the exact interpolation to \( u = u_{mn} \) of the input image defined by equation (11). This will require \( O(M^2 N^2) \) operations for the \( K_u \) summation, plus \( O(M^2) \) operations to multiply by \( \tilde{K}_x(u_{mn}) \).

4. Divide the DFT by the transform of the input PSF evaluated at \( u_{mn} \) to effect the deconvolution. The operation count is \( O(N^3) \).

5. Multiply by the transform of the output PSF evaluated at \( u'_{mn} \). This is \( O(N^2) \).

6. Execute the DFT back to the desired real-space grid, with \( O(M^2 \log M) \) operations.

The limiting step is the \( u \)-space convolution (3), which will generally take \( O(N^4) \) operations, although in some circumstances the two-dimensional (2D) convolution can be factored to yield \( O(N^3) \), still the slowest step. We will examine below the consequences of approximating this exact interpolation with a more compact kernel.

### 3. Interpolator Accuracy

Consider an interpolation kernel \( K(t) \), with Fourier transform \( \tilde{K}(\nu) \), that is used to reconstruct the value of a sine wave \( \exp(2\pi i \nu_0 t) \) from samples at integral \( t \) values to some nonintegral \( t \). The sinc function has the unique property that it interpolates without error for any frequency \(-1/2 < \nu_0 < 1/2\). We will assume more generally that the interpolation kernel is symmetric, \( K(-t) = K(t) \), such that \( \tilde{K}(\nu) \) is real and symmetric, and is known to be exact at the interpolation nodes, i.e., for integral arguments \( j \),

\[
K(j) = \begin{cases} 
1 & j = 0 \\
0 & j \neq 0
\end{cases}.
\]

(12)

The interpolated reconstruction of the sampled sine wave is

\[
R(\nu_0, t) = \sum_{j=-\infty}^{\infty} K(j - t)e^{2\pi i \nu_0 j}
\]

\[
= [e^{2\pi i \nu_0 t}^{\text{III}(t)}] * K(t)
\]

(13)

and construct a grid of the output-space image is zero outside a bounded region.

The steps are:

1. Obtain the \( \tilde{a}_{ij} \) from the input \( a_{ij} \) via an \( N \)-point DFT to give \( u \)-space values on a grid of pitch \( 1/N \). The operation count is \( O(N^2 \log N) \) using fast Fourier transform (FFT) methods.

2. Select appropriate \( M \) and construct a grid of the output-sampled frequencies \( u'_{mn} \) that will have pitch \( 1/M \Delta \). The real-space affine transformation will be equivalent to a linear transformation and phase change in Fourier domain, so each \( u'_{mn} \) will have a corresponding input \( u_{mn} \).

3. Assign to each \( u_{mn} \) the value obtained from the exact interpolation to \( u = u_{mn} \) of the input image defined by equation (11). This will require \( O(M^2 N^2) \) operations for the \( K_u \) summation, plus \( O(M^2) \) operations to multiply by \( \tilde{K}_x(u_{mn}) \).

4. Divide the DFT by the transform of the input PSF evaluated at \( u_{mn} \) to effect the deconvolution. The operation count is \( O(N^3) \).

5. Multiply by the transform of the output PSF evaluated at \( u'_{mn} \). This is \( O(N^2) \).

6. Execute the DFT back to the desired real-space grid, with \( O(M^2 \log M) \) operations.

The limiting step is the \( u \)-space convolution (3), which will generally take \( O(N^4) \) operations, although in some circumstances the two-dimensional (2D) convolution can be factored to yield \( O(N^3) \), still the slowest step. We will examine below the consequences of approximating this exact interpolation with a more compact kernel.

