Naval Research Laboratory

Washington, DC 20375-5320

NRL/FR/5320--11-10,207

Planar Arrays on Lattices and Their FFT Steering, a Primer

JEFFREY O. COLEMAN

Advanced Radar Systems Branch
Radar Division

April 29, 2011

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
29-04-2011	Formal Report	

4. TITLE AND SUBTITLE

Planar Arrays on Lattices and Their FFT Steering, a Primer

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61153N

6. AUTHOR(S)

Jeffrey O. Coleman

5d. PROJECT NUMBER

5e. TASK NUMBER
EW021-05-43

5f. WORK UNIT NUMBER
9895

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/5320--11--10,207

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375-5320

10. SPONSOR / MONITOR'S ACRONYM(S)
NRL

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This is a primer for practicing design engineers on two topics. The first is the theory of the simplest planar receive arrays, those positioning identical antenna elements on a point lattice and using terminated guard elements at the array periphery. The second is multi-beam phase-shift steering of such arrays using generalized Cooley-Tukey FFT structures. Array theory here largely avoids electromagnetics and instead uses classic LTI-system arguments from signals and systems. The FFT realization of the general multidimensional DFT for beam steering is developed using nested sublattice chains. The needed lattice basics are covered in detail, and the usual coset decompositions are avoided in favor of a simpler geometric approach based on tiling the element-position and beamspace (direction cosine) planes. Example array architectures use the hexagonal lattice (equilateral triangular grid) with the classic optimal zero-aliasing spacing.

15. SUBJECT TERMS

Array radar Digital arrays FFT beam steering Multidimensional FFT

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Jeffrey O. Coleman
Unclassified	Unclassified	Unlimited	SAR	67	**19b. TELEPHONE NUMBER** *(include area code)* (202) 404-8843

CONTENTS

PLANAR ARRAYS ON LATTICES AND THEIR FFT STEERING, A PRIMER

1. INTRODUCTION

A 2007 NRL memo report [1] formulated the problem of steering a narrowband planar receive array to many regularly spaced beam directions simultaneously using a generalized two-dimensional (2D) discrete Fourier transform (DFT) [2] and then detailed the design of custom fast Fourier transform (FFT) algorithms for realizing such steering digitally. The mathematical heart of the development was the notion of modular arithmetic in 2D point lattices, of taking a lattice modulo one of its sublattices or, equivalently, of taking an integer vector modulo a nonsingular square integer matrix. These are standard notions from elementary group theory, so the report was largely tutorial and simply aimed to give the engineer with no background in abstract algebra a way to acquire the tools needed to work with these algorithms.

The subsequent development of example designs of such array-steering systems for a new, second document—it will be released eventually as a formal NRL report [3]—led to an approach to the design of the necessary custom FFTs that reduced the fancy modular mathematics to the geometry of various tilings of the plane. While this approach is far more engineer friendly, the inclusion of a fully adequate tutorial on the tiling-based approach would have made that document overlong and entirely unwieldy.

So a third document, this report, was called for. The primary topic here is the tiling approach. However, in the spirit of enabling one-stop background shopping prior to tackling either of the first two documents or our group's other research papers on arrays, substantial additional material is included: (1) a basic array-output formulation using minimal electromagnetic theory, (2) those ideas from multidimensional digital signal processing (DSP) that are important in array-pattern and array-steering work, (3) such basics of the theory of point lattices as are needed to connect the DSP and array ideas, and (4) even some key underlying ideas from linear algebra that are perhaps less well grasped by many engineers than would be ideal. There are but two design examples in this document, and they aim not so much to demonstrate the range of what the FFT-steering approach can handle as to demonstrate that the simple cases can be handled simply. More extensive design examples will appear in the second document.

While it is certainly reasonable then to view this third report as a gentle, more manageable introduction to the second, it is just as reasonable to view it as a general-purpose tutorial, aimed at the nonspecialist electrical or even computer engineer with no prior array background, on basic array ideas and on the notations used to represent them in a whole family of array papers. Little is assumed beyond some comfort with matrix algebra and familiarity with basic signals and systems, particularly those aspects concerned with elementary DSP. The one exception is the Appendix, which shows how the generalized 2D DFT in certain special cases can be realized as an ordinary 1D DFT simply through correct bin labeling. That Appendix was left a touch more mathematically demanding simply to keep it relatively brief.

Manuscript approved March 4, 2011.

1.1 Structure of this Report and its Relation to the Literature

The usual time and frequency domains of signals and systems are dual in the sense that

- functions of one domain are Fourier transformed, or inverse transformed, to functions of the other, and
- when time `t` and frequency `f` are swapped in a Fourier-transform property, we speak of the new property obtained (which might require a sign change or a factor of 2π somewhere) as the dual of the first.

Of course we might use a discrete-time index n to represent time, but the essence of the matter is the same.

Likewise, planar-array work involves a position domain and a beamspace domain that are dual in very much the same way. Each of them is two dimensional, technically in the plane of the array in fact, and the first order of business below, in Section 2, is to review enough linear algebra and point-lattice basics to enable us to represent points of interest in those two domains in a convenient way. Using the lattice material just developed, Section 3 then presents a signal-processing engineer's simplified, minimal-electromagnetics view of the simplest of planar arrays: those that are periodic. Section 4 returns to tool building and presents the basics of working with DSP in the *position* $\leftrightarrow$ *beamspace* Fourier domains, including the basics of a 2D discrete-space Fourier transform, which turns out to be nothing at all mysterious but simply an ordinary discrete-time Fourier transform used twice, once for each spatial dimension. In Section 5, the heart of this document, the material of the preceding sections finally comes together in an array theory simple enough to be written in a few lines but detailed enough to permit us to design, as the heart of that section, a simple FFT steering system for an all-digital periodic planar array.

The approach to the 2D FFT of Section 5 is in fact the unified Cooley-Tukey approach presented by Mersereau and Speake [4] in 1981, though this development differs markedly from theirs and aims to be both simpler and much more application oriented. This report should, however, adequately prepare the reader to tackle that classic paper and so gain a sense of how these algorithms are dealt with in the multidimensional-DSP community. In applications where the twiddle-factor multiplications of the Cooley-Tukey approach look to be an undue computational burden, the alternative FFT approach of Guessoum and Mersereau [5] is certainly worth a look, as it uses the Smith normal form of an integer matrix to avoid the need for twiddle factors. The absence of those multiplications helps computational efficiency, but the approach is far less straightforward to grasp.

Every FFT algorithm discussed above implements a generalized 2D DFT in which the familiar number of points N of an ordinary 1D DFT has been replaced with some 2×2 nonsingular integer matrix $\mathbf{R}$ so that $1/N$ in the exponent of the complex exponential becomes $\mathbf{R}^{-1}$, and so forth. Factorization of this integer size parameter [1] or "periodicity matrix" [4, 5] $\mathbf{R}$ into smaller matrices of the same type, where "small" refers to $|\mathbf{R}|$, the absolute value of the determinant, is what leads to the FFT. As a matter of algebra, when $|\mathbf{R}|$ is prime, such factorization is not possible and the 2D FFT reduces to a straightforward, brute-force 2D DFT calculation. The surprise is that, even though the generalized DFT expression shows a factor of $\mathbf{R}^{-1}$ in the exponent of the complex exponential, for prime $|\mathbf{R}|$ that generalized DFT is just an ordinary 1D DFT of $|\mathbf{R}|$ points with carefully chosen 2D bin labels. This is proved in an Appendix.

1.2 Notation

The standard mathematical notation for the set of integers, all of them including the negatives and zero, is $\mathbb{Z}$ and comes from the German verb *zahlen*, to count. It is customary to write the set of all possible N-vectors with elements from $\mathbb{Z}$ as $\mathbb{Z}^N$, but some authors use this notation to mean column vectors, and others use it to mean rows. This document uses both integer row vectors and integer column vectors a great deal, so the usual $\mathbb{Z}^N$ notation is modified here: the set of integer column two-vectors is $\mathbb{Z}^2$, and the set of integer row two-vectors is $\mathbb{Z}^2$.

Other than the above and in complex exponentials $\mathrm{e}^{\mathrm{j}2\pi\langle\text{whatever}\rangle}$, powers arise rarely in this document and are applied directly only to quantities in mathematical "containers" like ( ), | |, or ‖ ‖. This lets us use superscripts relatively freely for identifiers. We can speak of signal s^1, signal s^2, etc., with the latter being "s two" and not "s squared." Subscripts are frequently used as identifiers also, but most often with some spatial meaning. Just as subscripts identify the position of an element in a vector or matrix so that $\boldsymbol{n} = \left[\begin{smallmatrix} n_1 \\ n_2 \end{smallmatrix}\right]$ or $\mathbf{R} = \left[\begin{smallmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{smallmatrix}\right]$, here subscripts can indicate position within some signal that has elements spread over space or time, so that $s^2_{\boldsymbol{n}}$ is the spatial sample of signal s^2 at the position indicated by $\boldsymbol{n}$.

Table 1 summarizes this document's notational conventions for quantities of various types. Most of them are obvious in context, and there is no particular reason to go through them now. The table is simply provided as an easy-to-locate reference for when a quick review of notation is needed.

Table 1 — Notational Conventions

discrete-space function	s^0	(the function as a whole; the "0" could be any label)
particular sample of above	$s^0_{\boldsymbol{n}}$	(vector $\boldsymbol{n}$ could be any allowed index)
above Fourier transformed on $\boldsymbol{n}$ index	$S^0(\boldsymbol{f})$	
or Fourier transformed on $\boldsymbol{n}$ and t both	$\mathrm{S}^0(\boldsymbol{f}, \mathsf{f})$	(seldom used)
matrix	$\mathbf{B}$	(uppercase bold)
three-vector	$\mathbf{k}, \mathbf{x}$	(lowercase bold)
two-vector	$\boldsymbol{k}, \boldsymbol{f}, \boldsymbol{n}$	(lowercase bold italic)
three-vector unit vector	$\widehat{u}, \widehat{v}$	(pointy hat because it points)
real or real vector	$\boldsymbol{f}, \boldsymbol{w}, \mathbf{x}$	(not from center of alphabet)
integer or integer two-vector	$n_1, \boldsymbol{n}, \boldsymbol{m}$	(from center of alphabet)
constants $2.71828\ldots$ and $\sqrt{-1}$	e, j	(the usual letters but in a roman font)
column vector to do with position	$\boldsymbol{\ell}, \boldsymbol{m}, \boldsymbol{n}, \mathbf{x}$	(especially these letters)
row vector to do with beamspace	$\boldsymbol{f}, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}, \mathbf{k}$	(especially these letters)

The conventions for position-related quantities are much as expected. Variable $\mathbf{x}$ is a continuously valued three-vector position, and variable $\boldsymbol{n}$ is a discrete-space sample index (see next section). But the convention here for beamspace quantities, the last line of Table 1, is a touch tricky, as it violates the Table 1 convention of using letters from the center of the alphabet only for integer valued quantities. In particular, this report's beamspace is closely related to that property of a propagating plane wave that in the electromagnetics literature is typically denoted by $\mathbf{k}$ and termed its wavenumber vector, wave vector, or propagation vector. For this variations of the letter k and nearby letters of the alphabet are used for most beamspace quantities. In Section 3.3 below, we see that the vector $\mathbf{k}$ itself in this report represents the same information as that conventional wavenumber vector but in a slightly different way. Furthermore, our use of the discrete Fourier transform (DFT) results in its frequency domain being a subset of beamspace, so the usual DSP convention

of indexing DFT frequency bins with a vector $\boldsymbol{k}$ fits this report's notation scheme naturally enough. Finally, since beamspace is actually a space of vector spatial frequencies, our spatial-frequency vector $\boldsymbol{f}$ falls into this group as well.

2. LATTICE BASICS

2.1 The Array Plane: Bases and their Duals

This one section develops the only linear algebra we need that goes beyond ordinary matrix algebra. The aim is to go slowly and carefully here but to depend as much on a basic visual understanding of the geometry of the plane as on anything very formal.

Certain important basics about the geometry of planes are quite familiar and easily visualized. Let's think about the array plane in particular, and let's suppose it passes through the origin. For now we need not constrain its orientation. Three noncollinear points determine a plane, so take the origin as one of the three points and choose any pair of nonzero, noncollinear three-vectors $\mathbf{b}^1$ and $\mathbf{b}^2$ in the array plane as the other two points. Figure 1 shows an example, with the origin in the center and with the half arrowhead extending clockwise or counterclockwise from a vector's shaft according as it depicts $\mathbf{b}^1$ or $\mathbf{b}^2$, conventions used throughout this document. Making these vectors noncollinear makes them linearly independent, which in essence means that any point $\mathbf{p}$ in the array plane can be reached by summing some amount α of vector $\mathbf{b}^1$ and some amount β of vector $\mathbf{b}^2$ so that $\mathbf{p} = \alpha\mathbf{b}^1 + \beta\mathbf{b}^2$. Such vectors $\mathbf{b}^1$ and $\mathbf{b}^2$ are *basis vectors* for the array plane, and the pair $(\mathbf{b}^1, \mathbf{b}^2)$ of two vectors is a *basis* for the array plane.

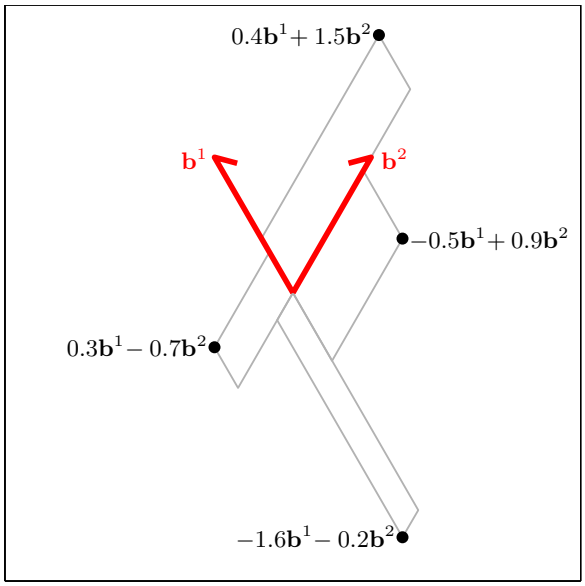


Fig. 1 — Any point in the array plane can be written as a weighted sum of basis vectors $\mathbf{b}^1$ and $\mathbf{b}^2$, shown with half arrowheads extending clockwise and counterclockwise from the shaft respectively.

The whole point of working with a basis in this setting is to be able to conveniently specify a position like $\mathbf{p}$ in the plane with two numbers α and β, a sort of coordinate pair, instead of having to specify x, y, and z components of $\mathbf{p}$. Also, using a basis means we need not be concerned with which combinations of x, y,

and z lie in the plane. We can make α and β anything we like, and $\mathbf{p}$ is in the array plane automatically. We need not restrict basis vectors $\mathbf{b}^1$ and $\mathbf{b}^2$ to have unit length or to be orthogonal to each other. While such orthonormal bases are quite common in other settings, we see below in Section 2.2 that here we have good reasons to avoid requiring orthonormality. It is enough for us that the basis vectors are in the array plane and are linearly independent, i.e., not collinear, that neither is just a scalar multiple of the other.

Matrix notation is very helpful in working with basis vectors. When the vectors are all column vectors, the sum $\mathbf{p} = \alpha\mathbf{b}^1 + \beta\mathbf{b}^2$ can be put in the form

$$\mathbf{p} = \underbrace{\left[\,\mathbf{b}^1\ \mathbf{b}^2\,\right]}_{\substack{3\times 2\\ \text{basis matrix}}} \overbrace{\begin{bmatrix}\alpha\\ \beta\end{bmatrix}}^{\substack{\text{column}\\ \text{two-vector}\\ \text{of weights}}}. \tag{1}$$

When the vectors are instead all row vectors,

$$\mathbf{p} = \overbrace{\begin{bmatrix}\alpha & \beta\end{bmatrix}}^{\substack{\text{row}\\ \text{two-vector}\\ \text{of weights}}} \underbrace{\begin{bmatrix}\mathbf{b}^1\\ \mathbf{b}^2\end{bmatrix}}_{\substack{2\times 3\\ \text{basis matrix}}}.$$

Both column and row forms are used heavily below, columns for position vectors and rows for beamspace vectors. Using both in this way saves us many transposes, because the frequency-time product `ft` that occurs so often in ordinary Fourier mathematics has its counterpart here in the inner product of a beamspace vector and a position vector. Here product ⟨beamspace vector⟩ ⟨position vector⟩ can be written without transposing either quantity.

Let us focus first on column vectors. The matrix-product representation of $\mathbf{p}$ of (1) can be written $\mathbf{p} = \mathbf{B}\boldsymbol{w}$, where 3×2 *basis matrix* $\mathbf{B}$ has basis vectors $\mathbf{b}^1$ and $\mathbf{b}^2$ as its columns and where real column two-vector $\boldsymbol{w}$ contains the particular weights needed to produce $\mathbf{p}$. But given $\mathbf{B}$ and $\mathbf{p}$, can we solve for $\boldsymbol{w}$? We certainly cannot just write $\boldsymbol{w} = \mathbf{B}^{-1}\mathbf{p}$, because the basis matrix $\mathbf{B}$, being nonsquare, cannot be inverted! But in fact the *Gram matrix* of all the dot products of the basis vectors, given by $\mathbf{B}^{\mathrm{T}}\mathbf{B}$, always can be inverted. Before we try to use this fact, let us see why it is reasonable to believe it is true.

Suppose real two-vector $\boldsymbol{u}$ is given. Must there exist a real array-plane three vector $\mathbf{x}$ such that $\boldsymbol{u} = \mathbf{B}^{\mathrm{T}}\mathbf{x}$? If we express the basis vectors as lengths times unit vectors, as $\mathbf{b}^1 = \|\mathbf{b}^1\|\,\widehat{b}^1$ and $\mathbf{b}^2 = \|\mathbf{b}^2\|\,\widehat{b}^2$, the condition $\boldsymbol{u} = \mathbf{B}^{\mathrm{T}}\mathbf{x}$ becomes

$$\begin{aligned} u_1 &= \mathbf{b}^1\cdot\mathbf{x} \\ u_2 &= \mathbf{b}^2\cdot\mathbf{x} \end{aligned} \quad \begin{matrix}\text{or}\\ \text{equivalently}\\ \text{just}\end{matrix} \quad \begin{aligned} u_1/\|\mathbf{b}^1\| &= \widehat{b}^1\cdot\mathbf{x} \\ u_2/\|\mathbf{b}^2\| &= \widehat{b}^2\cdot\mathbf{x}. \end{aligned}$$

The quantities on the far right are just the components of $\mathbf{x}$ in the $\widehat{b}^1$ and $\widehat{b}^2$ directions, so just thinking visually, the answer is yes, there is certainly some array-plane $\mathbf{x}$ that extends to a prescribed degree along

the directions of the basis vectors, and it is in fact unique. (The linear independence of the basis vectors is critical!) And of course that $\mathbf{x}$, simply because it is in the array plane, can itself be written as $\mathbf{x} = \mathbf{B}\boldsymbol{w}$ for some real two-vector $\boldsymbol{w}$ of weights (even though we don't quite know yet how to actually solve for $\boldsymbol{w}$) so that $\boldsymbol{u} = \mathbf{B}^{\mathrm{T}}\mathbf{x} = \mathbf{B}^{\mathrm{T}}\mathbf{B}\boldsymbol{w}$. We have in effect argued that for any $\boldsymbol{u}$ it's possible to invert the system $\boldsymbol{u} = \mathbf{B}^{\mathrm{T}}\mathbf{B}\boldsymbol{w}$ of linear equations to find unique solution $\boldsymbol{w}$. We can also transpose that system of linear equations and write it as $\boldsymbol{u}^{\mathrm{T}} = \boldsymbol{w}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{B}$. Since we can obtain $\boldsymbol{w}$ from $\boldsymbol{u}$ in either case, we can effectively undo a multiplication by $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ irrespective of whether that multiplication was done on the left or right. This is the very meaning of the statement that matrix $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ is invertible.