### 3. Interpolator Accuracy

Consider an interpolation kernel \( K(t) \), with Fourier transform \( \tilde{K}(\nu) \), that is used to reconstruct the value of a sine wave \( \exp(2\pi i \nu_0 t) \) from samples at integral \( t \) values to some nonintegral \( t \). The sinc function has the unique property that it interpolates without error for any frequency \(-1/2 < \nu_0 < 1/2\). We will assume more generally that the interpolation kernel is symmetric, \( K(-t) = K(t) \), such that \( \tilde{K}(\nu) \) is real and symmetric, and is known to be exact at the interpolation nodes, i.e., for integral arguments \( j \),

\[
K(j) = \begin{cases} 
1 & j = 0 \\
0 & j \neq 0
\end{cases}.
\]

(12)

The interpolated reconstruction of the sampled sine wave is

\[
R(\nu_0, t) = \sum_{j=-\infty}^{\infty} K(j - t)e^{2\pi i \nu_0 j}
\]

\[
= [e^{2\pi i \nu_0 t}^{\text{III}(t)}] * K(t)
\]

(13)

The prefactor is the correctly interpolated sine wave, so let us define an error function \( E(\nu_0, t) \equiv e^{-2\pi i \nu_0 t}R(\nu_0, t) - 1 \). Then this error function is

\[
E(\nu_0, t) = \sum_{j=-\infty}^{\infty} \tilde{K}(\nu_0 + j)e^{2\pi i j t} - 1
\]

\[
= \sum_{j=-\infty}^{\infty} \tilde{K}(\nu_0 + j)e^{2\pi i j t} - 1.
\]

(19)

(20)

In the second line we have made use of the fact that equation (12) requires \( \sum_j \tilde{K}(j + \nu_0) = 1 \). The sinc filter has \( \tilde{K}(\nu) = \Pi(\nu) \), which vanishes for \( |\nu| > 1/2 \). The \( E \) function hence vanishes for \( |\nu_0| < 1/2 \), as expected.

### 3.1. Background Conservation

Many astronomical images have signals atop a large constant \( (\nu = 0) \) background. It is hence important that the \( x \)-domain interpolant \( K_x \) have an error function satisfying \( E(0, t) = 0 \), otherwise a resampled image will have background fluctuations as \( t \) varies. Note also that the same criterion dictates whether object flux will be conserved when the interpolant is used to shift the samples by a constant fraction \( t \) of a pixel. Putting \( \nu_0 = 0 \) in equation (20), the fractional background fluctuations will be

\[
E(0, t) = 2 \sum_{j=1}^{\infty} \tilde{K}(j)(\cos 2\pi jt - 1)
\]

\[
\approx 2\tilde{K}(1)(\cos 2\pi t - 1).
\]

(21)

(22)

From the first line, we can see that any interpolant having \( \tilde{K}(j) = 0 \) for \( j \neq 0 \) will conserve background level. The nearest-neighbor and linear filters (see Appendix for definitions) satisfy this exactly since \( \tilde{K}(\nu) = \text{sinc}(\nu) \) and \( \text{sinc}^2(\nu) \), respectively, and the polynomial interpolants are designed to meet this
The Lanczos interpolants do not, however, meet this criterion. In the second line, we have assumed that we are using interpolants, like Lanczos, that attempt to approximate the band-limiting properties of the sinc filter, and will hence have \(|\tilde{K}(1)| \ll 1\) and \(|\tilde{K}(j)| \ll |\tilde{K}(1)|\) for \(|j| \geq 2\). In this case, we can see that

1. The interpolated background error will oscillate as \(\cos 2\pi t_0 - 1\), i.e., will be worst for interpolation to the \(t_0 = 0.5\) midpoint between pixels, and
2. The maximum fractional background error will be \(-4\tilde{K}(1)\).

In practice, the Lanczos interpolant or any other can be normalized to conserve a constant background via

\[
K(t) \rightarrow K(t)/[1 + E(0, t)]
\]

\[
\approx K(t)[1 - 2\tilde{K}(1)\cos 2\pi t - 1]\]

\[
\Rightarrow \tilde{K}(\nu) \rightarrow [1 + 2\tilde{K}(1)]\tilde{K}(\nu) - \tilde{K}(1)[\tilde{K}(\nu + 1) + \tilde{K}(\nu - 1)].
\]

The effect of enforcing background conservation on the interpolant is hence to degrade slightly the band-limiting characteristic of the filter, adding \(O(\tilde{K}(1))\) “wings” to the frequency responses that extend to \(|\nu| = 1.5\).