So back to our original problem, that of solving $\mathbf{p} = \mathbf{B}\boldsymbol{w}$ for $\boldsymbol{w}$. We can premultiply everything by $\mathbf{B}^{\mathrm{T}}$ to obtain $\mathbf{B}^{\mathrm{T}}\mathbf{p} = \mathbf{B}^{\mathrm{T}}\mathbf{B}\boldsymbol{w}$, and now we can use the inverse that we know must exist and write $(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{p} = \boldsymbol{w}$. Comparing to our original relationship $\mathbf{p} = \mathbf{B}\boldsymbol{w}$ shows that $(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}$ inverts left multiplication by $\mathbf{B}$. It does not, however, invert right multiplication by a tall $\mathbf{B}$, so we cannot properly term it the inverse of $\mathbf{B}$. A different terminology is called for.

The *Moore-Penrose pseudoinverse* of any matrix $\mathbf{B}$, which is calculated in MATLAB by the `pinv( )` function, is defined for a "tall" matrix $\mathbf{B}$ like ours by[1]

$$\mathbf{B}^{+} \triangleq (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}. \tag{2}$$

Now if we had three basis vectors in $\mathbf{B}$, the largest number of linearly independent vectors one can ever have in threespace, basis matrix $\mathbf{B}$ would be square. Definition (2) would simplify to yield $\mathbf{B}^{+} = \mathbf{B}^{-1}$, a true inverse. But our basis matrix $\mathbf{B}$ has only two basis-vector columns, so that simplification is not possible here. If does not quite always invert then, what exactly does it do? How does it behave?

Several properties of the pseudoinverse are important to us here. The rows of $\mathbf{B}^{\mathrm{T}}$ on the right in definition (2) are just $\mathbf{b}^{1\mathrm{T}}$ and $\mathbf{b}^{2\mathrm{T}}$, the transposes of our original array-plane basis vectors. If we for the moment write the inverse of the Gram matrix as $\mathfrak{D} \triangleq (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}$ so that pseudoinverse definition (2) becomes just $\mathbf{B}^{+} = \mathfrak{D}\mathbf{B}^{\mathrm{T}}$, the rows of $\mathbf{B}^{+}$ are then given by

$$(\mathbf{B}^{+})_{\text{row 1}} = \mathfrak{D}_{1,1}\,\mathbf{b}^{1\mathrm{T}} + \mathfrak{D}_{1,2}\,\mathbf{b}^{2\mathrm{T}},$$
$$(\mathbf{B}^{+})_{\text{row 2}} = \mathfrak{D}_{2,1}\,\mathbf{b}^{1\mathrm{T}} + \mathfrak{D}_{2,2}\,\mathbf{b}^{2\mathrm{T}}.$$

The four elements of the 2×2 matrix $\mathfrak{D}$ become weights used to combine the original two basis vectors, now oriented as rows, to make two new vectors, the rows of $\mathbf{B}^{+}$. A simple conclusion follows.

First useful property: the rows of $\mathbf{B}^{+}$ are each in the array plane.

Next let's ask, are these rows linearly independent? Vectors are linearly independent if their linear combinations are unique. For example, since the basis-vector columns of basis matrix $\mathbf{B}$ are linearly independent, whenever both $\mathbf{p} = \mathbf{B}\boldsymbol{u}$ and $\mathbf{p} = \mathbf{B}\boldsymbol{v}$ are true it must be that $\boldsymbol{u} = \boldsymbol{v}$ as well, because there can

[1] We won't actually need the pseudoinverse of any wide matrices, but if we did we would use the easily remembered fact that $\mathbf{B}^{+\mathrm{T}} = \mathbf{B}^{\mathrm{T}+}$, that transposing and taking the pseudoinverse commute and so can be done in either order. For a wide $\mathbf{B}$ then, we can transpose both sides of this to obtain $\mathbf{B}^{+} = \mathbf{B}^{\mathrm{T}+\mathrm{T}}$ and then apply the definition of pseudoinverse to tall matrix $\mathbf{B}^{\mathrm{T}}$. For the record, combining the steps algebraically yields that when $\mathbf{B}$ is wide, $\mathbf{B}^{+} = \mathbf{B}^{\mathrm{T}}(\mathbf{B}\mathbf{B}^{\mathrm{T}})^{-1}$.

be only one set of weights that can be used to combine the basis vectors to reach a given point. Of course, an equivalent way to say this is that if $\mathbf{B}(\boldsymbol{u}-\boldsymbol{v}) = 0$, it follows that $\boldsymbol{u}-\boldsymbol{v} = 0$. The classic mathematical definition of linear independence is along those lines: the columns of $\mathbf{B}$ are linearly independent if and only if $\mathbf{B}\boldsymbol{w} = 0$ implies $\boldsymbol{w} = 0$.

Applying a row version of the definition of linear independence to the pseudoinverse then, rows of $\mathbf{B}^+$ are linearly independent if and only if $\boldsymbol{f}\mathbf{B}^+ = 0$ implies row two-vector $\boldsymbol{f} = 0$. So suppose $\boldsymbol{f}\mathbf{B}^+ = 0$. By definition (2) then, $\big(\boldsymbol{f}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\big)\mathbf{B}^{\mathrm{T}} = 0$. But the rows of $\mathbf{B}^{\mathrm{T}}$ are just the columns of $\mathbf{B}$ and so are linearly independent, so the weight vector $\boldsymbol{f}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1} = 0$. Right multiply by Gram matrix $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ and we have that $\boldsymbol{f} = 0$, revealing another fact.

Second useful property: the rows of $\mathbf{B}^+$ are linearly independent.

Another useful property then follows from the first two.

Third useful property: the rows of $\mathbf{B}^+$ form a basis for the array plane.

In general, the rows of $\mathbf{B}^+$ are a different basis than the one we began with, the columns of $\mathbf{B}$. (Actually they are the same basis vectors if but only if $\mathbf{b}^1$ and $\mathbf{b}^1$ are orthonormal.) The rows of $\mathbf{B}^+$ are just as legitimate a basis for the array plane, however. A simple identity is the key to the usefulness of using $\mathbf{B}^+$ as a basis.

Fourth useful property:

$$\mathbf{B}^+\mathbf{B} = \mathbf{I}. \tag{3}$$

Substitution of definition (2) gives the proof immediately: $\mathbf{B}^+\mathbf{B} = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{B} = \mathbf{I}$.

Our basis-choice strategy will be to use $\mathbf{B}$ and $\mathbf{B}^+$ respectively as bases for array-plane row vectors and column vectors, for two reasons. First , identity (3) then always makes solving for the weights easy. Multiply column vector $\mathbf{p} = \mathbf{B}\boldsymbol{w}$ on the left by $\mathbf{B}^+$ and use the identity to write $\mathbf{B}^+\mathbf{p} = \mathbf{B}^+\mathbf{B}\boldsymbol{w} = \boldsymbol{w}$. Or multiply row vector $\mathbf{k} = \boldsymbol{f}\mathbf{B}^+$ on the right by $\mathbf{B}$ and use the identity to write $\mathbf{k}\mathbf{B} = \boldsymbol{f}\mathbf{B}^+\mathbf{B} = \boldsymbol{f}$. Second, identity (3) greatly simplifies products of the general form ⟨beamspace vector⟩ ⟨position vector⟩, where both vectors are in the array plane. Any such row vector ⟨beamspace vector⟩ can be written as $\boldsymbol{f}\mathbf{B}^+$ for some weight vector $\boldsymbol{f}$, and any such column vector ⟨position vector⟩ can be written as $\mathbf{B}\boldsymbol{n}$ for some weight vector $\boldsymbol{n}$. (Beginning in Section 2.2 the latter weights are integers, which is why the letter n seems a good choice here.) Then the product simplifies because identity (3) gives ⟨beamspace vector⟩ ⟨position vector⟩ $= (\boldsymbol{f}\mathbf{B}^+)(\mathbf{B}\boldsymbol{n}) = \boldsymbol{f}\boldsymbol{n}$. Our two bases simply drop out, an enormous convenience mathematically. In honor of this very special property, we can reasonably call the columns of $\mathbf{B}$ and the rows of $\mathbf{B}^+$ *dual bases*. Each basis is the dual of the other. (Some authors instead term them *reciprocal bases*.)

Some examples of bases and their duals are shown in Fig. 2, along with several properties helpful in sketching dual bases without actually computing a pseudoinverse, particularly for the case common in array work in which basis vectors have equal lengths. Proofs of those properties are omitted here because they are tedious, but none is particularly difficult. Of special note is the last property listed: basis vectors of equal lengths result in equal-length vectors in the dual basis as well, and the stems of the original and dual "tees," drawn lightly in the figure, have lengths that multiply to exactly $1/2$. Of note as a result:

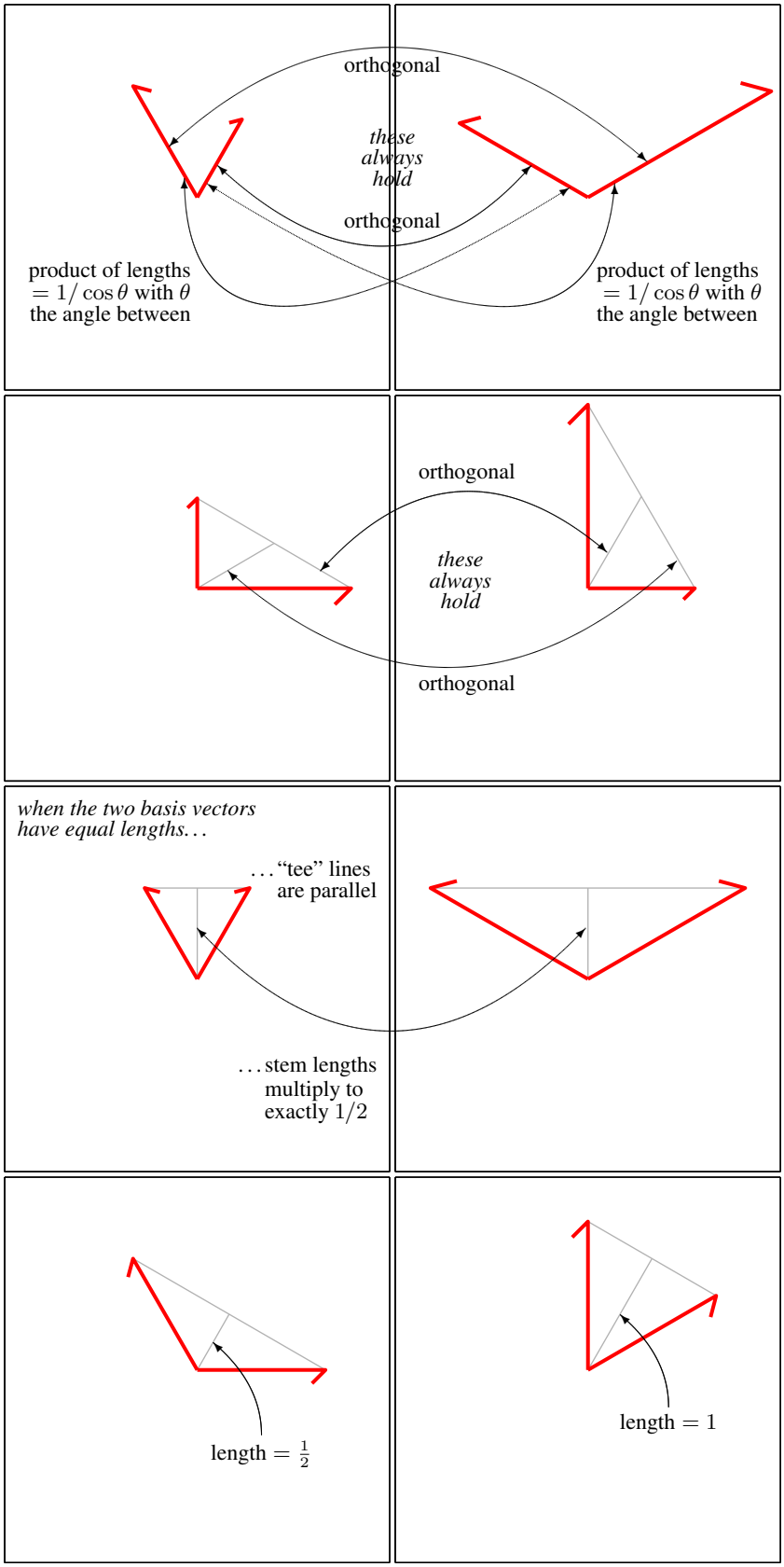


Fig. 2 — Properties that relate array-plane bases (sketched on the left) and their corresponding dual bases (sketched on the right). The most practically useful of these properties is the last:

- Always: $\mathbf{B}^{+}\mathbf{B} = \mathbf{I}$. Illustrated in the top row.
- Always: Spacing basis vectors by ϕ spaces dual-basis vectors by complementary angle $180^{\circ} - \phi$.
- Always: Drawing “tees” with tops connecting vector tips and with stems from the origin to top midpoints yields a basis stem/top orthogonal to the dual-basis top/stem. Illustrated in the second, third, and fourth rows.
- When basis vectors are orthogonal: lengths are reciprocal to the lengths of the corresponding dual-basis vectors. Illustrated in the second row.
- When basis vectors have identical lengths: dual-basis vectors also have identical (to each other) lengths, parallel stems, parallel tops, and a product of stem lengths of exactly $1/2$. Illustrated in the third and fourth rows.

- Once orientations of equal-length vectors are fixed, vectors in $\mathbf{B}$ and $\mathbf{B}^+$ have lengths inversely proportional to each other. Halving one doubles the other, and so forth.
- That number $1/2$ is dimensionless, so the units of the vectors in $\mathbf{B}$ and $\mathbf{B}^+$ must be inverses. In array work, these units are often length and inverse length respectively, representing position and—it turns out—spatial frequency, or those units can be normalized out so that all vectors are dimensionless. (We follow the latter approach below.)

Of course, identity (3) gives $\mathbf{B}^+$ one of the key properties of an inverse. If it were a true inverse, however, we'd also have $\mathbf{B}\mathbf{B}^+ = \mathbf{I}$, but in fact this cannot be true when $\mathbf{B}$ is nonsquare. We have to settle for a weaker statement.

Fifth useful property: $\mathbf{B}\mathbf{B}^+$ *acts like an identity matrix when multiplying array-plane vectors.*

To see this, suppose column three-vector $\mathbf{x}$ is in the array plane and write it in terms of the basis and a real column two-vector $\boldsymbol{w}$ of weights as $\mathbf{x} = \mathbf{B}\boldsymbol{w}$. We know this can be done, and in fact we know now how we could even solve for the weights, though we need not do so here. Instead, just substitute into the left side of what we want to prove and watch the right side fall out through the use of identity (3) above: $(\mathbf{B}\mathbf{B}^+)\mathbf{x} = (\mathbf{B}\mathbf{B}^+)(\mathbf{B}\boldsymbol{w}) = \mathbf{B}(\mathbf{B}^+\mathbf{B})\boldsymbol{w} = \mathbf{B}\boldsymbol{w} = \mathbf{x}$. It is very similar if we start with a row three-vector $\mathbf{k}$ expressed in the dual basis using a two-vector $\boldsymbol{f}$ of weights: $\mathbf{k} = \boldsymbol{f}\mathbf{B}^+$. Now $\mathbf{k}(\mathbf{B}\mathbf{B}^+) = (\boldsymbol{f}\mathbf{B}^+)(\mathbf{B}\mathbf{B}^+) = \boldsymbol{f}(\mathbf{B}^+\mathbf{B})\mathbf{B}^+ = \boldsymbol{f}\mathbf{B}^+ = \mathbf{k}$. It would be very reasonable to view $\mathbf{B}^+\mathbf{B}\mathbf{x} = \mathbf{x}$ and $\mathbf{k}\mathbf{B}^+\mathbf{B} = \mathbf{k}$ as dual properties.

2.2 Array-Plane Lattices and their Duals

In the discussion above an arbitrary array-plane column-vector position $\mathbf{p}$ often takes form $\mathbf{B}\boldsymbol{w}$ using a real two-vector $\boldsymbol{w}$ of weights and a 3×2 basis matrix $\mathbf{B}$ that has basis vectors as columns. Now let us consider a very special case, that of integer weights. The set of all possible integer-weighted linear combinations of a set of basis vectors is a *point lattice* or just a *lattice*. We write $\mathbf{B}\mathbb{Z}^2$ for the lattice comprising column vectors of the form $\mathbf{B}\boldsymbol{n}$ with $\boldsymbol{n}$ an arbitrary column two-vector of integers—recall that we write just $\boldsymbol{n} \in \mathbb{Z}^2$ to specify such $\boldsymbol{n}$ concisely. We write $\mathbb{Z}^2\mathbf{B}^+$ for the lattice comprising row vectors of the form $\boldsymbol{k}\mathbf{B}^+$ where $\boldsymbol{k} \in \mathbb{Z}^2$, that is, where $\boldsymbol{k}$ is an arbitrary row two-vector of integers. Let us term integer column two-vector $\boldsymbol{n}$ and integer row two-vector $\boldsymbol{k}$ the *index vectors* that specify lattice points $\mathbf{B}\boldsymbol{n}$ and $\boldsymbol{k}\mathbf{B}^+$. The lattices $\mathbf{B}\mathbb{Z}^2$ and $\mathbb{Z}^2\mathbf{B}^+$ constructed from dual bases are *dual lattices*.

The left and right columns of Fig. 3 show basis vectors and points of lattice $\mathbf{B}\mathbb{Z}^2$ and its dual $\mathbb{Z}^2\mathbf{B}^+$ respectively. Matrix $\mathbf{B}$ is the same for both top plots, so the bases shown in those plots are duals. A different matrix $\mathbf{B}$ was chosen for the bottom plots, but it is the same for both of them, making the bases shown in the bottom row duals as well. The lattices in the top and bottom rows are the same, even though the index vectors for corresponding points are not, so it is clear that a lattice basis is not unique. In fact, we could correctly claim that the lattices in the upper-left plot and the lower-right plot are duals even though the bases shown and used to construct them are not. Dual bases generate dual lattices, but dual lattices can be constructed from bases that are not dual.

The dual lattices shown are *hexagonal lattices*, lattices on equilateral-triangular grids. Hexagonal lattices with the nearest-neighbor distances shown, $1/\sqrt{3}$ on the left and 2 on the right, are classic in array work.

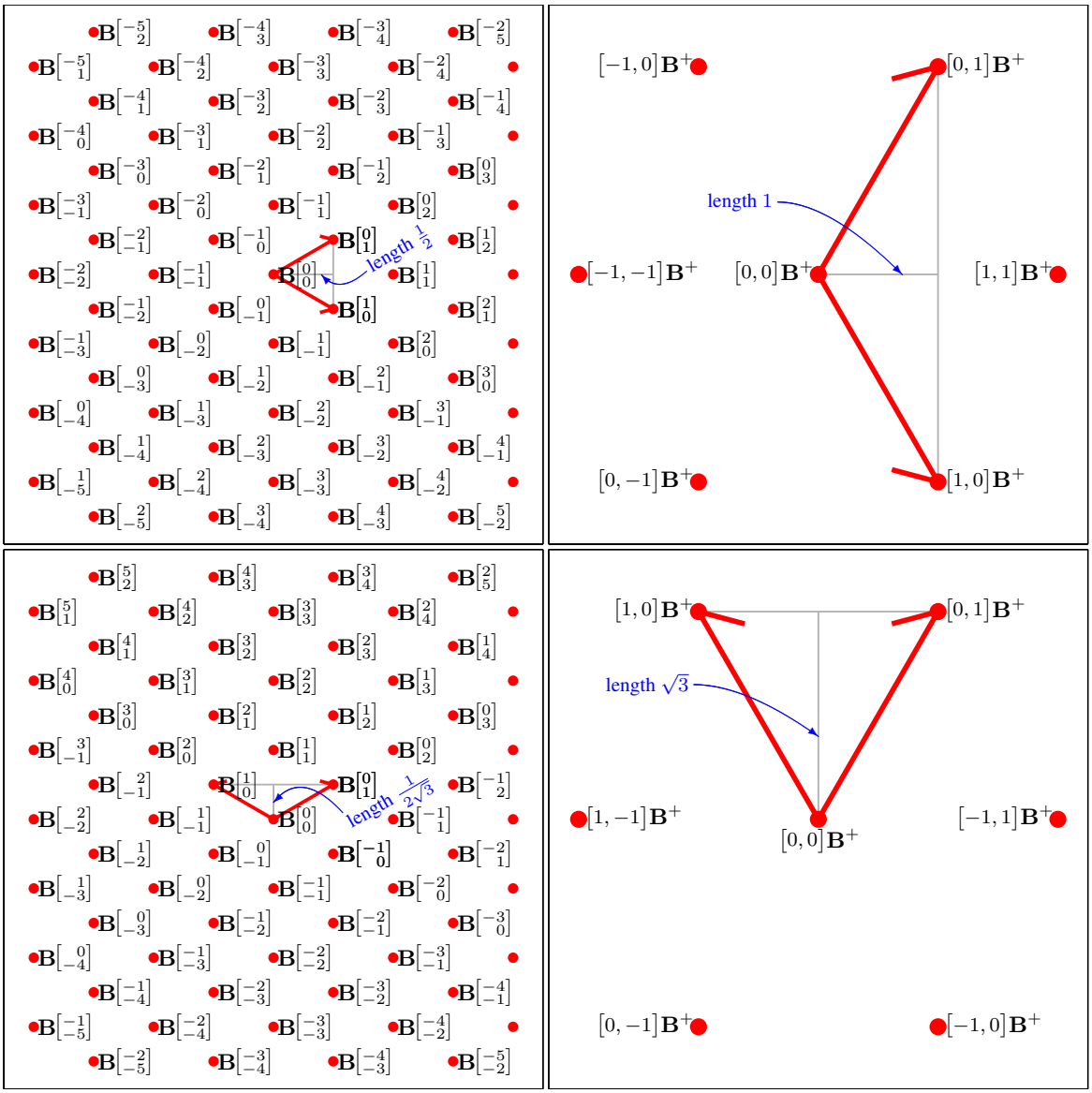


Fig. 3 — Left column: a hexagonal lattice $\mathbf{B}\mathbb{Z}^2$ with $1/\sqrt{3}$ nearest-neighbor spacing. Right column: the associated dual lattice $\mathbb{Z}^2\mathbf{B}^+$ is also hexagonal but with a nearest-neighbor spacing of 2. Different basis matrices $\mathbf{B}$ are used in the top and bottom rows, illustrating that bases are not unique and that dual lattices can be generated from nondual bases as well as from dual bases.