### 3.2. Interpolation in Fourier Domain

In §2, we inferred that a wrapped sinc interpolation on the grid of DFT Fourier coefficients at \(u_j = j/N\) is the correct way to calculate the transform \(\hat{F}(u)\) at values of \(u \neq u_j\). Here, we examine the errors from approximating the exact \(K_u\) with a smaller interpolation kernel. Let us keep in mind that we now have two distinct interpolants: the \(K_x\) used in real space to define the function \(F(x)\) from the samples \(a_j\), and the \(K_u\) used in Fourier domain to approximate the sinc interpolation \(\hat{f}(u)\) between the samples \(\tilde{a}_j\) at \(u_j = j/N\).

Since all of our operations are linear, we can fully understand the accumulated errors by considering the case when the input \(a_j\) are zero except for a single element \(a_0 = 1\). The DFT produces \(\tilde{a}_j = \exp(-2\pi i j n/N)\), a sine wave advancing by an amount \(\nu_0 = n/N\) cycles per sample, which will be perfectly interpolated to the expected \(\hat{f}(u) = \exp(-2\pi i u n)\) at any \(|u| < 0.5\) if we use the sinc kernel.

With an imperfect \(K_u\), however, we can use equation (20) to infer the error in the interpolated sine wave, putting \(\nu_0 = n/N\) and \(t = uN\):

\[
\hat{f}(u) \rightarrow [1 - E_0(n/N)]\hat{f}(u) + \sum_{j \neq 0} \tilde{K}_u(j + n/N)e^{2\pi i j u} \hat{f}(u),
\]

\[
E_0(\nu) \equiv \sum_{j \neq 0} \tilde{K}_u(j + \nu) = \sum_{j > 0} [\tilde{K}_u(j + \nu) + \tilde{K}_u(j - \nu)].
\]

Upon transforming back to real space, the first term will yield a scaled version of the input function, while the sum becomes a series of ghost images at distance \(jN\) from the original point:

\[
f(x) \rightarrow [1 - E_0(x/N)]f(x) + \sum_{j \neq 0} \tilde{K}_u(j + x/N)f(x - jN).
\]

Now we may generalize to an arbitrary configuration of input samples \(a_j\). The \(\hat{f}(x)\) that is obtained by executing an inverse transform after interpolation of the \(\tilde{a}_j\) will be

\[
\hat{f}(x) = [1 - E_0(x/N)]f(x) + \sum_{j \neq 0} \tilde{K}_u(j + x/N)f(x - jN).
\]

These alterations to \(f(x)\) will be essentially preserved when convolved with \(K_x\) to produce the function \(F(x)\) and its approximation \(\hat{F}(x)\), namely:

1. The approximated version will be multiplied by the function \(1 - E_0(x/N)\).
2. A series of ghosts images appear, displaced by \(jN\) from \(F(x)\) and each multiplied by the function \(\tilde{K}_u(j + x/N)\).

We note further that \(|x/N| < 0.5\), and we are choosing interpolants intended to approximate the band-limited behavior \(\tilde{K}(\nu) = 0\) for \(|\nu| > 0.5\), so it is generally true that \(|\tilde{K}_u(j \pm x/N)| \ll |\tilde{K}_u(1 \pm x/N)|\) for \(j > 1\). In the conditions considered here, the first ghost image dominates.

To quantify the first effect, the left panel of Figure 1 plots the function \(E_0(0)\) that describes the multiplicative errors in the central image of the reconstructed \(\hat{F}(x)\), for the interpolants cataloged in the Appendix. Table 1 lists the interpolants, their kernel sizes, and their error levels at chosen maximum values of \(x/N\). Clearly, any of these interpolants will accrue errors \(\gg 1%\) for \(|x/N| > 0.3\). We can hold \(|x/N| < 0.25\) by zero padding the initial array by a factor of 2 before the initial DFT. The Lanczos interpolants are designed to minimize \(\tilde{K}(\nu)|\nu| > 0.5\) and hence perform well in reducing the interpolation errors determined by \(\tilde{K}(\nu)\) for \(0.75 < |x/N| < 1.25\). The Lanczos filters of order \(\geq 4\) attain \(|E_0(x/N)| < 0.005\) with twofold padding.