A basis of equal-length vectors separated by $60°$ has as its dual a basis of equal-length vectors separated by $120°$, but these basis-vector lengths produce "tee" stem lengths very different in the top and bottom rows. The top-row lengths of $1/2$ and 1 are easier to remember, so in this document that basis geometry is favored.

Given a basis matrix $\mathbf{B}$, the easiest way to check whether it generates one of the hexagonal lattices shown is to compute Gram matrix $\mathbf{B}^{\mathrm{T}}\mathbf{B}$. Its diagonal elements are the squared lengths of the basis vectors, and given that those are the same for the two vectors, the off-diagonal elements are that squared length times the cosine of the angle between the vectors. This works out nicely here because the cosines of $60°$ and $120°$ are $1/2$ and $-1/2$ respectively. The four bases in Fig. 3 then should have these four Gram matrices.

$$\begin{array}{ll} \text{Fig. 3}\\ \text{upper left} \end{array} \quad \mathbf{B}^{\mathrm{T}}\mathbf{B} = \frac{1}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \qquad \begin{array}{ll} \text{Fig. 3}\\ \text{upper right} \end{array} \quad \mathbf{B}^{+}\mathbf{B}^{+\mathrm{T}} = 2\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix},$$

$$\begin{array}{ll} \text{Fig. 3}\\ \text{lower left} \end{array} \quad \mathbf{B}^{\mathrm{T}}\mathbf{B} = \frac{1}{6}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}, \qquad \begin{array}{ll} \text{Fig. 3}\\ \text{lower right} \end{array} \quad \mathbf{B}^{+}\mathbf{B}^{+\mathrm{T}} = 2\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}.$$

Rectangular lattices with nearest-neighbor distances of $1/2$ and 2 are also common in array work and have bases and dual bases with gram matrices

$$\mathbf{B}^{\mathrm{T}}\mathbf{B} = \frac{1}{4}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{B}^{+}\mathbf{B}^{+\mathrm{T}} = 4\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

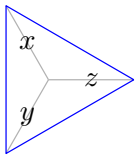
though rectangular lattices are not be considered further in this document.

One simple consistency check on the above matrices is provided by the fact that the Gram matrix $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ of a basis matrix $\mathbf{B}$ and the Gram matrix $\mathbf{B}^{+}\mathbf{B}^{+\mathrm{T}}$ of dual basis matrix $\mathbf{B}^{+}$ are always inverses because, substituting pseudoinverse definition (2) and canceling adjacent inverse matrices, $\mathbf{B}^{+}\mathbf{B}^{+\mathrm{T}} = \big((\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}}\big)\big(\mathbf{B}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\big) = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}$. For simplicity then, it is natural to write the Gram matrix of dual basis matrix $\mathbf{B}^{+}$ as just $(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}$.

2.2.1 Constructing a Basis for Array Work

Let us construct the matrix $\mathbf{B}$ with column basis vectors as shown in the upper-left plot of Fig. 3. We could pick any orientation for the array plane, but let us pick one that makes the construction simple. Visualize three-space unit vectors in the usual x, y, and z directions and notice that if we connect their tips and view it from a point far out in the $[1\ \ 1\ \ 1]$ direction, we see an equilateral triangle. We can therefore obtain a pair of vectors at the desired $60°$ angle by subtracting unit vectors, following which we can normalize to get the desired lengths. These steps yield

$$\begin{aligned} \mathbf{b}^1 &= \frac{1}{\sqrt{3}}\left(\frac{1}{\sqrt{2}}\left(\begin{bmatrix}0\\0\\1\end{bmatrix} - \begin{bmatrix}1\\0\\0\end{bmatrix}\right)\right), \\ \mathbf{b}^2 &= \frac{1}{\sqrt{3}}\left(\frac{1}{\sqrt{2}}\left(\begin{bmatrix}0\\0\\1\end{bmatrix} - \begin{bmatrix}0\\1\\0\end{bmatrix}\right)\right). \end{aligned} \qquad (4)$$



We can immediately assemble basis vectors $\mathbf{b}^1$ and $\mathbf{b}^2$ into basis matrix $\mathbf{B}$ and compute dual-basis matrix $\mathbf{B}^+$ from it:

$$\mathbf{B} = \frac{1}{\sqrt{6}}\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix}, \qquad \mathbf{B}^+ = \sqrt{2/3}\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \end{bmatrix}. \tag{5}$$

It is easy to check that $\mathbf{B}^+\mathbf{B} = \mathbf{I}$, that basis matrix $\mathbf{B}$ has Gram matrix $\mathbf{B}^{\mathrm{T}}\mathbf{B} = \frac{1}{6}\left[\begin{smallmatrix} 2 & 1 \\ 1 & 2 \end{smallmatrix}\right]$, and that dual-basis matrix $\mathbf{B}^+$ has Gram matrix $\mathbf{B}^+\mathbf{B}^{+\mathrm{T}} = 2\left[\begin{smallmatrix} 2 & -1 \\ -1 & 2 \end{smallmatrix}\right] = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}$ as required.

2.2.2 Plotting Axes: Azimuth and Elevation

The *boresight* vector is normal to the array plane and generally presumed in array discussions to be towards the horizon or be at some modest array "tilt back" angle above the horizon. Most often we view the array plane in diagrams from behind, looking in the boresight direction.

In construction (4) the geometry is viewed from far off in the $[1\ \ 1\ \ 1]$ direction, so we are actually looking in the $[-1\ \ -1\ \ -1]$ direction. We can see in (5) that all our basis vectors, the rows of $\mathbf{B}$ and the columns of $\mathbf{B}^+$, are orthogonal to—have zero dot products with—this vector, so here let us take the direction of $[-1\ \ -1\ \ -1]$ as the boresight direction.

The rightward and upward directions on the page in construction (4) are then named the *azimuth* and *elevation* directions simply because if we start with a boresight vector extending towards the horizon and rotate it towards vectors in those azimuth or elevation directions, we respectively increase our boresight vector's compass direction or angle above the horizon. We can formalize this by constructing unit vectors in the three orthogonal directions, azimuth, elevation, and boresight. In (4) the azimuth direction appears to be the z direction, but if we start with an arbitrary vector in that direction, for example $[0\ \ 0\ \ 3]$, it is not orthogonal to the boresight direction as required. We can correct this by adding a boresight component, in this case just $[-1\ \ -1\ \ -1]$, so that the total, $[-1\ \ -1\ \ \ 2]$, is indeed orthogonal to boresight. Similarly, the elevation direction appears in (4) to be in the direction of $[1\ \ -1\ \ \ 0]$, with equal components in the x and $-y$ directions, and we find this vector is orthogonal to boresight without adjustment. Let us now construct a matrix—call it $\mathbf{P}$ for plotting—with our three orthogonal vectors as columns and with adjusting scale factors included as a diagonal post-multiplying matrix to give each column unit length.

in azimuth direction ↓ ↓ in elevation direction

$$\mathbf{P} = \begin{bmatrix} -1 & 1 & -1 \\ -1 & -1 & -1 \\ 2 & 0 & -1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{6}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} \end{bmatrix}$$

in boresight direction ↑ — normalization to unit length

It is easy to verify that $\mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{P}\mathbf{P}^{\mathrm{T}} = \mathbf{I}$, making it an orthogonal matrix as intended. Now array-plane points like column vector $\mathbf{x} = \mathbf{B}\boldsymbol{n}$ or row vector $\mathbf{k} = \boldsymbol{f}\mathbf{B}^+$ can be projected onto the columns of $\mathbf{P}$ to obtain their az-el-boresight coordinates as $\mathbf{P}^{\mathrm{T}}\mathbf{x} = \mathbf{P}^{\mathrm{T}}\mathbf{B}\boldsymbol{n}$ or $\mathbf{k}\mathbf{P} = \boldsymbol{f}\mathbf{B}^+\mathbf{P}$ respectively. If we have done everything correctly, the last component, boresight, should be zero. The first and second coordinates, az and

el, are plotted on horizontal and vertical axes to make array-plane plots. If we wish, we can precompute the matrix products

$$\begin{aligned}
\mathbf{P}^{\mathrm{T}}\mathbf{B} &= \tfrac{1}{\sqrt{6}}\begin{bmatrix} \frac{1}{\sqrt{6}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} \end{bmatrix}\begin{bmatrix} -1 & -1 & 2 \\ 1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix}\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{6} & 0 & 0 \\ 0 & \frac{1}{2\sqrt{3}} & 0 \\ 0 & 0 & \frac{1}{3\sqrt{2}} \end{bmatrix}\begin{bmatrix} 3 & 3 \\ -1 & 1 \\ 0 & 0 \end{bmatrix} \\
&= \tfrac{1}{2}\begin{bmatrix} 1 & 1 \\ \frac{-1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ 0 & 0 \end{bmatrix}, \\
\mathbf{B}^{+}\mathbf{P} &= \sqrt{2/3}\begin{bmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} -1 & 1 & -1 \\ -1 & -1 & -1 \\ 2 & 0 & -1 \end{bmatrix}\begin{bmatrix} \frac{1}{\sqrt{6}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{3}} \end{bmatrix} \\
&= \begin{bmatrix} 3 & -3 & 0 \\ 3 & 3 & 0 \end{bmatrix}\begin{bmatrix} \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & \frac{2}{3} \end{bmatrix} \\
&= \begin{bmatrix} 1 & -\sqrt{3} & 0 \\ 1 & \sqrt{3} & 0 \end{bmatrix}.
\end{aligned}$$

Of course we can omit the boresight column from $\mathbf{P}$ if we wish and make $\mathbf{P}$ a 3×2 az-el plotting matrix only, making both $\mathbf{P}^{\mathrm{T}}\mathbf{B}$ and $\mathbf{B}^{+}\mathbf{P}$ into 2×2 matrices. When working with the hexagonal lattice, many authors avoid three-dimensional representations and work with two array-plane coordinates only, effectively using these 2×2 versions of $\mathbf{P}^{\mathrm{T}}\mathbf{B}$ and $\mathbf{B}^{+}\mathbf{P}$, which are (easily verified to be) inverses, as their basis and dual-basis matrices. At first glance it seems attractive to work in threespace for the naturalness with which the 60° and 120° angles can be obtained so that radicals are confined to scalar factors out front, rather than have radicals scattered across various elements of the 2×2 versions. But it actually makes no difference, because one plots az and el and so must compute azimuth and elevation for plotting using $\mathbf{P}^{\mathrm{T}}\mathbf{B}$ and $\mathbf{B}^{+}\mathbf{P}$ anyway, and in other contexts the $\mathbf{B}$ and $\mathbf{B}^{+}$ matrices most often simply cancel.

2.3 Sublattices and their Duals

What happens when we use points of a lattice as basis vectors to generate a new lattice?

Suppose we begin with the bases $\mathbf{B}$ and $\mathbf{B}^{+}$ and the lattices $\mathbf{B}\mathbb{Z}^2$ and $\mathbb{Z}^2\mathbf{B}^{+}$ they generate, exactly as pictured in the top row of Fig. 3, and choose two points of the lattice $\mathbf{B}\mathbb{Z}^2$ on the left to use as new basis vectors. The upper half of Fig. 4 shows one such choice, the specific points $\mathbf{B}\left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$ and $\mathbf{B}\left[\begin{smallmatrix}-1\\2\end{smallmatrix}\right]$. We can make these the columns of new basis matrix

$$\left[\mathbf{B}\begin{bmatrix}1\\1\end{bmatrix},\quad \mathbf{B}\begin{bmatrix}-1\\2\end{bmatrix}\right] = \mathbf{B}\begin{bmatrix}1 & -1\\ 1 & 2\end{bmatrix} = \mathbf{BR} \text{ with } \mathbf{R} = \begin{bmatrix}1 & -1\\ 1 & 2\end{bmatrix}.$$

Choosing points of the original lattice as new basis vectors gives us a new basis matrix $\mathbf{BR}$.

The lower-left plot in Fig. 4 shows the new lattice $\mathbf{BR}\mathbb{Z}^2$ generated by this new basis matrix $\mathbf{BR}$. A point in that new lattice is of the form $\mathbf{BR}\boldsymbol{n}$ with $\boldsymbol{n}\in\mathbb{Z}^2$, which can be looked at in two ways. It is of course basis matrix $\mathbf{BR}$ times $\boldsymbol{n}\in\mathbb{Z}^2$, which is what makes it a point of new lattice $\mathbf{BR}\mathbb{Z}^2$, but it is also

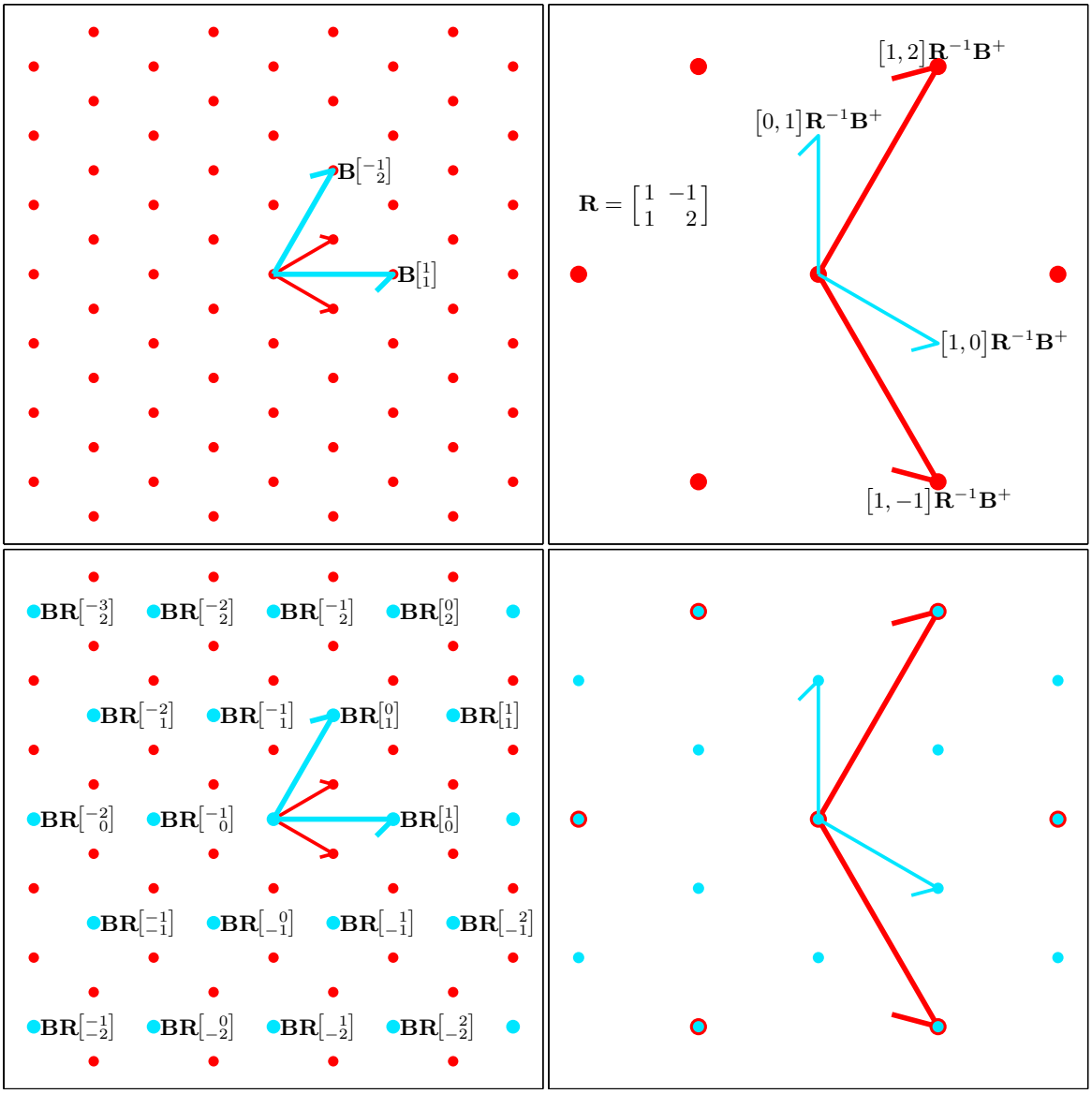


Fig. 4 — Top left: make $\mathbf{R}$ a nonsingular integer resampling matrix to make new basis vectors, the columns of $\mathbf{BR}$, be points of an existing lattice $\mathbf{B}\mathbb{Z}^2$. Top right: the original dual basis vectors are integer combinations of the new dual basis vectors, the rows of $\mathbf{R}^{-1}\mathbf{B}^{+}$. Bottom left: the new basis vectors generate a sublattice $\mathbf{BR}\mathbb{Z}^2$ of original lattice $\mathbf{B}\mathbb{Z}^2$. Bottom right: the dual of the new sublattice is a superlattice $\mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}^{+}$ of the original dual lattice $\mathbb{Z}^2\mathbf{B}^{+}$. Summary of the relationships:

$$\begin{array}{ccccc} \textit{original lattice} & \mathbf{B}\mathbb{Z}^2 & \overset{\text{dual}}{\longleftrightarrow} & \mathbb{Z}^2\mathbf{B}^{+} & \textit{dual of original lattice} \\ & \cup & \text{density ratio } |\mathbf{R}| & \cap & \\ \textit{sublattice of original lattice} & \mathbf{BR}\mathbb{Z}^2 & \longleftrightarrow & \mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}^{+} & \textit{dual of sublattice of original lattice} \end{array}$$

basis matrix $\mathbf{B}$ times $\mathbf{R}\boldsymbol{n} \in \mathbb{Z}^2$, which makes it simultaneously point of the original lattice $\mathbf{B}\mathbb{Z}^2$. In this way, every point in $\mathbf{BR}\mathbb{Z}^2$ is also in $\mathbf{B}\mathbb{Z}^2$, so we write $\mathbf{BR}\mathbb{Z}^2 \subset \mathbf{B}\mathbb{Z}^2$ and say the new lattice $\mathbf{BR}\mathbb{Z}^2$ is a *sublattice* of the original lattice $\mathbf{B}\mathbb{Z}^2$. This only works because $\mathbf{R}$ is a *resampling matrix*, a nonsingular square integer matrix. ("Nonsingular" here corresponds to the new basis vectors being linearly independent[2]) Of course in Fig. 4 we could have chosen completely different points for our new basis vectors in a way that yielded the same sublattice, so resampling matrices are not unique. Here $\mathbf{R} = \left[\begin{smallmatrix} 1 & -1 \\ 1 & 2 \end{smallmatrix}\right]$ as pictured, but $\mathbf{BR}$ with $\mathbf{R} = \left[\begin{smallmatrix} -1 & 2 \\ -1 & -1 \end{smallmatrix}\right]$, for example, would have generated the same sublattice.

What does the dual of a sublattice look like?

The (row) basis $\mathbf{B}^+$ and the lattice $\mathbb{Z}^2\mathbf{B}^+$ of row vectors it generates are duplicated from the upper right in Fig. 4 to the upper right in Fig. 3, where the dual of our new basis is then added. That dual is easily computed using a simple algebraic property of the pseudoinverse. Here the derivation of that property applies pseudoinverse definition (2) to $\mathbf{BR}$ in the first line, distributes transposes and inverses across products in the second and third lines, cancels inverses in the fourth line, and recognizes the right side of definition (2) at the end:

$$\begin{aligned}
(\mathbf{BR})^+ &= \left((\mathbf{BR})^{\mathrm{T}}\mathbf{BR}\right)^{-1}(\mathbf{BR})^{\mathrm{T}} \\
&= (\mathbf{R}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{BR})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} \\
&= \mathbf{R}^{-1}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{R}^{-\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}} \\
&= \mathbf{R}^{-1}(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}^{\mathrm{T}} \\
&= \mathbf{R}^{-1}\mathbf{B}^+. \qquad (6)
\end{aligned}$$

The similarity in form to the matrix-algebra identity $(\mathbf{UV})^{-1} = \mathbf{V}^{-1}\mathbf{U}^{-1}$ makes this easy to remember. The basis dual to our sublattice basis, the columns of $\mathbf{BR}$, comprises just the rows of $\mathbf{R}^{-1}\mathbf{B}^+$ or, using the Fig. 4 labels, $\begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{R}^{-1}\mathbf{B}^+$ and $\begin{bmatrix} 0 & 1 \end{bmatrix}\mathbf{R}^{-1}\mathbf{B}^+$.

Hidden within identity (6) is an important relationship. Look at the basis vectors of the original dual lattice. In Fig. 3, the upper and lower of these basis vectors were labeled $\begin{bmatrix} 0 & 1 \end{bmatrix}\mathbf{B}^+$ and $\begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{B}^+$ because these vectors are just the two rows of $\mathbf{B}^+$. But since $\mathbf{RR}^{-1} = \mathbf{I}$, we are free to insert $\mathbf{RR}^{-1}$ into these expressions anywhere that matrix-vector sizes are compatible. We can say, for example, that the two basis vectors are $\begin{bmatrix} 0 & 1 \end{bmatrix}\mathbf{RR}^{-1}\mathbf{B}^+$ and $\begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{RR}^{-1}\mathbf{B}^+$, revealing them to be $\begin{bmatrix} 0 & 1 \end{bmatrix}\mathbf{R} = \begin{bmatrix} 1 & -1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 0 \end{bmatrix}\mathbf{R} = \begin{bmatrix} 1 & 2 \end{bmatrix}$ times $\mathbf{R}^{-1}\mathbf{B}^+$, which is how they are labeled in Fig. 4. So in addition to being basis vectors of our original dual lattice, they are also points in some dual lattice $\mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}^+$ generated by dual basis $\mathbf{R}^{-1}\mathbf{B}^+$, the pseudoinverse and therefore dual of sublattice basis matrix $\mathbf{BR}$. We have discovered that $\mathbb{Z}^2\mathbf{B}^+ \subset \mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}^+$.

This new lattice $\mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}^+$ is shown on the lower right in Fig. 4. On the left, before we took duals, the new lattice was a sublattice of the old, but on the right, where we are looking at their duals, this is reversed. The old lattice is a sublattice of the new, or the new lattice is a *superlattice* of the old. The dual and sublattice relationships are summarized at the end of the Fig. 4 caption.

[2] We know that $\mathbf{BRw} = 0$ if and only if $\mathbf{Rw} = 0$, because the columns of $\mathbf{B}$ are linearly independent, and if we choose linearly independent new basis vectors, we also know that $\mathbf{BRw} = 0$ if and only if $\mathbf{w} = 0$. Combine the two statements and we have that $\mathbf{Rw} = 0$ if and only if $\mathbf{w} = 0$, which for a square $\mathbf{R}$ is just the statement that $\mathbf{R}$ is nonsingular.

Figure 4 also illustrates a convention used throughout this document. On the left a lattice is always plotted before its sublattice. As each is plotted, its dual is plotted on the right, where a lattice is therefore plotted before its superlattice. Larger dots are always used for less-dense lattices, so on the left larger sublattice dots obscure some points of the original lattice, forcing us to remember that the obscured points are there. There is no such difficulty on the right.

2.3.1 The Density of a Lattice in the Plane

How do we determine the density of a lattice, i.e., the number of its points per unit area in the plane?

Consider the example lattice on the left in Fig. 5, where lattice points are shown as enormous dots to simplify this discussion and where crosshatching through the lattice points has tiled the plane with parallelograms. The area of one of these parallelogram tiles is the *fundamental volume* of the lattice (where the term "volume" refers to the possible generalization to more than two dimensions). The four pieces of a dot in each parallelogram total to exactly one dot, so we can regard the tiles as unit cells and take the density of the lattice as the reciprocal of its fundamental volume.

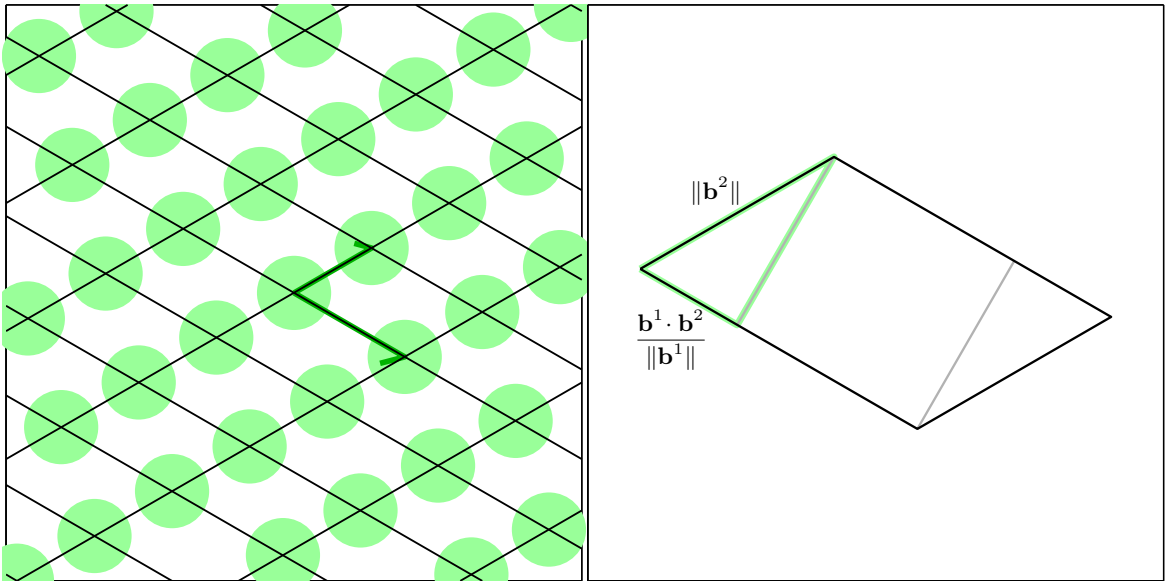


Fig. 5 — Left: division of the plane into unit cells of the lattice.
Right: The derivation of the fundamental volume of lattice $\mathbf{B}\mathbb{Z}^2$ as $|\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{1/2}$.

The basis vectors are sketched Fig. 5 with less than $90°$ between them, a property we depend on here in deriving the fundamental volume. If the actual angle is more than $90°$, just replace one of the basis vectors with its negative to make the angle between them less than $90°$. It becomes clear below that we can actually just forget about having done this step, because the final algebraic result is not affected by negating either basis vector.

To obtain the fundamental volume, consider the enlarged unit cell on the right. It is a parallelogram comprising a square and two triangles. If the unshaded triangle is clipped off, slid over, and joined to the

shaded triangle hypotenuse-to-hypotenuse, the original square and two triangles become one large rectangle. In the sketch the long side of the rectangle has length $\|\mathbf{b}^1\|$. The length of the short side of the rectangle is the same as that of the unlabeled side of the shaded right triangle shown. The hypotenuse of that triangle has length $\|\mathbf{b}^2\|$, and the short side has length that is just the dot product of $\mathbf{b}^2$ with a unit vector $\frac{1}{\|\mathbf{b}^1\|}\mathbf{b}^1$ in the direction of $\mathbf{b}^1$. The Pythagorean theorem gives the missing triangle side length as

$$\sqrt{\|\mathbf{b}^2\|^2 - \Big(\frac{\mathbf{b}^1 \cdot \mathbf{b}^2}{\|\mathbf{b}^1\|}\Big)^2}$$

so the large rectangle area, the area of the unit cell or fundamental volume, is

$$\|\mathbf{b}^1\|\sqrt{\|\mathbf{b}^2\|^2 - \Big(\frac{\mathbf{b}^1 \cdot \mathbf{b}^2}{\|\mathbf{b}^1\|}\Big)^2} = \Big(\big(\|\mathbf{b}^1\|^2\|\mathbf{b}^2\|^2 - \big(\mathbf{b}^1 \cdot \mathbf{b}^2\big)^2\big)\Big)^{1/2} = \big|\det(\mathbf{B}^{\mathrm{T}}\mathbf{B})\big|^{1/2}.$$

For brevity we ordinarily write this fundamental volume as $|\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{1/2}$ and the corresponding lattice density as $|\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{-1/2}$. The determinant is not affected by changing the sign of a basis vector, so this is a universal result when the basis vectors are columns of a matrix $\mathbf{B}$.

When the basis vectors are rows of a matrix, the transpose must go on the other factor to create the desired dot products and squared lengths, so the fundamental volume of dual lattice $\mathbb{Z}^2\mathbf{B}^+$ is $|\mathbf{B}^+\mathbf{B}^{+\mathrm{T}}|^{1/2}$. The determinant is applied to the Gram matrix of dual basis $\mathbf{B}^+$, and of course that Gram matrix can be written $(\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}$. The fundamental volume of dual lattice $\mathbb{Z}^2\mathbf{B}^+$ therefore is $|\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{-1/2}$, and this dual's density is $|\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{1/2}$. The lattice and its dual have reciprocal fundamental volumes and reciprocal densities, and the fundamental volume of one is the density of the other.

A sublattice $\mathbf{B}\mathbf{R}\mathbb{Z}^2$ has fundamental volume

$$|(\mathbf{B}\mathbf{R})^{\mathrm{T}}(\mathbf{B}\mathbf{R})|^{1/2} = |\mathbf{R}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{B}\mathbf{R}|^{1/2} = \big(|\mathbf{R}|\,|\mathbf{B}^{\mathrm{T}}\mathbf{B}|\,|\mathbf{R}|\big)^{1/2} = |\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{1/2}|\mathbf{R}|.$$

This is an integer factor $|\mathbf{R}|$ larger than the fundamental volume of lattice $\mathbf{B}\mathbb{Z}^2$, so sublattice $\mathbf{B}\mathbf{R}\mathbb{Z}^2$ has a density lower than that of lattice $\mathbf{B}\mathbb{Z}^2$ by a factor of $|\mathbf{R}|$.

Dual lattices have reciprocal fundamental volumes, so lattice $\mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}^+$, a superlattice of lattice $\mathbb{Z}^2\mathbf{B}^+$ and the dual of lattice $\mathbf{B}\mathbf{R}\mathbb{Z}^2$, has fundamental volume $|\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{-1/2}/|\mathbf{R}|$, a factor of $|\mathbf{R}|$ less than that of lattice $\mathbb{Z}^2\mathbf{B}^+$.

3. A SIGNAL-PROCESSING ENGINEER'S VIEW OF THE ELEMENT SYSTEM

Let's establish the array plane by supposing we have a normalized, dimensionless 3×2 basis matrix $\mathbf{B}$ such that any physical position in the array plane can be represented as a linear combination of the basis vector columns of matrix $\lambda\mathbf{B}$, where λ is some length-dimensioned scaling constant (not necessarily wavelength, as we shall see). We eventually discover rationales below for choosing both $\mathbf{B}$ and λ.

We can now develop a convenient characterization of the element outputs.

3.1 A Simple and Effectively Periodic Element System

As a thought experiment, suppose there is nothing in the universe besides incident electromagnetic fields (and something far, far away that creates them) and a receive antenna structure driving a receiver front end that in turn yields a single output. For simplicity suppose the structure uses only ordinary linear materials and that the electronics are LTI, i.e., just linear amplification and filtering. Now physically translate—slide without changing orientation—this entire system through space by some vector $\mathbf{x}$ and write the output signal as a function of that translation vector as $y(\mathsf{t}, \mathbf{x})$. Given two arbitrary field configurations such that

$$\begin{aligned} &\langle\text{incident field configuration 1}\rangle(\mathsf{t}, \mathbf{x}) && \text{yields} && y^1(\mathsf{t}, \mathbf{x}), \\ &\langle\text{incident field configuration 2}\rangle(\mathsf{t}, \mathbf{x}) && \text{yields} && y^2(\mathsf{t}, \mathbf{x}), \end{aligned}$$

then for arbitrary constants α and β,

$$\alpha\langle\text{field configuration 1}\rangle(\mathsf{t}, \mathbf{x}) + \beta\langle\text{field configuration 2}\rangle(\mathsf{t}, \mathbf{x}) \quad \text{yields} \quad \alpha y^1(\mathsf{t}, \mathbf{x}) + \beta y^2(\mathsf{t}, \mathbf{x})$$

just by the linearity of Maxwell's equations and the linearity of the front end, so the system is linear. In addition,

$$\langle\text{field configuration 1}\rangle(\mathsf{t} - \mathsf{t}', \mathbf{x}) \qquad \text{yields} \qquad y^1(\mathsf{t} - \mathsf{t}', \mathbf{x}).$$

for any time delay t', by the time invariance of Maxwell's equations and the front end. So the system is time invariant. But does

$$\langle\text{field configuration 1}\rangle(\mathsf{t}, \mathbf{x} - \mathbf{x}') \qquad \text{yield} \qquad y^1(\mathsf{t}, \mathbf{x} - \mathbf{x}'),$$

for arbitrary vector $\mathbf{x}'$? Is the system space-invariant? Indeed it is, because we defined the spatial argument of $y(\mathsf{t}, \mathbf{x})$ as the vector specifying the amount by which we slide the entire structure through space. If we move the fields east by one meter (by moving the far-away sources east one meter) and move the antenna structure east by one meter also, the output function of time is unchanged.

If we want to use a linear, time-invariant, space-invariant (LTSI) antenna and front-end system in a way that takes advantage of space invariance, what are we to do? We can't very well be constantly moving this antenna system all about, can we?

We almost can, in fact. What we can do is create what we pretend is a two-stage system. The first stage is linear, time-invariant, and space-invariant, and the second stage comprises spatial sampling at the points of lattice $\lambda\mathbf{B}\mathbb{Z}^2$. Now we only have to be able to translate the first-stage structure by an arbitrary $\mathbf{x} \in \lambda\mathbf{B}\mathbb{Z}^2$, because the second-stage sampler never asks to see the first-stage output for other translations. The one way to make this easy to do is to make the structure periodic. We make it invariant to translation by an arbitrary $\mathbf{x} \in \lambda\mathbf{B}\mathbb{Z}^2$ so that moving it changes nothing. We are free to pretend we moved it, as no one can ever know we didn't. It is like the moment in the famous mirror scene of the 1933 Marx Brothers' film *Duck Soup* in which Harpo, pretending to be Groucho's reflection in a nonexistent mirror, is caught off guard by Groucho suddenly spinning around and just pretends he spun as well, with Groucho none the wiser!

To create a periodic structure, in principle we just begin with some physical antenna-element structure that fits in a unit cell of lattice $\lambda\mathbf{B}\mathbb{Z}^2$. Let's suppose this structural cell has a single feedpoint at nominal location $\mathbf{x} = 0$, and let's suppose this feedpoint drives LTI front-end amplification and filtering that, in principle anyway, we can suppose is also contained in the unit cell. For convenience let's assume that this front-end processing includes enough buffering that its output can be observed, measured, passed on to other processing, and so forth, without affecting the fields.

Copying this unit-cell structure an infinite number of times, offsetting copies' physical positions from the original by amounts $\lambda\mathbf{B}\boldsymbol{n}$ with $\boldsymbol{n} \in \mathbb{Z}^2$, creates a periodic structure as desired. We can take the original, uncopied front-end output, the one nominally at feedpoint location $\mathbf{x} = 0$, as the output $y(\mathtt{t}, 0)$ of the unmoved first-stage system. Then we can write the second-stage output as $s^0_{\boldsymbol{n}}(\mathtt{t}) \triangleq y(\mathtt{t}, \lambda\mathbf{B}\boldsymbol{n})$ for sample index $\boldsymbol{n} \in \mathbb{Z}^2$. For simplicity we can just refer to this as the output of the element "at" physical position $\lambda\mathbf{B}\boldsymbol{n}$ or the element indexed by vector $\boldsymbol{n}$ or just element $\boldsymbol{n}$. Here the superscript zero is just an identifying flag referring to these processed element outputs. Going forward $\lambda\mathbf{B}\boldsymbol{n}$ and $\lambda\mathbf{B}\mathbb{Z}^2$ are respectively termed the *physical element position* and *physical element-position lattice*, and λ-normalized versions $\mathbf{B}\boldsymbol{n}$ and $\mathbf{B}\mathbb{Z}^2$ are just an *element position* and the *element-position lattice* respectively.

The only part of our thought experiment that is apparently impractical to approximate in practice is its infinite spatial extent. But if we design our back-end processing to use only a finite number of the element outputs, the remaining purpose of those unit cells with unused outputs is to provide a periodic electromagnetic environment for the unit cells with outputs we actually do use. This is important enough that in general we should not simply discard—fail to physically construct—unit cells with unused outputs. But we can provide the desired electromagnetic environment for the outputs used by keeping nearby unused cells, the antenna portions anyway, and replacing the missing electronics with equivalent termination impedances. These *guard elements* or *dummy elements* make our array seem infinite, at least as far as we can tell by inspecting the finite number of outputs actually available to us.

Finally, let's use the term *elements* from here forward for the combination of the physical antenna elements and any LTI processing that is either actually performed identically on all physical-element outputs or that can be referred to the feedpoints and incorporated into our notion of the elements for analysis purposes. This would naturally include any frequency responses associated with per-element front-end processing, but it can also include frequency responses associated with frequency conversion and/or DSP that is actually performed downstream, after front-end processing, as long as there is an equivalent front-end operation that can be included in the element model. This referral is more than a minor convenience, as the simplified approach to analysis taken below requires that certain frequency responses be included in the elements for proper array-system functioning to be easily seen.