Equivalent precision is available from the smaller cubic interpolant kernel if we precompute the DFT with fourfold
zero-padding to keep $|x/N| < 1/8$. In this range, we note that the cubic interpolant performs just as well as the Lanczos interpolants with larger (hence slower) kernels. This is attributable to the cubic interpolant being designed to be exact for polynomials of order $\leq 2$, which is equivalent to the requirement that $(d/d\nu)^m \tilde{K}(\nu) = 0$ for integral $\nu$ and $m \leq 2$. Thus, the cubic interpolant has $\tilde{K}(\nu)$ smaller than a Lanczos interpolant of equal size if we stay sufficiently close to integer values of $\nu$.

Inspired by the success of the standard cubic interpolant, we seek an interpolant that is exact for polynomial functions beyond quadratic order. This will require a six-point interpolant, which must in turn be a piecewise-quintic polynomial function. This quintic filter, described in the Appendix, also has continuous second derivatives. The quintic interpolant indeed satisfies our expectation of performing extremely well if we confine the data to $\nu = x/N < 1/8$ by a fourfold zero-padding of the data.

| Name      | $N_{pts}$ | $u_{max}$ | $2 \times x/N < 1/4$ | $4 \times x/N < 1/8$ | $6 \times x/N < 1/12$
|-----------|-----------|-----------|-----------------------|-----------------------|-----------------------
| Nearest   | 1         | 317.5     | (large)               | (large)               | (large)               |
| Linear    | 2         | 9.6       | 0.18                  | 0.049                 | 0.022                 |
| Cubic     | 4         | 2.74      | 0.061                 | 0.0061                | 0.0016                |
| Quintic   | 6         | 3.62      | 0.037                 | 0.0012                | 0.00015               |
| Lanczos 3 | 6         | 1.49      | 0.014                 | 0.0035                | 0.0035                |
| Lanczos 4 | 8         | 1.35      | 0.005                 | 0.0030                | 0.0019                |
| Lanczos 5 | 10        | 1.08      | 0.004                 | 0.0022                | 0.0012                |
| Sinc      | $\infty$ | 0.5       | 0                     | 0                     | 0                     |

**Notes.**—Interpolants are defined in the Appendix. The parameter $N_{pts}$ is the number of grid points per dimension summed during the interpolation; $u_{max}$ is the largest $|u|$ for which $|\tilde{K}(u)| > 0.001$, giving some estimate of the power beyond the Nyquist frequency $u = 0.5$ induced by the interpolant. The maximum of $|E_0(u)|$ and $|\tilde{K}(1 \pm u)|$ attained for $u < x/N$ is listed for three values of $x/N$, corresponding to 2 times, 4 times, and 6 times zero-padding of the data, respectively.
before the DFT. The multiplicative error $E_0(x/N)$ is $<5 \times 10^{-4}$ in this case.

As an illustration of the second effect, the right panel of Figure 1 plots $\tilde{K}(1 + u)$ for the interpolants, which determines the relative brightness of the first ghost image. We find essentially the same criteria on padding and choice of interpolant as we did from the threshold we imposed on $E_0$. This is assured, as $E_0(\nu) \approx \tilde{K}(1 - u) + \tilde{K}(1 + u)$, given that all filters beyond the linear one have $|\tilde{K}(2 + u)| \leq 0.001$ for $|u| < 0.25$. With fourfold zero-padding, the polynomial filters are again the most efficient at reducing the amplitude of the ghost images. The four-element zero-padding limits the ghost amplitude to 0.006 of the original $a_j$, and the six-element quintic filter reduces the ghost amplitude to 0.0012.

Limiting both the amplitude of the ghost images and the multiplicative error on the central image of $\tilde{F}(x)$ to be $<0.001$ is hence achieved by fourfold zero-padding of the input array, with a DFT followed by interpolation in $u$-space with the six-point quintic interpolant. Similar performance is possible with the more compact four-point cubic interpolant if the initial DFT has sixfold zero-padding.