3.2 Fourier Representation of the Element Outputs

3.2.1 2D Discrete-Space Fourier Transforms

Given some function $s(n_1, n_2)$ of integer indices n_1 and n_2, one certainly might treat n_2 as a fixed parameter and take a discrete-time—here it actually represents space—Fourier transform on n_1 to obtain, if we use f_1 as the frequency variable,

$$\mathrm{S}(f_1, n_2) = \sum_{n_1=-\infty}^{\infty} s(n_1, n_2)\, \mathrm{e}^{-\mathrm{j}2\pi f_1 n_1} \,.$$

There is nothing to prevent us from then proceeding to take f_1 as a fixed parameter and further transforming $\mathrm{s}(f_1, n_2)$ on n_2 using a new frequency variable f_2, resulting in, if we leave the infinite limits as understood,

$$S(f_1, f_2) = \sum_{n_2} \mathrm{s}(f_1, n_2)\, \mathrm{e}^{-\mathrm{j}2\pi f_2 n_2} = \sum_{n_2} \sum_{n_1} s(n_1, n_2)\, \mathrm{e}^{-\mathrm{j}2\pi(f_1 n_1 + f_2 n_2)}\,.$$

Except for the order of summation, which is irrelevant given practical restrictions on the function $s(n_1, n_2)$, the same result exactly would be obtained if we did the transforms in the other order. So we may as well define vectors $\boldsymbol{n} = \left[\begin{smallmatrix} n_1 \\ n_2 \end{smallmatrix}\right]$ and $\boldsymbol{f} = \begin{bmatrix} f_1 & f_2 \end{bmatrix}$ and write the whole thing using a vector notation as

$$S(\boldsymbol{f}) = \sum_{\boldsymbol{n}} s_{\boldsymbol{n}}\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}}\,. \tag{7}$$

In the interests of uncluttered brevity, in this document sums over integer column vectors or integer row vectors are by default over all elements of $\mathbb{Z}^2$ or $\mathbb{Z}^2$ respectively, so if $\boldsymbol{n}$ is used in the summand as a column two-vector of integers,

$$\sum_{\boldsymbol{n}} \text{ means } \sum_{\boldsymbol{n} \in \mathbb{Z}^2}, \text{ which is just } \sum_{n_1} \sum_{n_2} \text{ or } \sum_{n_2} \sum_{n_1} \text{ with } \boldsymbol{n} = \left[\begin{smallmatrix} n_1 \\ n_2 \end{smallmatrix}\right]$$

and if, as in later sections below, $\boldsymbol{k}$ is used in the summand as a row two-vector of integers,

$$\sum_{\boldsymbol{k}} \text{ means } \sum_{\boldsymbol{k} \in \mathbb{Z}^2}, \text{ which is just } \sum_{k_1} \sum_{k_2} \text{ or } \sum_{k_2} \sum_{k_1} \text{ with } \boldsymbol{k} = [k_1, k_2].$$

These sums are formally infinite but in reality finite, because only finitely many of their terms are ever nonzero. There are no convergence questions with finite sums, so their terms can be added in any order, and, consequently, their component sums on individual vector components can be ordered in any way desired.

The inverse transforms are handled similarly, except that integrals replace sums. Inverse transforms of $S(f_1, f_2)$ on frequency variables f_1 and f_2 can be done in either order. Inverse transforming on f_2 yields

$$\mathrm{s}(f_1, n_2) = \int_{\text{period}_2} S(f_1, f_2)\, \mathrm{e}^{\mathrm{j}2\pi f_2 n_2}\, df_2$$

where region of integration "period$_2$" is any one period of the integrand, for example a unit interval like $-1/2$ to $1/2$ or 0 to 1. Then inverse transforming on f_1 in an exactly similar way yields

$$s(n_1, n_2) = \int_{\text{period}_1} \mathrm{s}(f_1, n_2)\, \mathrm{e}^{\mathrm{j}2\pi f_1 n_1}\, df_1 = \int_{\text{period}_1} \int_{\text{period}_2} S(f_1, f_2)\, \mathrm{e}^{\mathrm{j}2\pi(f_1 n_1 + f_2 n_2)}\, df_2\, df_1.$$

Again we may as well use a vector notation and write

$$s_{\boldsymbol{n}} = \int_{\square} S(\boldsymbol{f})\, \mathrm{e}^{\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}}\, d\boldsymbol{f}, \tag{8}$$

where we adopt the convention that $d\boldsymbol{f}$ means differential area $df_2\, df_1$ and take the integral to be a double integral because two variables are actually integrated. Writing $\square$ for the region of integration here is intended as shorthand for any period in $\boldsymbol{f}$. This could be (but doesn't have to be) just a rectangle extending over a period in f_1 and a period in f_2. Certainly the simplest choice is a unit interval in each variable, which of course makes a square in the (f_1, f_2) or $\boldsymbol{f}$ plane.

Together (7) and (8) define 2D discrete-space Fourier transform pair $s_{\boldsymbol{n}} \leftrightarrow S(\boldsymbol{f})$.

3.2.2 The Element Outputs in 2D and 3D Frequency Domains

Inverse discrete-space Fourier transform

$$s^0_{\boldsymbol{n}}(\mathsf{t}) = \int_{\square} S^0(\boldsymbol{f}, \mathsf{t})\, \mathrm{e}^{\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}}\, d\boldsymbol{f} \tag{9}$$

expresses each element output as an integral in 2D normalized frequency $\boldsymbol{f} = \big[f_1, f_2\big]$ over a unit square, one period of the integrand. Because that integrand is periodic with the region of integration as a period, and because that region has unit area, inverse transform (9) can also be written as just the average value of the integrand, averaging over vector $\boldsymbol{f}$, i.e.,

$$s^0_{\boldsymbol{n}}(\mathsf{t}) = \operatorname*{avg}_{\boldsymbol{f}} \Big\{ S^0(\boldsymbol{f}, \mathsf{t})\, \mathrm{e}^{\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}} \Big\}.$$

This lets us do what amounts to a change of variable in the integral without fussing with determinants of Jacobians, because those determinants would simply deal with changes to the $1/\text{area}$ factor that expresses an average as an integral. Using an average keeps those messy factors swept under the mathematical rug.

Let us in fact use change of variable

$$\mathbf{k} = \boldsymbol{f}\mathbf{B}^{+}, \qquad\qquad \boldsymbol{f} = \mathbf{k}\mathbf{B} \tag{10}$$

to write

$$s^0_{\boldsymbol{n}}(\mathsf{t}) = \operatorname*{avg}_{\mathbf{k}} \Big\{ S^0(\mathbf{k}\mathbf{B}, \mathsf{t})\, \mathrm{e}^{\mathrm{j}2\pi \mathbf{k}\mathbf{B}\boldsymbol{n}} \Big\}$$

where of course the average is taken over the array plane only, as that is the range of $\mathbf{k}$ in transformations (10). We can then go just a little further and use an ordinary inverse Fourier transform in continuous time to write $S^0(\mathbf{k}\mathbf{B}, \mathsf{t}) = \int \mathrm{S}^0(\mathbf{k}\mathbf{B}, \mathsf{f})\, \mathrm{e}^{\mathrm{j}2\pi \mathsf{f}\mathsf{t}}\, d\mathsf{f}$ so that the above average becomes

$$s^0_{\boldsymbol{n}}(\mathsf{t}) = \operatorname*{avg}_{\mathbf{k}} \bigg\{ \int \mathrm{S}^0(\mathbf{k}\mathbf{B}, \mathsf{f})\, \mathrm{e}^{\mathrm{j}2\pi (\mathsf{f}\mathsf{t} + \mathbf{k}\mathbf{B}\boldsymbol{n})}\, d\mathsf{f} \bigg\}.$$

That this represents an LTSI system with output spatially sampled by evaluation at $\mathbf{x} = \lambda \mathbf{B}\boldsymbol{n}$ as discussed in Section 3.1 above can now be made explicit using two canceling factors of λ to rewrite the above as

$$s^0_{\boldsymbol{n}}(\mathsf{t}) = \operatorname*{avg}_{\mathbf{k}} \bigg\{ \int \mathrm{S}^0(\mathbf{k}\mathbf{B}, \mathsf{f})\, \mathrm{e}^{\mathrm{j}2\pi \left(\mathsf{f}\mathsf{t} + \frac{1}{\lambda}\mathbf{k}\mathbf{x}\right)}\, d\mathsf{f} \bigg\}_{\mathbf{x} = \lambda \mathbf{B}\boldsymbol{n}}. \tag{11}$$

The LTSI portion of this is interpreted in detail in the next section.

If embedded filtering were real with real outputs, then conjugate symmetry $\mathrm{S}^0(\mathbf{kB},\mathsf{f}) = (\mathrm{S}^0(-\mathbf{kB},-\mathsf{f}))^*$, which would put components averaged/integrated over into conjugate pairs, would be guaranteed. But we see in Section 3.4 below that we should eliminate the system's vulnerability to image beams by removing half of each such pair with filtering, and it's convenient to refer that filtering into the element system. With the balance between complex pairs thus disturbed, the element outputs $s^0_{\boldsymbol{n}}(\mathsf{t})$ can no longer be assumed real. They must be complex.

3.3 Physical Interpretation of the Element-Output Averaging Operation

3.3.1 A Complex Plane Wave

To interpret element-output representation (11) correctly, we need to take a short detour into the mathematical representation of electromagnetic plane waves. For mathematical convenience let us use this slightly unusual parameterization of a complex plane wave:

$$A\,\mathrm{e}^{\mathsf{j}2\pi\left(\mathsf{ft}+\frac{1}{c/|\mathsf{f}|}\widehat{u}\cdot\mathbf{x}\right)}\,. \tag{12}$$

That this is a propagating wave is not difficult to work out. If we let $\theta(\mathsf{t},\mathbf{x})$ be the phase of the complex exponential so that $\theta(\mathsf{t},\mathbf{x}) = 2\pi(\mathsf{ft} + \frac{\mathsf{f}}{c}\,\mathrm{sgn}(\mathsf{f})\widehat{u}\cdot\mathbf{x})$, we can predict $\Delta\mathsf{t}$ into the future by looking in the $\widehat{u}$ direction by Δx so that $\theta(\mathsf{t},\mathbf{x}+\widehat{u}\Delta x) = \theta(\mathsf{t}+\Delta\mathsf{t},\mathbf{x})$ if we set $\mathsf{f}\Delta\mathsf{t} = \frac{\mathsf{f}}{c}\,\mathrm{sgn}(\mathsf{f})\Delta x$ or, equivalently, by choosing $\Delta x = c\Delta\mathsf{t}\,\mathrm{sgn}(\mathsf{f})$. Certainly we could also include a component normal to $\widehat{u}$ in $\Delta\mathbf{x}$, but it would only be lost in the dot product. So including any such component would have us look further away to get the same information. When $\mathsf{f}>0$ then, $\Delta x/\Delta\mathsf{t} = c$ implies that the wave is coming from the $\widehat{u}$ direction at speed c. Likewise, $\Delta x/\Delta\mathsf{t} = -c$ when $\mathsf{f}<0$, and the wave is moving in the $\widehat{u}$ direction at speed c. Evolving through one full phase cycle requires $\Delta\mathsf{t} = 1/|\mathsf{f}|$ or $\Delta x = c/\mathsf{f}$, so the wavelength is $|\Delta x| = c/|\mathsf{f}|$.

A notationally modified version of complex plane wave (12) is more useful to us. Suppose we decompose vector $\frac{\lambda}{c/|\mathsf{f}|}\widehat{u}$ into a sum of two row vectors, an array-plane component $\mathbf{k}$ and a component $\mathbf{k}_\perp$ in the boresight direction. Replacing $\frac{1}{c/|\mathsf{f}|}\widehat{u}$ in (12) with $\frac{1}{\lambda}(\mathbf{k}+\mathbf{k}_\perp)$ yields a wave in the form

$$A\,\mathrm{e}^{\mathsf{j}\frac{2\pi}{\lambda}\mathbf{k}_\perp\mathbf{x}}\;\mathrm{e}^{\mathsf{j}2\pi\left(\mathsf{ft}+\frac{1}{\lambda}\mathbf{kx}\right)}\,. \tag{13}$$

The length of $\mathbf{k}$ is restricted of course, because $\|\mathbf{k}+\mathbf{k}_\perp\| = \lambda\|\frac{1}{\lambda}(\mathbf{k}+\mathbf{k}_\perp)\| = \lambda\|\frac{1}{c/|\mathsf{f}|}\widehat{u}\| = \frac{\lambda}{c/|\mathsf{f}|}$. The orthogonality of $\mathbf{k}$ and $\mathbf{k}_\perp$ then yields Pythagorean relationship

$$\|\mathbf{k}\|^2 + \|\mathbf{k}_\perp\|^2 = \|\mathbf{k}+\mathbf{k}_\perp\|^2 = \left(\frac{\lambda}{c/|\mathsf{f}|}\right)^2$$

from which it follows that $\|\mathbf{k}\| \le \frac{\lambda}{c/|\mathsf{f}|}$ with $\|\mathbf{k}_\perp\|$ fully determined by $\|\mathbf{k}\|$ and with $\mathbf{k}_\perp$ itself extending distance $\|\mathbf{k}_\perp\|$ along either the boresight direction or its opposite.

3.3.2 Our LTSI System Representation Refers to Plane Waves

Choose any value for vector $\mathbf{k}$ with $\|\mathbf{k}\| \leq \frac{\lambda}{c/|\mathsf{f}|}$. If $\|\mathbf{k}\| = \frac{\lambda}{c/|\mathsf{f}|}$, it follows that $\mathbf{k}_\perp = 0$. If $\|\mathbf{k}\| < \frac{\lambda}{c/|\mathsf{f}|}$ however, there are two possible $\mathbf{k}_\perp$ values, and they are negatives of each other. This $\mathbf{k}$ then is associated with either one or two possible incident electromagnetic plane waves. Such an incident electromagnetic wave has x, y, and z electric-field and magnetic-field components, and each of these six components takes the form of a conjugate pair of complex plane waves. One half of the pair takes form (13) with $\mathsf{f} > 0$, and, to make the sum of the two halves of the pair real, the second half is just the conjugate of the first. That conjugate also takes form (12) and simply has different A, f, $\mathbf{k}$, and $\mathbf{k}_\perp$ parameters. In particular, because we assume the first half of the conjugate pair had $\mathsf{f} > 0$, the second half must have $\mathsf{f} < 0$.

Consider the $\mathsf{f} < 0$ waves of these field components in more detail. There are 6 such waves if $\|\mathbf{k}\| = \frac{\lambda}{c/|\mathsf{f}|}$ and 12 of them if $\|\mathbf{k}\| < \frac{\lambda}{c/|\mathsf{f}|}$. If we use $i = 1, \ldots, i_{\max}$ with $i_{\max}$ either 6 or 12 to number them, we can say that the i^{th} such component wave takes the form

$$C_i(\mathbf{k}, \mathsf{f})\, \mathrm{e}^{\mathrm{j}2\pi\left(\mathsf{f}\mathsf{t} + \frac{1}{\lambda}\mathbf{k}\mathbf{x}\right)} \tag{14}$$

with one value of f and one value of $\mathbf{k}$ common to all such component waves and with complex amplitude $C_i(\mathbf{k}, \mathsf{f})$ corresponding to factor $A\, \mathrm{e}^{\mathrm{j}\frac{2\pi}{\lambda}\mathbf{k}_\perp\mathbf{x}}$ in complex wave (13). Among our $i_{\max}$ component waves, only these complex amplitudes differ, even having different units according as the component represented is an electric or magnetic field. The relationships among these complex amplitudes are determined by Maxwell's equations in the usual ways, but none of that concerns us here. We can more or less ignore that level of detail and just consider these basics, which are true no matter what the polarization or other properties of the physical wave might be.

Our LTSI system, being linear, responds to these components together as the sum of its responses to them separately.[1] Further, an LTSI system's response to a complex exponential in t and $\mathbf{x}$ is that same complex exponential scaled by a frequency response, here a function of temporal frequency f, spatial frequency $\mathbf{k}$, and of course the $\mathbf{k}_\perp$ and component-identification information associated with i. So if we let $G_i(\mathbf{k}, \mathsf{f})$ be that frequency response for the i^{th} component, our LTSI system's output becomes

$$\sum_{i=1}^{i_{\max}} G_i(\mathbf{k}, \mathsf{f})\, C_i(\mathbf{k}, \mathsf{f})\, \mathrm{e}^{\mathrm{j}2\pi\left(\mathsf{f}\mathsf{t} + \frac{1}{\lambda}\mathbf{k}\mathbf{x}\right)} \,. \tag{15}$$

We can just as well invoke linearity again (and, technically speaking, an appropriate form of bounded-input, bounded-output stability as well) and suppose that each incident field component actually comprises an integral combination of corresponding components from many such incident waves so that the output becomes

$$\operatorname*{avg}_{\mathbf{k}} \left\{ \int \sum_{i=1}^{i_{\max}} G_i(\mathbf{k}, \mathsf{f}) C_i(\mathbf{k}, \mathsf{f})\, \mathrm{e}^{\mathrm{j}2\pi\left(\mathsf{f}\mathsf{t} + \frac{1}{\lambda}\mathbf{k}\mathbf{x}\right)}\, d\mathsf{f} \right\}.$$

[1] Of course, Maxwell's equations tell us that the $i_{\max}$ complex amplitudes associated with one physical wave cannot be determined independently, and as a result we cannot unambiguously determine $i_{\max}$ separate response characteristics. But that does not really affect our argument here. There would certainly be no harm in reducing the number of terms to account for this lowering of rank, but the rest of the argument would be essentially the same either way.

Here f has been allowed to range over negative as well as positive values so that both halves of conjugate pairs of waves are included. Comparing this to the pre-sampling average in element-output representation (11), we see that for $\mathbf{k}$ with $\|\mathbf{k}\| \leq \frac{\lambda}{c/|\mathsf{f}|}$ we can interpret $\mathrm{S}^0(\mathbf{kB}, \mathsf{f})$ as $\sum_{i=1}^{i_{\max}} G_i(\mathbf{k}, \mathsf{f}) C_i(\mathbf{k}, \mathsf{f})$. Therefore, for such $\mathbf{k}$ the quantity $\mathrm{S}^0(\mathbf{kB}, \mathsf{f})$ characterizes the LTSI system's total response to all incident complex waves, of whatever polarization, that have frequency f and DOA parameterized by $\mathbf{k}$.

3.3.3 The Visible Region and Choice of λ

For a narrowband array it is natural to choose our normalizing distance λ as wavelength $c/|\mathsf{f}|$ so that $\frac{\lambda}{c/|\mathsf{f}|} = 1$, so that our $\mathbf{k}$ for plane waves is just the array-plane component of DOA unit vector $\widehat{u}$. This choice results in easy-to-remember bound $\|\mathbf{k}\| \leq 1$. For a wideband or tunable array, however, if λ is truly a constant, as is our intent, we cannot actually have λ equal to wavelength $c/|\mathsf{f}|$ for more than one frequency. Grating-lobe considerations discussed below in Section 5.1.3 make it convenient in this case to choose $\lambda = c/|\mathsf{f}|$ for $|\mathsf{f}|$ at the top end of the band of interest. At that frequency then, $\mathbf{k}$ becomes the array-plane component of $\widehat{u}$ and for plane waves ranges over the closed unit disk, the unit circle and the area within it. At lower frequencies our requirement that $\mathbf{k}$ be the array-plane component of $\frac{\lambda}{c/|\mathsf{f}|}\widehat{u}$ means $\mathbf{k}$ ranges over a smaller disk. In all cases the range of $\mathbf{k}$ corresponding to plane waves in element output (11) is called the *visible region* of the $\mathbf{k}$ plane or just *visible space*. The rest of the plane is then termed *invisible space*. A specific value of $\mathbf{k}$ is said to be visible or invisible according to the region in which it lies.

An invisible $\mathbf{k}$ by definition does not correspond to any plane wave. But if the element system is in the far field of all sources, all electromagnetic fields take the form of plane waves, and in the average in element output (11) these are fully accounted for by the portion of the region of integration that intersects with visible space. In a far-field setting the portion of the integration that intersects invisible space must not contribute to the average, so for invisible $\mathbf{k}$ complex amplitude $\mathrm{S}^0(\mathbf{kB}, \mathsf{f}) = 0$. More generally, invisible components of $\mathrm{S}^0(\mathbf{kB}, \mathsf{f})$ represent near-field, evanescent field components. In this report, however, far-field conditions as given and the possibility of near-field components is ignored.

3.4 Assume Analytic Filtering and Take the Element Output at the Origin as a Reference

General element-output form (11) reduces when $\boldsymbol{n} = 0$ to

$$s_0^0(\mathsf{t}) = \operatorname*{avg}_{\mathbf{k}} \left\{ \int \mathrm{S}^0(\mathbf{kB}, \mathsf{f})\, \mathrm{e}^{\mathrm{j}2\pi \mathsf{f}\mathsf{t}}\, d\mathsf{f} \right\} \tag{16}$$

$$= \operatorname*{avg}_{\mathbf{k}} \left\{ S^0(\mathbf{kB}, \mathsf{t}) \right\} \tag{17}$$

where an ordinary inverse Fourier transform has been replaced by an appropriate function of t. From here on let us assume that the filtering referred into the elements from downstream signal processing includes analytic or positive-pass filtering. This lets us assume that the integrand in (16) is effectively zero for $\mathsf{f} < 0$. Such an assumption is most convenient because it resolves the direction of arrival/propagation question unambiguously: $\mathbf{k}$ is always the array-plane component of $\frac{\lambda}{c/|\mathsf{f}|}\widehat{u}$, which points in the DOA, and never its negative. Equation (17) then simply expresses the output of the element at the origin as an integral combination of components originating from different directions, with DOA parameterized by $\mathbf{k}$.