The $O(N^3)$ computational bottleneck at step 3 of our recipe can hence be replaced by $k^2N^2$ operations in convolution with an interpolant with $k \times k$ footprint, where $k = 6(4)$ for the quintic (cubic) interpolant. The oversampling by factor $s = 4(6)$ increases the time required for the input DFT (step 1) to $O(s^2N^2 \log sN)$, leaving the DFT as the likely slowest step of the process. However, the whole operation now scales, within constant/logarithmic factors, as the number of pixels in the input galaxy images. We note that if we are rendering many versions of a given input galaxy, it may be efficient to precompute and cache the oversampled input DFT. If storage space for this cache is not an issue, then the cubic interpolant with sixfold DFT padding may be faster than quintic interpolant with fourfold DFT.

Figure 2 shows the errors induced in reconstruction of a 2D bull’s-eye image using three different $u$-space interpolation schemes.

4. EFFECT OF WRAPPING

The above section considers the transform from $u$-space back to $x$-space to be continuous, but a discrete transform is necessary in practice. Consider this DFT to produce $M$ samples of the $x$-space image at spacing $\Delta x$. The DFT will sample the $u$-space at intervals $\Delta u = (M\Delta x)^{-1}$ to yield an output image that has been wrapped with period $P = M\Delta x$, such that the output point $\hat{F}_j = \sum_k \hat{F}(j\Delta x + kP)$. The input image $F(x)$ is confined to the extent $\pm N_x/2$ of the original input array $a_j$, thus the radius of the nonzero region for the interpolant $K_x$. Hence, as long as the output DFT has extent $P$ that would contain the input image, the wrapped regions are nominally zero, and the DFT yields an exact sampled representation of the output image.

However, we must recall that, in the case where we have used an approximate interpolant $K_u$, ghost images are present in the output image at the locations $\pm N_u = \pm sN$, where $N_u$ is the size of the $x \rightarrow u$ input DFT, and $s$ is the zero-padding factor applied to the input data array with $N$ samples. If $P = N_x/n$ for some integer $n$, then the principal ghosts will be folded directly atop the primary image. This might seem damaging, but in fact means that the reconstructed $\tilde{F}_j$ will be better, in fact perfect, because the summed ghost images will exactly cancel the multiplicative error $E_0(x/N_x)$ that affects the primary image. Another way to see this is that if $P = N_x/n$, then the $u \rightarrow x$ DFT is using exactly the $u$ values produced by the $x \rightarrow u$ DFT, and no interpolation is being done at all.

Wrapping of the ghost images can be a major problem, though, in the following important application: suppose our goal is to take $\tilde{F}(x)$, dilate it to $G(x) = \tilde{F}(x/(1 + \epsilon))$, then convolve with some PSF function. This is, for example, exactly what one wants to do to simulate a sky that has been sheared by weak...
gravitational lensing—the image must be dilated by $\epsilon = \gamma$ in one direction and contracted with $\epsilon = -\gamma$ along a perpendicular axis. If we wish to execute this dilation in the Fourier domain, setting $\tilde{G}(u) = \tilde{F}[(1 + \epsilon)u]$, then the ghosts will appear at locations $x \approx \pm(1 - \epsilon)N_x$. If we do the $u \to x$ DFT with $P = N_x$, as might be common, then the ghosts are wrapped to positions $x = \pm\epsilon N_x = \pm\epsilon s N$, just slightly displaced. Figure 3 illustrates the results.

These displaced ghosts can generate spurious broadening of the reconstructed image $\tilde{G}_r$. This is a problem when simulating sheared 2D sky images because the interpolation errors induce a quadrupole moment atop that induced by the shearing of the image. We take our "bull's-eye" galaxy and calculate the ellipticity from unweighted quadrupole moments of the image:

$$M_{xx} \equiv \sum_{jk} \tilde{G}_{jk}x_jx_k,$$

$$M_{yy} \equiv \sum_{jk} \tilde{G}_{jk}y_jy_k,$$

$$\epsilon \equiv \frac{M_{xx} - M_{yy}}{M_{xx} + M_{yy}}.$$ (30) (31) (32)