3.5 Relating $\mathbf{k}$ to the Wavenumber Vector

Readers from the electromagnetics community might naturally wish to relate their familiar notations to what is used here. This section is effectively an aside to address that and can be skipped without harm.

In electromagnetics a complex plane wave is generally written $A\,\mathrm{e}^{\mathrm{j}(\omega \mathsf{t} - \mathbf{k}_{\mathrm{ec}}\cdot\mathbf{x})}$ with radian frequency $\omega > 0$. The "ec" subscript is simply to distinguish their $\mathbf{k}$ from the one used in this report. The negative-frequency wave of a conjugate pair is seldom written explicitly but is instead generated automatically by the conjugation involved in taking the real part with $\mathrm{Re}\{z\} = (z + z^*)/2$:

$$\mathrm{Re}\big\{A\,\mathrm{e}^{\mathrm{j}(\omega \mathsf{t} - \mathbf{k}_{\mathrm{ec}}\cdot\mathbf{x})}\big\} = \tfrac{A}{2}\,\mathrm{e}^{\mathrm{j}(\omega \mathsf{t} - \mathbf{k}_{\mathrm{ec}}\cdot\mathbf{x})} + \tfrac{A^*}{2}\mathrm{e}^{-\mathrm{j}(\omega \mathsf{t} - \mathbf{k}_{\mathrm{ec}}\cdot\mathbf{x})}\,.$$

The two complex waves are expressed in terms of the same two parameters ω and $\mathbf{k}_{\mathrm{ec}}$ but differ by a sign on the entire exponent. By contrast, in the Section 3.3.2 discussion above the two complex waves of a pair presumably look alike algebraically but simply use different f and $\mathbf{k}$ values. Their complex amplitudes also differ from those used here by a factor of two, but this is really irrelevant since we never refer to those amplitudes alone in the mathematics but only to products representing element output signals.

More important is how direction of arrival (DOA) is parameterized in the two notational systems. For simplicity let's assume $\mathsf{f} > 0$ and compare the positive-frequency waves. Their wave above matches complex plane wave (12) when $\omega = 2\pi\mathsf{f}$ and $\mathbf{k}_{\mathrm{ec}} = -\frac{2\pi}{c/\mathsf{f}}\,\widehat{u}$, so that their $\mathbf{k}_{\mathrm{ec}}$ points in the direction of propagation and has length equal to 2π over wavelength. For $f > 0$ our $\mathbf{k}$ is the array-plane component of $\frac{\lambda}{c/\mathsf{f}}\,\widehat{u}$, which makes it the array-plane component of $-\frac{\lambda}{2\pi}\mathbf{k}_{\mathrm{ec}}$. Our DOA parameter $\mathbf{k}$ can be obtained from their radian wavenumber vector $\mathbf{k}_{\mathrm{ec}}$ by

1. multiplying by $\frac{1}{2\pi}$ to change its units from radians per unit length to cycles per unit length,
2. negating it to change its direction from that of propagation to that of arrival,
3. multiplying it by normalizing constant λ to make it a unit vector at the top of the signal band, and
4. taking the array-plane component.

These vectors can be related to the orientation of the array using the orthonormal plotting matrix $\mathbf{P}$ constructed above in Section 2.2.2 with unit-vector columns in azimuth, elevation, and boresight directions. The antenna community typically uses *direction cosines* u, v, and w to denote components of DOA unit vector $\widehat{u}$ in the azimuth, elevation, and boresight directions respectively. Using those variables, and remembering that $\mathbf{k}$ is a row vector while $\mathbf{k}_{\mathrm{ec}}$ is typically written as a column vector,

$$\widehat{u} = \mathbf{P}\begin{bmatrix} u \\ v \\ w \end{bmatrix}, \qquad \mathbf{k}_{\mathrm{ec}} = -\tfrac{2\pi}{c/\mathsf{f}}\mathbf{P}\begin{bmatrix} u \\ v \\ w \end{bmatrix}, \qquad \mathbf{k} = \tfrac{\lambda}{c/\mathsf{f}}\begin{bmatrix} u & v & 0 \end{bmatrix}\mathbf{P}^{\mathrm{T}}.$$

For planar arrays the electromagnetics community typically plots "in sine space" by letting horizontal and vertical axes represent u and v respectively and using using either a color scale or contours to represent DOA-dependent function values. Because $\widehat{u}$ is a unit vector, w is implied by u and v and need not be shown. In this report plots assume $f > 0$, in respect of the analytic-filtering assumption above in Section 3.4, and

horizontal and vertical axes represent the azimuth and elevation components of $\mathbf{k}$ respectively, in other words the first two components of $\mathbf{kP} = \frac{\lambda}{c/\mathtt{f}}\begin{bmatrix} u & v & 0 \end{bmatrix}$, with color then representing function values. Therefore, in the usual case of a narrowband array with λ set to $c/|f|$, our plots are the same as those of the antenna community.

4. MULTIDIMENSIONAL DSP BASICS

We need very little from multidimensional DSP actually, just a very few basics.

4.1 Three Operators and a Frequency-Referral Identity

The element output indexed by $\boldsymbol{n}$ is written everywhere above as $s^0_{\boldsymbol{n}}(\mathtt{t})$, but beginning here it is just $s^0_{\boldsymbol{n}}$. Our processing from here on is purely spatial, so even though all of our signals remain time dependent, the $\mathtt{t}$ dependence is omitted from the notation. Just s^0, with no subscript, refers to the entire signal, the function of $\boldsymbol{n}$ with all its many spatial samples, and that concise approach is used for other discrete-space signals as well.

Three simple discrete-space operators are central to phased arrays. The first is of course quite standard. The other two are not, but they are most convenient when analyzing systems with phase-shift steering.

convolution $s \star h$ of discrete-space quantities s and h	$(\ s \star h\ )_{\boldsymbol{n}} \triangleq \sum_{\boldsymbol{m}} s_{\boldsymbol{m}}\, h_{\boldsymbol{n}-\boldsymbol{m}} = \sum_{\boldsymbol{m}} s_{\boldsymbol{n}-\boldsymbol{m}}\, h_{\boldsymbol{m}}$	(18)
phase delay $s \curvearrowright \boldsymbol{f}_\Delta$ of signal s by phase gradient $\boldsymbol{f}_\Delta$	$(s \curvearrowright \boldsymbol{f}_\Delta)_{\boldsymbol{n}} \triangleq s_{\boldsymbol{n}}\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}_\Delta \boldsymbol{n}}$	(19)
steering $\boldsymbol{f}_\Delta \curvearrowleft h$ of impulse response h by frequency $\boldsymbol{f}_\Delta$	$(\boldsymbol{f}_\Delta \curvearrowleft h)_{\boldsymbol{n}} \triangleq \mathrm{e}^{\mathrm{j}2\pi \boldsymbol{f}_\Delta \boldsymbol{n}}\, h_{\boldsymbol{n}}$	(20)

Our order-of-operations convention is that the three operations above are done after multiplication and division but before addition and subtraction. Unparenthesized sequences of $\star$ and $\curvearrowright$ operations are evaluated left to right, so that $s \curvearrowright \boldsymbol{f}_\Delta \star h$ means $(s \curvearrowright \boldsymbol{f}_\Delta) \star h$, and unparenthesized sequences of $\star$ and $\curvearrowleft$ operations are evaluated—this is not standard but is most convenient—right to left, so that $h^1 \star \boldsymbol{f}_\Delta \curvearrowleft h^2$ means $h^1 \star (\boldsymbol{f}_\Delta \curvearrowleft h^2)$.

Recall that when ordinary filtering after a frequency conversion is to be replaced by filtering before the conversion, the filter's frequency response has to be modified appropriately. We can derive the equivalent idea for spatial processing using very simple steps:

$$
\begin{aligned}
(s \curvearrowright \boldsymbol{f}_\Delta \star h)_{\boldsymbol{n}} &= \big((s \curvearrowright \boldsymbol{f}_\Delta) \star h\big)_{\boldsymbol{n}} && \text{(order of operations)} \\
&= \sum_{\boldsymbol{m}} (s \curvearrowright \boldsymbol{f}_\Delta)_{\boldsymbol{n}-\boldsymbol{m}}\, h_{\boldsymbol{m}} && \text{(convolution definition (18))} \\
&= \sum_{\boldsymbol{m}} s_{\boldsymbol{n}-\boldsymbol{m}}\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}_\Delta (\boldsymbol{n}-\boldsymbol{m})}\, h_{\boldsymbol{m}} && \text{(phase-delay definition (19))}
\end{aligned}
$$

$$
\begin{aligned}
&= \Big(\sum_{\boldsymbol{m}} s_{\boldsymbol{n}-\boldsymbol{m}}\, \mathrm{e}^{\mathrm{j}2\pi \boldsymbol{f}_\Delta \boldsymbol{m}}\, h_{\boldsymbol{m}}\Big)\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}_\Delta \boldsymbol{n}} && \text{(split the complex exponential)}\\
&= \Big(\sum_{\boldsymbol{m}} s_{\boldsymbol{n}-\boldsymbol{m}}\, (\boldsymbol{f}_\Delta \curvearrowleft h)_{\boldsymbol{m}}\Big)\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}_\Delta \boldsymbol{n}} && \text{(steering definition (20))}\\
&= \Big(s \star (\boldsymbol{f}_\Delta \curvearrowleft h)\Big)_{\boldsymbol{n}}\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}_\Delta \boldsymbol{n}} && \text{(convolution definition (18))}\\
&= \big(s \star (\boldsymbol{f}_\Delta \curvearrowleft h) \curvearrowright \boldsymbol{f}_\Delta\big)_{\boldsymbol{n}} && \text{(phase-delay definition (19)).}
\end{aligned}
$$

This *frequency-referral identity* takes an easy-to-remember form when written using whole-signal notation with one line above the other as

$$
\begin{aligned}
& s \curvearrowright \boldsymbol{f}_\Delta \star h\\
&= s \star (\boldsymbol{f}_\Delta \curvearrowleft h) \curvearrowright \boldsymbol{f}_\Delta.
\end{aligned}
\tag{21}
$$

The mnemonic convenience of having the operands in the same order on the page on both sides is the real reason for the nonstandard right-to-left convention adopted for sequences containing the $\curvearrowleft$ operation.

4.2 Three Simple Fourier Properties

Four simple Fourier properties are important in this report. Each is analogous to a common 1D Fourier property, but they are proved here for the 2D case as a review for the rusty and to increase the comfort level of readers new to multidimensional transforms.

Average of transform: *Given Fourier pair $s_{\boldsymbol{n}} \leftrightarrow S(\boldsymbol{f})$, the average of the transform is just the value of the sample at the origin:*

$$
s_0 = \int_{\square} S(\boldsymbol{f})\, d\boldsymbol{f} = \operatorname*{avg}_{\boldsymbol{f}} \big\{ S(\boldsymbol{f}) \big\},
$$

This is just inverse-transform definition (8) with $\boldsymbol{n} = 0$ and with the integral re-expressed as an average.

Convolution property: *Space-domain convolution corresponds to frequency-domain multiplication, so that $x_{\boldsymbol{n}} \leftrightarrow X(\boldsymbol{f})$ and $y_{\boldsymbol{n}} \leftrightarrow Y(\boldsymbol{f})$ imply $(x \star y)_{\boldsymbol{n}} \leftrightarrow X(\boldsymbol{f})\, Y(\boldsymbol{f})$.*

Fourier-transform definition (7) and convolution definition (18) applied to $z_{\boldsymbol{n}} = (x \star y)_{\boldsymbol{n}}$ together yield

$$
\begin{aligned}
Z(\boldsymbol{f}) &= \sum_{\boldsymbol{n}} (x \star y)_{\boldsymbol{n}}\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f} \boldsymbol{n}}\\
&= \sum_{\boldsymbol{n}} \sum_{\boldsymbol{m}} x_{\boldsymbol{m}}\, y_{\boldsymbol{n}-\boldsymbol{m}}\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f} \boldsymbol{n}}\,.
\end{aligned}
$$

Multiplying the summand on the right by unity, in the form of canceling product $\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{m}}\,\mathrm{e}^{\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{m}}$ of complex exponentials, and then rearranging,

$$\begin{aligned} Z(\boldsymbol{f}) &= \sum_{\boldsymbol{n}}\sum_{\boldsymbol{m}} x_{\boldsymbol{m}}\, y_{\boldsymbol{n}-\boldsymbol{m}} \left(\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{m}}\,\mathrm{e}^{\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{m}}\right) \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}} \\ &= \sum_{\boldsymbol{m}} x_{\boldsymbol{m}}\,\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{m}} \sum_{\boldsymbol{n}} y_{\boldsymbol{n}-\boldsymbol{m}}\,\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}(\boldsymbol{n}-\boldsymbol{m})}\,. \end{aligned}$$

Let $\boldsymbol{n}' = \boldsymbol{n} - \boldsymbol{m}$ in the inner sum. Summing over all $\boldsymbol{n}$ and summing over all $\boldsymbol{n}'$ are equivalent, as the same terms get summed, so

$$Z(\boldsymbol{f}) = \sum_{\boldsymbol{m}} x_{\boldsymbol{m}}\,\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{m}} \sum_{\boldsymbol{n}'} y_{\boldsymbol{n}'}\,\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}'} = X(\boldsymbol{f})\,Y(\boldsymbol{f}).$$

Frequency shift: *Fourier pair $h_{\boldsymbol{n}} \leftrightarrow H(\boldsymbol{f})$ implies Fourier pair $(\boldsymbol{f}_\Delta \curvearrowleft h)_{\boldsymbol{n}} \leftrightarrow H(\boldsymbol{f}-\boldsymbol{f}_\Delta)$.*

Fourier-transform definition (7) gives $H(\boldsymbol{f}) = \sum_{\boldsymbol{n}} h_{\boldsymbol{n}}\,\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}}$. Replace $\boldsymbol{f}$ with $\boldsymbol{f}-\boldsymbol{f}_\Delta$ to obtain

$$\begin{aligned} H(\boldsymbol{f}-\boldsymbol{f}_\Delta) &= \sum_{\boldsymbol{n}} h_{\boldsymbol{n}}\,\mathrm{e}^{-\mathrm{j}2\pi (\boldsymbol{f}-\boldsymbol{f}_\Delta)\boldsymbol{n}} \\ &= \sum_{\boldsymbol{n}} \left(h_{\boldsymbol{n}}\,\mathrm{e}^{\mathrm{j}2\pi \boldsymbol{f}_\Delta \boldsymbol{n}}\right) \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}} \\ &= \sum_{\boldsymbol{n}} (\boldsymbol{f}_\Delta \curvearrowleft h)_{\boldsymbol{n}}\,\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}}, \end{aligned}$$

using steering definition (20) at the end. The last line is just Fourier-transform definition (7) applied to $\boldsymbol{f}_\Delta \curvearrowleft h$.

Periodicity: *Every 2D discrete-space Fourier transform $H(\boldsymbol{f})$ is periodic in the sense that $H(\boldsymbol{f}) = H(\boldsymbol{f}-\boldsymbol{k})$ for all $\boldsymbol{k} \in \mathbb{Z}^2$.*

Just replace $\boldsymbol{f}$ with $\boldsymbol{f}-\boldsymbol{k}$ in Fourier-transform definition $H(\boldsymbol{f}) = \sum_{\boldsymbol{n}} h_{\boldsymbol{n}}\,\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}}$ to see that

$$\begin{aligned} H(\boldsymbol{f}-\boldsymbol{k}) &= \sum_{\boldsymbol{n}} h_{\boldsymbol{n}}\,\mathrm{e}^{-\mathrm{j}2\pi (\boldsymbol{f}-\boldsymbol{k})\boldsymbol{n}} \\ &= \sum_{\boldsymbol{n}} h_{\boldsymbol{n}}\,\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}\boldsymbol{n}}\,\cancelto{1}{\mathrm{e}^{\mathrm{j}2\pi \boldsymbol{k}\boldsymbol{n}}} \\ &= H(\boldsymbol{f}) \end{aligned}$$

using the fact that $\mathrm{e}^{\mathrm{j}2\pi\langle\text{any integer}\rangle} = 1$ and that inner product $\boldsymbol{k}\boldsymbol{n}$ is an integer because the elements of vectors $\boldsymbol{k}$ and $\boldsymbol{n}$ are integers.

5. CREATING AND STEERING BEAMS

5.1 The Single-Beam Output

5.1.1 Array Steering, Weighting, and Combining Creates a Beam

Assuming for the moment that only a single beam is required, the array output is created from the element outputs using a row two-vector phase gradient $\boldsymbol{f}_\Delta$ and using combining weights $h_{\boldsymbol{n}}$ that here take the form of a discrete-space impulse response. We represent that array output here as the $\boldsymbol{n} = 0$ sample of a fictional signal $s_{\boldsymbol{n}}$, fictional because no other sample is ever realized. Here the first line specifies the array output exactly as it is ultimately implemented from definitions (18) and (19), and the other lines begin the analysis of why it works:

$$\begin{aligned} s_0 &= \big(s^0 \curvearrowright \boldsymbol{f}_\Delta \star\ h\big)_0 \qquad (22)\\ &= \big(s^0 \star (\boldsymbol{f}_\Delta \curvearrowleft h) \curvearrowright \boldsymbol{f}_\Delta\big)_0 \\ &= \big(s^0 \star (\boldsymbol{f}_\Delta \curvearrowleft h)\big)_0. \qquad (23) \end{aligned}$$

The second equality follows from frequency-referral identity (21), and the third equality just recognizes that, by delay definition (19), a phase delay leaves the zero sample unchanged.

Next we take Fourier transforms. Our frequency-domain notation assumes Fourier pairs

$$\begin{aligned} s^0_{\boldsymbol{n}} &\leftrightarrow S^0(\boldsymbol{f}),\\ h_{\boldsymbol{n}} &\leftrightarrow\ H(\boldsymbol{f}) \end{aligned}$$

and the frequency-shift, convolution, and average-of-transform properties of Section 4.2 above then respectively yield

$$\begin{aligned} (\boldsymbol{f}_\Delta \curvearrowleft h)_{\boldsymbol{n}} \ &\leftrightarrow\ H(\boldsymbol{f} - \boldsymbol{f}_\Delta),\\ \big(s^0 \star (\boldsymbol{f}_\Delta \curvearrowleft h)\big)_{\boldsymbol{n}} &\leftrightarrow\ S^0(\boldsymbol{f})\, H(\boldsymbol{f} - \boldsymbol{f}_\Delta),\\ \big(s^0 \star (\boldsymbol{f}_\Delta \curvearrowleft h)\big)_0 \ &= \operatorname*{avg}_{\boldsymbol{f}} \big\{ S^0(\boldsymbol{f})\, H(\boldsymbol{f} - \boldsymbol{f}_\Delta) \big\}. \end{aligned}$$

The last of these allows us to rewrite array output (23) with DOA parameterized either by $\boldsymbol{f}$ or by $\mathbf{k}$ as defined in transformation (10). In the latter case we also need analogous relationships

$$\mathbf{k}_\Delta = \boldsymbol{f}_\Delta \mathbf{B}^+, \qquad\qquad \boldsymbol{f}_\Delta = \mathbf{k}_\Delta \mathbf{B}. \qquad (24)$$

The two forms of our result then are

$$\begin{aligned} s_0 &= \operatorname*{avg}_{\boldsymbol{f}} \big\{ S(\ \boldsymbol{f}\ )\, H(\ \boldsymbol{f} - \boldsymbol{f}_\Delta\quad) \big\}\\ &= \operatorname*{avg}_{\mathbf{k}} \big\{ S(\mathbf{k}\mathbf{B})\, H((\mathbf{k} - \mathbf{k}_\Delta)\mathbf{B}) \big\}. \qquad (25) \end{aligned}$$

The latter is just zero-element output $s_0^0(\mathtt{t})$ from (17) but now with $H\big((\mathbf{k}-\mathbf{k}_\Delta)\mathbf{B}\big)$, termed the *array factor*, scaling the net complex amplitude $S(\mathbf{kB})$ of any signal components arriving from the DOA specified by $\mathbf{k}$. *Unsteered array factor* $H(\mathbf{kB})$ is designed to have a narrow beam in the $\mathbf{k}=0$ boresight direction, and the beam is then steered by setting $\mathbf{k}_\Delta$ to the value of DOA parameter $\mathbf{k}$ where the beam is desired. For this reason array-plane three-vector $\mathbf{k}_\Delta$ might reasonably be termed a *steering vector*.