The original bull's-eye pattern has a diameter of 32 pixels, and is 4 times zero-padded to 128 pixels before the $x \to u$ DFT. We construct a sheared image of this galaxy by interpolating in $u$ space and executing the $u \to x$ DFT on a $512 \times 512$ grid with $\Delta x = 0.25$. The quadrupole moment of this reconstruction is compared to that of an image constructed by direct interpolation of the $x$-space image with a third-order Lanczos kernel. If the applied shear produces an ellipticity $\epsilon$ on the $x$-domain interpolation of the sheared image, we find that the reconstruction from cubic $u$-space interpolation produces an ellipticity that is systematically biased by about $0.04\epsilon$. The quintic filter is better, with biases of $\approx 0.004\epsilon$. If we increase the input zero-padding from 4 times to 6 times, we find the quintic filter reduces spurious ellipticities to $\approx 0.001\epsilon$, sufficient for highest-precision simulation of applied shear.

Note that the bull's-eye image is a worst case in the respect that it has full amplitude out to the edge of the initial domain. In practice, we can expect postage-stamp images of galaxies to have near-zero flux at their edges. Since the ghosts of the quintic interpolation rise very rapidly at the edges, a typical galaxy or PSF cutout that has very little flux at the borders will have lower spurious contribution from wrapping of the ghost images.

5. SUMMARY

We seek a recipe for performing high-precision Fourier-domain image simulation and analysis of galaxies and PSFs given as pixelized data, with values between the pixel samples specified by some interpolant $K_x$.

Our first question was: what is the exact expression for the Fourier transform $\tilde{F}(u)$ of such an interpolated, sampled image? Equation (11) gives the answer: first calculate the $\tilde{a}_k$ from a DFT of the input samples; then, interpolate between the $u_k = k/N$ using a wrapped sinc function; and then, multiply by the transform $K_x(u)$ of the $x$-domain interpolant. The maximum frequency $u_{\text{max}}$, with nonzero $\tilde{F}(u)$ hence depends on the choice of interpolant: $K_x = \text{sinc}(x)$ provides strictly band-limited $u_{\text{max}} = 0.5$ but produces an $F(x)$ that extends to a very large distance beyond the original samples. The Lanczos interpolants are a good choice to define an $F(x)$ that is not much larger than the original samples, while not extending $\tilde{F}(u)$ far beyond the original Nyquist frequency.

The sinc interpolation of the $\tilde{a}_k$ is often too slow for practical implementation at $\mathcal{O}(N^4)$, so how could we efficiently obtain the values $\tilde{F}(u)$ at arbitrary $u$ that are needed to implement simulation of sheared and convolved renditions of $F(x)$? Use a $u$-space interpolant $K_u$ that, with a more compact kernel, produces two errors in the simulated images: The first is a multiplicative error $E_0(x/N)$, where $N$ is the size of the $x \to u$ DFT, and $E_0$ is a function characteristic of the interpolant $K_u$. The second error is the appearance of a pair of ghost images located $\pm N$ units from the original image, each ghost multiplied by the function $K_u(1 \pm x/N)$, which is nearly the same size as $E_0(x/N)$.

Figures 1 and Table 1 show the size of these errors for common interpolants. We find that part-per-thousand accuracy on the resampled image and its ellipticity will be realized by the following recipe:

1. Zero pad the input data by a factor of 4 before performing the $x \to u$ DFT.
2. Use the $6 \times 6$ pixel piecewise-quintic-polynomial interpolant in $u$-space.

We therefore recommend use of the quintic filter for $u$-space interpolation after zero-padding and DFT. The cubic filter can be used to gain a factor of 2–3 in speed at the expense of $\approx 5$ times larger simulation errors, and these larger errors can be eliminated by sixfold zero-padding of the input image.

There is one important caveat to this recipe: if the simulated $u$-domain image is transformed back to $x$-domain using a DFT with period $P$ that nearly evenly divides $N$, then the ghosts will be folded atop the primary image. A simulated shear or magnification will cause the ghosts to move relative to the primary image, which, in turn, can change the quadrupole moments of the image in a way that biases shear measurements. In a simple pessimistic trial case, we find that the standard $4\times$ padding + quintic recipe induces 0.4% error in a simple measure of applied shear, somewhat too big for testing state-of-the-art shear measurement techniques. We find it possible to reduce the spurious shear to $<0.001$ of the applied shear by combining the quintic $K_u$ with 6 times zero-padding of the initial DFT, which should be sufficiently accurate for foreseeable cosmic-shear simulations. In fact, this would be overkill for most galaxy images that one is likely to be simulating, since they are already nearly zero at their edges and hence not in need of 6 times padding.

The methods described herein have been implemented as the core of the Fourier-domain image rendering code for the GalSim public-domain sky simulation code$^4$ (B. Rowe et al. 2014, in preparation).

This work was supported by Department of Energy grant DE-SC0007901, National Science Foundation grant AST-0908027, and NASA grant NNX11AI25G. DG was supported by SFB-Transregio 33 “The Dark Universe” by the Deutsche Forschungsgemeinschaft (DFG) and the DFG cluster of excellence “Origin and Structure of the Universe.” We thank Rachel Mandelbaum, Barney Rowe, and R. Michael Jarvis for their comments and efforts in producing usable public code based on these derivations.

APPENDIX.

FUNCTION AND INTERPOLANT DEFINITIONS

We adopt the following conventions for functions and Fourier transforms:

\[ K(x) = \begin{cases} 
1 - |x| & |x| \leq 1 \\
0 & |x| > 1
\end{cases} \]  \hspace{1cm} (A6)

\[ \tilde{K}(u) = \sin^2(u). \]  \hspace{1cm} (A7)

Cubic. — Piecewise-cubic polynomial interpolant with continuous first derivatives designed to interpolate quadratic polynomials perfectly. This implies $K'(j) = K''(j) = 0$ for integers $j \neq 0$. The Fourier-domain expression is analytic but too complex to merit detailing:

\[ K(x) = \begin{cases} 
\frac{1}{2} |x|^3 - \frac{1}{2} x^2 & |x| \leq 1 \\
-\frac{1}{2} |x|^3 + \frac{1}{2} x^2 - 4|x| + 2 & 1 \leq |x| \leq 2
\end{cases} \]  \hspace{1cm} (A8)

Quintic. — A six-point interpolant that provides exact interpolation of fourth-order polynomial functions, and hence has $K(j + \nu) \sim \nu^5$ for $j \neq 0$, can be produced from a piecewise-quintic polynomial kernel. This kernel also has continuous second derivatives:

\[^4\text{https://github.com/GalSim-developers.}\]
Values at $x \leq 0$ are obtained via $K(x) = K(-x)$.

**Lanczos.** — At order $m$ truncates the sinc filter after its $m$th null:

$$K(x) \equiv \text{sinc}(x)\text{sinc}(x/m)\Pi(x/2m),$$

(A10)

$$\tilde{K}(u) \equiv 2m^2\Pi(u) \ast \Pi(u/m) \ast \text{sinc}(2mu)$$

$$= \pi(2mu - m - 1)\text{Si}(2mu - m - 1)$$

$$- \pi(2mu - m + 1)\text{Si}(2mu - m + 1)$$

$$- \pi(2mu + m - 1)\text{Si}(2mu + m - 1)$$

$$+ \pi(2mu + m + 1)\text{Si}(2mu + m + 1).$$

(A11)

**REFERENCES**

Abramowitz, M., & Stegun, I. 1965, Handbook of Mathematical Functions (New York: Dover)

Amara, A., & Réfrégier, A. 2008, MNRAS, 391, 228

Bernstein, G. M. 2010, MNRAS, 406, 2793

Bertin, E. 2011, in ASP Conf. Ser. 442, Astronomical Data Analysis Software and Systems XX, ed. I. N. Evans, et al. (San Francisco: ASP), 435

Huterer, D., Takada, M., Bernstein, G., & Jain, B. 2006, MNRAS, 366 101

Kitching, T. D., et al. 2012, MNRAS, 423, 3163

Mandelbaum, R., Hirata, C. M., Leauthaud, A., Massey, R. J., & Rhodes, J. 2011, MNRAS, 2107

Mandelbaum, R., et al. 2013, preprint (arXiv:1308.4982)