5.1.2 Array-Factor Periodicity

Array factors are periodic. To see this, begin with unsteered array factor $H(\mathbf{kB})$ and offset DOA parameter $\mathbf{k}$ by an arbitrary point from *array-factor periodicity lattice* $\mathbb{Z}^2\mathbf{B}^+$, the dual of the element-position lattice. For an arbitrary $\boldsymbol{k}\in\mathbb{Z}^2$ then,

$$\begin{aligned} H\big((\mathbf{k}-\boldsymbol{k}\mathbf{B}^+)\mathbf{B}\big) &= H(\mathbf{kB}-\boldsymbol{k}\mathbf{B}^+\mathbf{B}) \\ &= H(\mathbf{kB}-\boldsymbol{k}) \\ &= H(\mathbf{kB}). \end{aligned}$$

The second and third equalities follow respectively from identity $\mathbf{B}^+\mathbf{B}=\mathbf{I}$ and the periodicity property of the Fourier transform in Section 4.2 above.

5.1.3 Grating-Lobe Prevention Determines Element Spacing

On the upper left and upper right respectively of Fig. 6 are an example element-position lattice and the corresponding array-periodicity lattice, respectively comprising points in the normalized position plane and in the beamspace plane, the latter a function of DOA parameter $\mathbf{k}$. A visual representation of array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ like that shown can also be taken as a sort of schematic of the ideal unsteered array factor itself. This is because unsteered array factor $H(\mathbf{kB})$ is most commonly designed to have a narrow beam at boresight or $\mathbf{k}=0$, a point of array-periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$. The periodicity of array factor $H(\mathbf{kB})$ then gives it a beam at every other point of that lattice as well, so that all lattice points schematically represent unsteered array-factor beams.

As discussed above in Section 3.3, a propagating plane wave puts $\mathbf{k}$ in the visible region of beamspace. There it is just the array-plane component of $\frac{\lambda}{c/|\mathsf{f}|}\widehat{u}$, where unit vector $\widehat{u}$ points in the DOA. When frequency $|\mathsf{f}|$ is at the top of the signal band, scale factor $\lambda|\mathsf{f}|/c$ becomes unity and makes $\mathbf{k}$ just the array-plane component of the DOA unit vector $\widehat{u}$ itself. When we draw or plot the beamspace $\mathbf{k}$ plane then, we generally overlay a visible-region globe gridded with latitude and longitude lines representing azimuth and elevation angles of this DOA unit vector. We draw it with unit radius to represent the top of the signal band and simply understand that it shrinks proportionally with frequency elsewhere. When unsteered array factor $H(\mathbf{kB})$ is translated or slid in $\mathbf{k}$ by steering vector $\mathbf{k}_\Delta$ to create steered array factor $H\big((\mathbf{k}-\mathbf{k}_\Delta)\mathbf{B}\big)$, the boresight beam moves to $\mathbf{k}_\Delta$ and we can read the corresponding azimuth and elevation angles from the visible-region globe's grid. We seldom make this translation explicit in the graphics and instead leave it to the imagination.

In the beamspace $\mathbf{k}$ plane on the upper right of Fig. 6 however, to illustrate a point, one translated version of the lattice is shown very faintly with the schematic boresight beam translated to -65° azimuth and -55° elevation. When the boresight beam moves to these coordinates, all the other beams, depicted prior to steering by lattice points, move in parallel. In the example one such steered schematic beam appears just to the upper right of the visible-region globe.

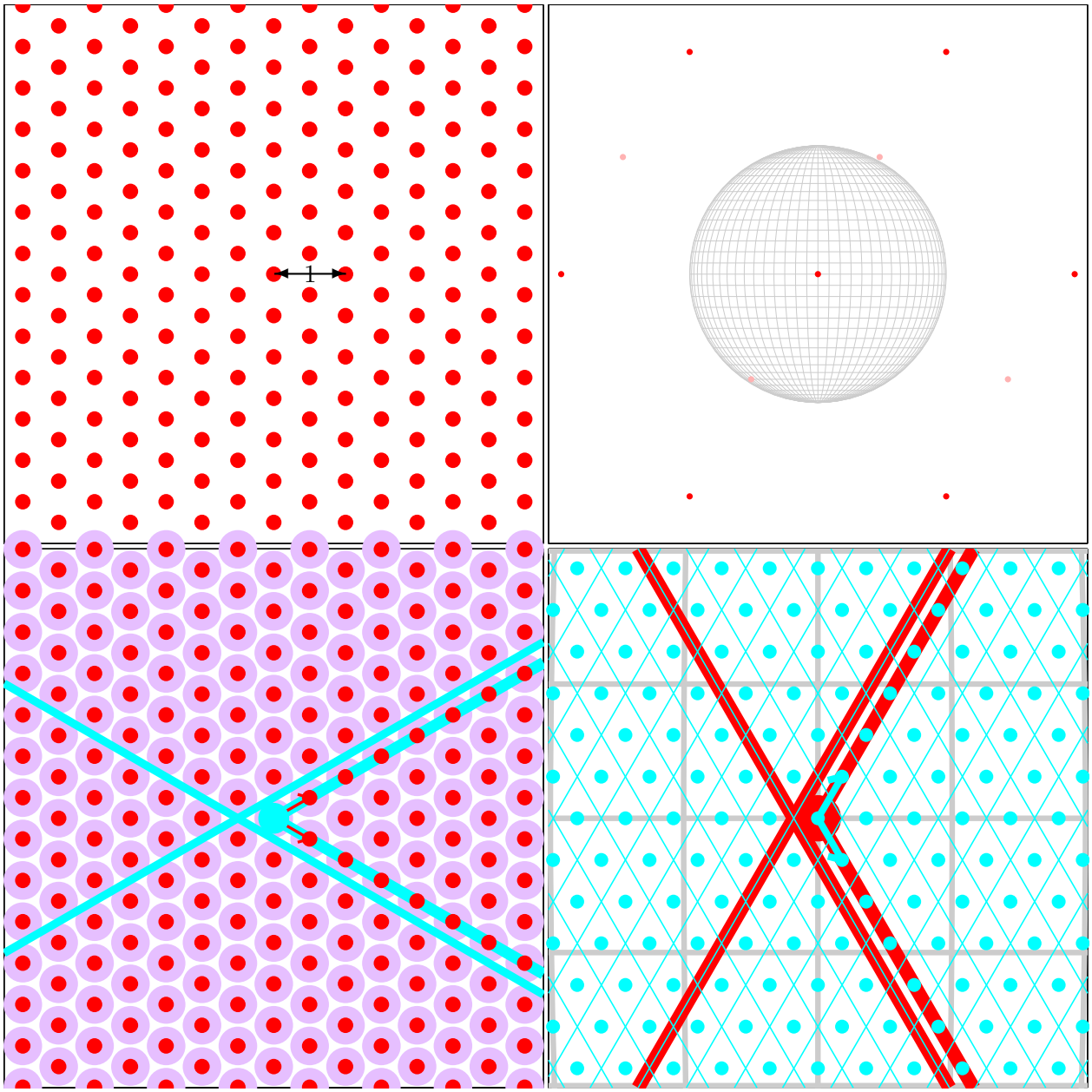


Fig. 6 — Top: element-position lattice $\mathbf{B}\mathbb{Z}^2$ (left) with classic nearest-neighbor spacing $1/\sqrt{3}$ and corresponding array-factor-periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ (right) with, to ensure no grating lobes, a nearest-neighbor spacing of 2, the latter lattice shown both untranslated and translated to move the boresight point to an example beam position at -65° azimuth and -55° elevation. Bottom: closeups of Fig. 8 (left) and Fig. 7 (right). The left plots, both of normalized position $\frac{1}{\lambda}\mathbf{x}$, are on the same scale. The right plots show the $\mathbf{k}$ plane with 5° grid lines and with the lower plot zoomed $12\times$ relative to the upper plot.

Here we see why the classic choice for nearest-neighbor spacing in array-periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ is exactly 2. In array design increasing element density $|\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{-1/2}/\lambda^2$ in the array plane increases cost, so generally that density is minimized. Equivalently, the beamspace density $|\mathbf{B}^{\mathrm{T}}\mathbf{B}|^{1/2}\lambda^2$ of array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ is maximized. That maximization must, however, never allow array-factor steering to put two beams in the visible-region globe simultaneously, for then one of those two beams would be an undesired *grating lobe*, potentially causing the array to admit contaminating signals from some undesired direction. The minimum safe spacing for these periodic beams is the globe's diameter of 2, assuming the boresight beam could be steered until its edge just touched the globe perimeter's inner edge. (Restrictions on steering in some systems permit nearest-neighbor spacings of less than 2.) This argument is for the upper end of the signal band, but the globe gets smaller for lower frequencies, so a spacing that's grating-lobe safe at the upper end is grating-lobe safe at lower frequencies as well. Given such a lower bound on nearest-neighbor distance, the beamspace density of array-periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ is maximized by making it a hexagonal lattice. This makes element-position lattice $\mathbf{B}\mathbb{Z}^2$ hexagonal as well, and the spacing of 2 in $\mathbb{Z}^2\mathbf{B}^+$ results in a spacing of $1/\sqrt{3}$ in $\mathbf{B}\mathbb{Z}^2$, giving the physical element centers the classic spacing[1] of $\lambda/\sqrt{3}$.

5.1.4 Single-Beam Implementation

To implement array-output computation (22), first substitute the right half of change of variable (24) into phase-delay definition (19) to obtain

$$(s^0 \curvearrowright \boldsymbol{f}_\Delta)_{\boldsymbol{n}} = s^0_{\boldsymbol{n}}\, \mathrm{e}^{-\mathrm{j}2\pi \mathbf{k}_\Delta \mathbf{B}\boldsymbol{n}},$$

and then use convolution definition (18) to see that

$$\big(s^0 \curvearrowright \boldsymbol{f}_\Delta \star h\big)_{\boldsymbol{n}} = \sum_{\boldsymbol{m}} \big(s^0_{\boldsymbol{m}}\, \mathrm{e}^{-\mathrm{j}2\pi \mathbf{k}_\Delta \mathbf{B}\boldsymbol{m}}\big)\, h_{\boldsymbol{n}-\boldsymbol{m}}.$$

To complete the implementation of array output (22), just sample at $\boldsymbol{n} = 0$ to obtain

$$s_0 = \sum_{\boldsymbol{m}} h_{-\boldsymbol{m}} \big(s^0_{\boldsymbol{m}}\ \mathrm{e}^{-\mathrm{j}2\pi \mathbf{k}_\Delta \mathbf{B}\boldsymbol{m}}\big) \tag{26}$$

$$= \sum_{\boldsymbol{m}} \big(h_{-\boldsymbol{m}}\ s^0_{\boldsymbol{m}}\big) \mathrm{e}^{-\mathrm{j}2\pi \mathbf{k}_\Delta \mathbf{B}\boldsymbol{m}}. \tag{27}$$

Computation order (26) is typically used in analog arrays, and either computation order, (26) or (27), is equally reasonable for digital arrays.

5.2 Steering Vectors from a Steering Lattice

The single-beam approach above can of course be used several times in parallel, with different steering vectors $\mathbf{k}_\Delta$, to obtain several beams simultaneously. When more than a very few beams are to be realized digitally, however, an FFT approach to steering is likely to require less real-time computation in the implementation. FFT steering's one limitation is that all steering vectors $\mathbf{k}_\Delta$ must be restricted to some *steering*

[1] When elements are required to be on a square lattice in spite of its nonoptimality, a spacing of 2 in the array-periodicity lattice results instead in the equally classic $\lambda/2$ physical element spacing and an element density higher than in the $\lambda/\sqrt{3}$ hexagonal-lattice case by a factor of $2/\sqrt{3}$, an element density penalty of some 15.5%.

lattice $\mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$. Because $\mathbf{R}_{\text{steer}}$ must be a 2×2 resampling matrix, certain sublattice and superlattice relationships are necessary.

$$\begin{array}{lcccl} \textit{element-position lattice} & \mathbf{B}\mathbb{Z}^2 & \overset{\text{dual}}{\longleftrightarrow} & \mathbb{Z}^2\mathbf{B}^+ & \textit{array-factor periodicity lattice} \\ & \cup & \text{density ratio } |\mathbf{R}_{\text{steer}}| & \cap & \\ \textit{dual steering lattice} & \mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2 & \longleftrightarrow & \mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+ & \textit{steering lattice} \end{array}$$

To determine the effect of this steering-lattice restriction on array-output computation (27), set $\mathbf{k}_\Delta = \boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ for some $\boldsymbol{k}\in\mathbb{Z}^2$ and, simply to put the result into the most familiar form, change $\boldsymbol{m}$ to $\boldsymbol{n}$:

$$\begin{aligned} \underset{\text{beam}\,\boldsymbol{k}}{s_0} &= \sum_{\boldsymbol{n}} \left(h_{-\boldsymbol{n}} s_{\boldsymbol{n}}^0\right) \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+\mathbf{B}\boldsymbol{n}} \\ &= \sum_{\boldsymbol{n}} \left(h_{-\boldsymbol{n}} s_{\boldsymbol{n}}^0\right) \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n}} . \end{aligned} \tag{28}$$

The second equality follows of course from identity $\mathbf{B}^+\mathbf{B} = \mathbf{I}$.

The big-picture diagram of beamspace $\mathbf{k}$ on the upper right of Fig. 6 is duplicated in Fig. 7 in a much larger size with several additions, most notably an example steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ based on $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} 64 & 0 \\ 0 & 64 \end{smallmatrix}\right]$. Each point of that steering lattice is enclosed in a tiny diamond-shaped tile for increased visibility. Of far more significance are the large tiles, which are discussed below in Section 5.4. The finer features of this diagram are more easily visible in the close up on the lower right in Fig. 6, which is at an approximately $6\times$ magnified scale on the page. The particularly simple form of resampling matrix $\mathbf{R}_{\text{steer}}$ used here, just an identity matrix times a scale factor, makes steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ just a scaled-down version of array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$. The analysis of Section 5.5 below implies that such an $\mathbf{R}_{\text{steer}}$ with the scale factor a power of two leads to the simplest possible structure for the beamsteering FFT.

5.3 The DFT Input Space

Making the steering lattice a superlattice of the array-factor periodicity lattice gives identical phases to the complex exponentials in many of the terms of multibeam array-output computation (28). Working out the details here allows us to factor out those common factors and organize beam-computation inputs into DFT- or FFT-style bins. The key is a particular decomposition of element index $\boldsymbol{n}$.

As we work this out, we can use as an example the system that we began discussing with Figs. 6 and 7. The array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ pictured in Fig. 7 and on the right in Figs. 6 is the dual of the element-position lattice $\mathbf{B}\mathbb{Z}^2$ shown in Fig. 8 and shown about $5\times$ magnified through a smaller viewing window on the lower left in Fig. 6. The 2,527 specific points of element-position lattice $\mathbf{B}\mathbb{Z}^2$ that are highlighted in Fig. 8, a few of which appear in Fig. 6 as well, are those points $\mathbf{B}\boldsymbol{n}$ on or inside a circle of radius $\sqrt{691/3} \approx 15.18$ centered on the origin. In this hypothetical design the outputs of the elements at these positions are the ones used to compute the array output: the output $s_{\boldsymbol{n}}^0$ of the element at such a position $\mathbf{B}\boldsymbol{n}$ is multiplied in multibeam computation (28) by a nonzero weight $h_{-\boldsymbol{n}}$, making these the active elements. There is nothing special about this array size or shape, but the ideas are easier to discuss in the context of a specific example.

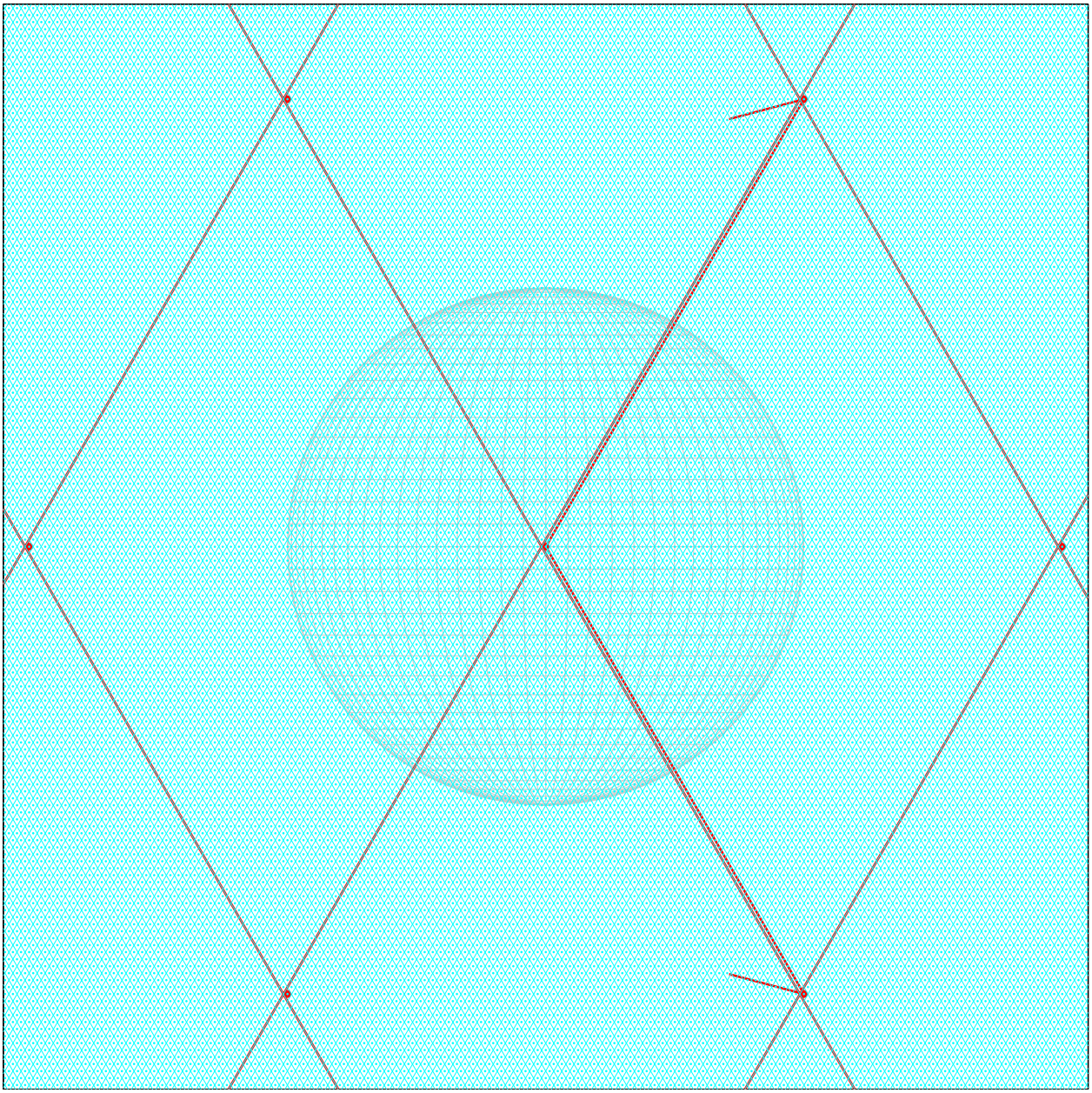


Fig. 7 — Tile boundaries partition beamspace into large DFT output tiles, identical except for translation, such that each point of steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ lies in exactly one tile and such that each tile contains exactly one point of array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$. In this example $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} 64 & 0 \\ 0 & 64 \end{smallmatrix}\right]$, so the tiles each contain 64^2 points of steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$. Here a simple choice of tile shape, based on the basis-vector rows of $\mathbf{B}^+$, lines up 64 points of the latter along each tile edge. Fine lines divide each tile into a 64×64 grid simply to make the steering lattice structure easier to see, both here and in the closeup on the lower right in Fig. 6.

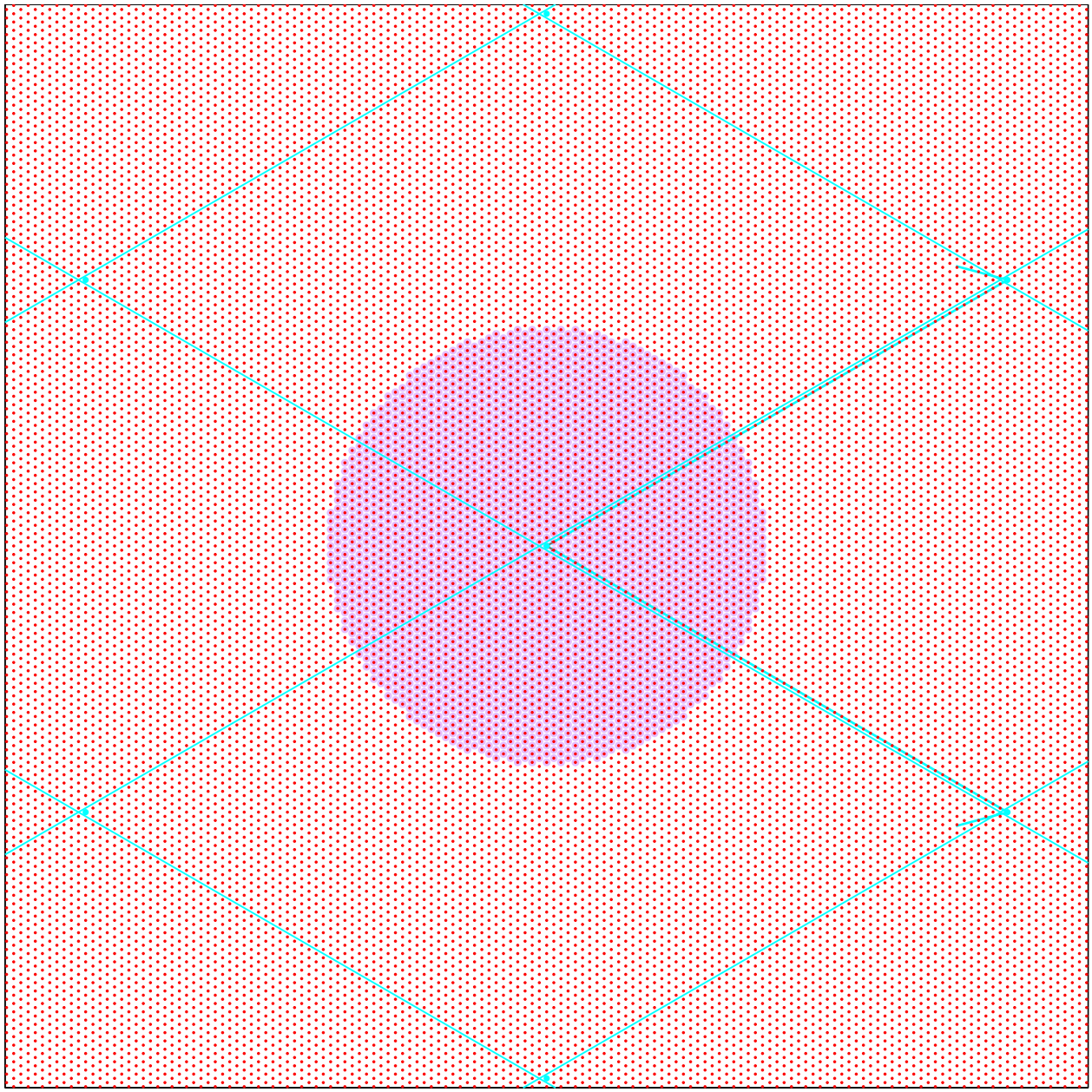


Fig. 8 — Tile boundaries partition the normalized array plane into large DFT input tiles, identical except for translation, such that each point of element-position lattice $\mathbf{B}\mathbb{Z}^2$ lies in exactly one tile and such that each tile contains exactly one point of dual steering lattice $\mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2$. Positions of actual active array elements, those that are given nonzero weights in the array-output computation, are **highlighted**. In this example $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} 64 & 0 \\ 0 & 64 \end{smallmatrix}\right]$, so the tiles must each contain 64^2 points of $\mathbf{B}\mathbb{Z}^2$. Here a simple choice of tile shape, based on the basis-vector columns of $\mathbf{B}\mathbf{R}_{\text{steer}}$, lines up 64 points of the latter along each tile edge. A closeup appears on the lower left in Fig. 6.

5.3.1 Input Tiling and Index Decomposition

We begin by tiling the element-position plane, as in the example of Figs. 8, into

> *(defining condition)*
> *identical "input tiles" such that every point of element-position lattice $\mathbf{B}\mathbb{Z}^2$ (or the whole array plane in fact) is in exactly one tile and such that each of the tiles contains exactly one point of dual steering lattice $\mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2$.*

Let us call that one point of dual steering lattice $\mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2$ the tile's *anchor*. Here "identical" means any tile can be translated—slid with no change in orientation—to create any of the others. It is convenient to use the tile anchored at the origin as a sort of prototype tile in that way, so let us call that tile the ⟨input prototile⟩. As long as the defining condition above is satisfied, it makes no actual difference what the tile shape is, but it is convenient to standardize in Sections 5.3.2 and 5.3.3 below on a simple way to use basis vectors to construct parallelogram-shaped tiles like the ones in Fig. 8. Regardless of tile shape the defining condition for tiles above results in $|\mathbf{R}_{\text{steer}}|$ element positions in each tile, simply because that is the ratio of the densities of the two lattices.

Given a satisfactory tiling, we next decompose the element index $\boldsymbol{n}$ using the tiles as a guide. The strategy comes from the twin facts that (1) every element position is in a tile and (2) every tile is just the ⟨input prototile⟩ translated so as to move its anchor, the origin, to a desired position. It follows from these facts that any point of element position lattice $\mathbf{B}\mathbb{Z}^2$ can be uniquely decomposed as a sum of an anchor position from dual steering lattice $\mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2$ and a point of element-position lattice $\mathbf{B}\mathbb{Z}^2$ from inside the ⟨input prototile⟩, an intra-tile offset. In other words, for any $\boldsymbol{n} \in \mathbb{Z}^2$ there is exactly one combination of an $\boldsymbol{m} \in \mathbb{Z}^2$ and an $\boxed{\boldsymbol{n}} \in \mathbb{Z}^2$ with $\mathbf{B}\boxed{\boldsymbol{n}} \in$ ⟨input prototile⟩ such that

$$\mathbf{B}\boldsymbol{n} = \mathbf{B}\mathbf{R}_{\text{steer}}\boldsymbol{m} + \mathbf{B}\boxed{\boldsymbol{n}}. \tag{29}$$

Multiply on the left by $\mathbf{B}^+$ and use identity $\mathbf{B}^+\mathbf{B} = \mathbf{I}$ to write equivalent decomposition

$$\boldsymbol{n} = \mathbf{R}_{\text{steer}}\boldsymbol{m} + \boxed{\boldsymbol{n}}. \tag{30}$$

Here the outer box in notation $\boxed{\boldsymbol{n}}$ signifies that its possible values are restricted to some finite set determined by a tiling.

5.3.2 The Simplest Case

The most straightforward system designs have $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$ for some integer expansion factor N_{ex}. The design of Figs. 6, 7, and 8 is such a system with $N_{\text{ex}} = 64$. In such systems steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ is just array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ shrunk by a factor of N_{ex}, and dual steering lattice $\mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2$ is just element-position lattice $\mathbf{B}\mathbb{Z}^2$ magnified by N_{ex}. Each tile contains exactly $|\mathbf{R}_{\text{steer}}| = N_{\text{ex}}^2$ element positions. Using notation

$$\boldsymbol{n} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \qquad \boldsymbol{m} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \qquad \boxed{\boldsymbol{n}} = \begin{bmatrix} \boxed{\boldsymbol{n}}_1 \\ \boxed{\boldsymbol{n}}_2 \end{bmatrix}$$

index-vector decomposition (30) becomes the pair of equations

$$\begin{aligned} n_1 &= N_{\text{ex}} m_1 + \boxed{\boldsymbol{n}}_1, \\ n_2 &= N_{\text{ex}} m_2 + \boxed{\boldsymbol{n}}_2. \end{aligned}$$

In this simple special case a simple tiling strategy that always works, the one used in Figs. 6 through 8, is to choose the ⟨input prototile⟩ to include element positions

$$\left\{ \mathbf{B}\boldsymbol{n} \text{ such that } \boldsymbol{n} = \left[\begin{smallmatrix} n_1 \\ n_2 \end{smallmatrix}\right] \text{ with } 0 \le n_1 < N_{\text{ex}} \text{ and } 0 \le n_2 < N_{\text{ex}} \right\}. \tag{31}$$

This choice results in

$$\begin{aligned} \boxed{\boldsymbol{n}}_1 &= n_1 \bmod N_{\text{ex}}, \\ \boxed{\boldsymbol{n}}_2 &= n_2 \bmod N_{\text{ex}}, \\ m_1 &= \lfloor n_1 / N_{\text{ex}} \rfloor, \\ m_2 &= \lfloor n_2 / N_{\text{ex}} \rfloor \end{aligned}$$

or

$$\begin{aligned} \boxed{\boldsymbol{n}} &= \begin{bmatrix} n_1 \bmod N_{\text{ex}} \\ n_2 \bmod N_{\text{ex}} \end{bmatrix}, \\ \boldsymbol{m} &= \begin{bmatrix} \lfloor n_1 / N_{\text{ex}} \rfloor \\ \lfloor n_2 / N_{\text{ex}} \rfloor \end{bmatrix}. \end{aligned}$$

The approach just outlined is actually the general approach discussed next applied to the special case of an $\mathbf{R}_{\text{steer}}$ of the form $\left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$.

5.3.3 Basis-Vector Tiles

Alternatively, the basis vectors of the dual steering lattice $\mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2$ as pictured in Fig. 8 can be used in a slightly more mathematical tile construction that is perfectly general, that works for any resampling matrix $\mathbf{R}_{\text{steer}}$, regardless of its form. An element position is naturally expressed as $\mathbf{p} = \mathbf{B}\boldsymbol{n}$ for some $\boldsymbol{n} \in \mathbb{Z}^2$, but we know how to express that position in any basis we like. In Fig. 8 we can see that if we express it as a linear combination of the basis vectors of the dual steering lattice and write it as $\mathbf{p} = \mathbf{B}\mathbf{R}_{\text{steer}}\boldsymbol{w}$ for some column two-vector $\boldsymbol{w} = \left[\begin{smallmatrix} w_1 \\ w_2 \end{smallmatrix}\right]$, point $\mathbf{p}$ is in the ⟨input prototile⟩ if both $0 \le w_1 < 1$ and $0 \le w_2 < 1$ hold. We can adopt this as the definition of *basis-vector tiles*, which we can make our standard scheme. In effect we've constructed our ⟨input prototile⟩ as a parallelogram with corner points given by the origin, the two basis-vector columns of $\mathbf{B}\mathbf{R}_{\text{steer}}$, and the sum of those two basis vectors. The specific choices of the four "$\le$" and "$<$" tests above define this ⟨input prototile⟩ to include two of the four parallelogram edges and one of the four parallelogram corners while excluding the others, choices that satisfy the defining condition for tiles in Section 5.3.1 above.

To find $\boldsymbol{w}$ such that $\mathbf{p} = \mathbf{B}\mathbf{R}_{\text{steer}}\boldsymbol{w}$, left multiply the latter by $\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^{+}$ and use identity $\mathbf{B}^{+}\mathbf{B} = \mathbf{I}$ to show that $\boldsymbol{w} = \mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^{+}\mathbf{p}$, then substitute the original form $\mathbf{p} = \mathbf{B}\boldsymbol{n}$ and use identity $\mathbf{B}^{+}\mathbf{B} = \mathbf{I}$ again to obtain $\boldsymbol{w} = \mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n}$. Once we adopt MATLAB's convention that the floor operation that rounds towards $-\infty$, denoted mathematically by enclosure in $\lfloor \quad \rfloor$ brackets, should apply to vectors elementwise,

conditions $0 \leq w_1 < 1$ and $0 \leq w_2 < 1$ can be jointly expressed as just $\lfloor \boldsymbol{w} \rfloor = 0$. Our $\langle$input prototile$\rangle$ therefore contains element positions $\mathbf{B}\boldsymbol{n}$ for which $\lfloor \mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} \rfloor = 0$.

This actually gives us a simple, general way to calculate $\boldsymbol{m}$ and $\boxed{\boldsymbol{n}}$, because the $\mathbf{B}\boxed{\boldsymbol{n}}$ term in lattice-point decomposition (29) is, by its definition as an intra-tile offset, itself a position in the $\langle$input prototile$\rangle$ and therefore has, using our new $\langle$input prototile$\rangle$ definition, $\lfloor \mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}} \rfloor = 0$. This makes it trivial to solve equivalent index-vector decomposition (30) for $\boldsymbol{m}$: just left multiply by $\mathbf{R}_{\text{steer}}^{-1}$ and observe that, since adding an integer can always be pulled out of a floor operation so that $\lfloor n + x \rfloor = n + \lfloor x \rfloor$,

$$\lfloor \mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} \rfloor = \lfloor \boldsymbol{m} + \mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}} \rfloor = \boldsymbol{m} + \lfloor \mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}} \rfloor = \boldsymbol{m}.$$

In principle then, obtaining index-vector decomposition (30) then amounts to just computing

$$\boldsymbol{m} = \lfloor \mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} \rfloor, \tag{32}$$

$$\boxed{\boldsymbol{n}} = \boldsymbol{n} - \mathbf{R}_{\text{steer}}\boldsymbol{m}. \tag{33}$$

5.3.4 Making Numerical Computation Robust

As written, anchor-point index computation (32) is vulnerable to numerical errors. For example, in MATLAB on the author's computer, using $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} 3 & -2 \\ 2 & 5 \end{smallmatrix}\right]$ and $\boldsymbol{n} = \left[\begin{smallmatrix} -19 \\ -19 \end{smallmatrix}\right]$ and computing $\mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n}$ naïvely using `inv(R)*n` or thoughtfully using `R\n` yields, respectively,

$$\mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} = \begin{bmatrix} -7 \\ -1 - 2^{-51} \end{bmatrix} \qquad \text{or} \qquad \mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} = \begin{bmatrix} -7 \\ -1 - 2^{-52} \end{bmatrix}$$

when actually $\mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} = \left[\begin{smallmatrix} -7 \\ -1 \end{smallmatrix}\right]$ would be correct. These numerical errors in the 51^{st} and 52^{nd} bits to the right of the binary point cause MATLAB's `floor( )` function to yield $\lfloor \mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} \rfloor = \left[\begin{smallmatrix} -7 \\ -2 \end{smallmatrix}\right]$ instead of the correct result $\lfloor \mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} \rfloor = \left[\begin{smallmatrix} -7 \\ -1 \end{smallmatrix}\right]$.

We can rework computation (32) to remove this numerical vulnerability. Multiply the argument of the floor operation by both $|\mathbf{R}_{\text{steer}}|$ and its inverse and write

$$\boldsymbol{m} = \left\lfloor \frac{1}{|\mathbf{R}_{\text{steer}}|} \left(|\mathbf{R}_{\text{steer}}| \mathbf{R}_{\text{steer}}^{-1} \right) \boldsymbol{n} \right\rfloor.$$

We can round the elements of the product in the parentheses to integers to remove the numerical noise of matrix inversion if we can argue that this product must actually be an integer matrix. For any invertible matrix $\mathbf{R}$ in fact, $|\mathbf{R}|\,\mathbf{R}^{-1} = \pm \det(\mathbf{R})\,\mathbf{R}^{-1} = \pm \operatorname{adj}(\mathbf{R})$, using in the last step that $\mathbf{R}^{-1} = \frac{1}{\det(\mathbf{R})} \operatorname{adj}(\mathbf{R})$. When $\mathbf{R}$ is an integer matrix, its adjugate matrix $\operatorname{adj}(\mathbf{R})$ is an integer matrix as well, as therefore are $\pm \operatorname{adj}(\mathbf{R})$ and $|\mathbf{R}|\,\mathbf{R}^{-1}$. Now

$$\boldsymbol{m} = \left\lfloor \frac{1}{|\mathbf{R}_{\text{steer}}|} \left(\mathsf{round}\big(|\mathbf{R}_{\text{steer}}| \mathbf{R}_{\text{steer}}^{-1} \big) \boldsymbol{n} \right) \right\rfloor. \tag{34}$$

For any integer m and positive integer N, one variety of integer division is defined by $m \,/_{\text{int}}\, N \triangleq \lfloor m/N \rfloor$ and can be computed in MATLAB (even for vector m) as `double(idivide(int64(m),N,'floor'))`. It's clumsy but absolutely safe. The quantity in large parentheses in computation (34) is an integer vector, so using such an integer division of every element of that parenthesized vector by $|\mathbf{R}_{\text{steer}}|$ allows us to write

$$\boldsymbol{m} = \Big(\mathsf{round}\big(|\mathbf{R}_{\text{steer}}|\mathbf{R}_{\text{steer}}^{-1}\big)\boldsymbol{n}\Big) \Big/_{\text{int}} |\mathbf{R}_{\text{steer}}| \tag{35}$$

and be fully confident that there are no numerical issues. Indeed, test computation yields $\lfloor \mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} \rfloor = \left[\begin{smallmatrix} -7 \\ -1 \end{smallmatrix}\right]$ for the example above.

An alternate approach, based on solving index-vector decomposition (30) for $\boldsymbol{m}$ after first computing $\boxed{\boldsymbol{n}}$ safely using MATLAB's `mod( )` function rather than integer division, is presented in Ref. 1. Integer matrix $\mathsf{round}(|\mathbf{R}_{\text{steer}}|\mathbf{R}_{\text{steer}}^{-1})$ is also at the heart of this alternate approach. Regardless of which approach is taken, this integer matrix is just $\left[\begin{smallmatrix} r_{22} & -r_{12} \\ -r_{21} & r_{11} \end{smallmatrix}\right]$ if $\det(\mathbf{R}_{\text{steer}}) > 0$ or $\left[\begin{smallmatrix} -r_{22} & r_{12} \\ r_{21} & -r_{11} \end{smallmatrix}\right]$ if $\det(\mathbf{R}_{\text{steer}}) < 0$, where $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{smallmatrix}\right]$.

5.3.5 Folding Weighted Inputs Creates Input Bins

Since decomposition (30) of $\boldsymbol{n}$ is always available, in multibeam computation (28) we can substitute for $\boldsymbol{n}$ and sum over $\boldsymbol{m}$ and $\boxed{\boldsymbol{n}}$ instead:

$$\underset{\text{beam}\,\boldsymbol{k}}{s_0} = \sum_{\substack{\boxed{\boldsymbol{n}} \in \mathbb{Z}^2 \text{ such that} \\ \mathbf{B}\boxed{\boldsymbol{n}} \in \langle\text{input prototile}\rangle}} x_{\boxed{\boldsymbol{n}}}\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}}}, \tag{36}$$

$$x_{\boxed{\boldsymbol{n}}} \triangleq \sum_{\boldsymbol{m}} \Big[h_{-\boldsymbol{n}} s_{\boldsymbol{n}}^{0}\Big]_{\boldsymbol{n}=\mathbf{R}_{\text{steer}}\boldsymbol{m}+\boxed{\boldsymbol{n}}}. \tag{37}$$

A complex-exponential factor $\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\mathbf{R}_{\text{steer}}\boldsymbol{m}} = \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{k}\boldsymbol{m}} = 1$ has disappeared. Engineering this disappearance is in fact the whole point of decomposition using input tiles, because that decomposition is what makes the only remaining complex exponential independent of $\boldsymbol{m}$ so it can be factored out of the sum in (37) to leave the preliminary folding (sometimes termed *data turning*) of weighted element outputs there independent of beam index $\boldsymbol{k}$. No matter how few or how many beams we compute, we compute (37) exactly $|\mathbf{R}_{\text{steer}}|$ times, once for each possible value of $\boxed{\boldsymbol{n}}$, in effect obtaining what in the next section are the $|\mathbf{R}_{\text{steer}}|$ distinct input bins of a generalized DFT.

What's less obvious but no less important is that folding sum (37), a sum over all tiles that is calculated for each position within a tile, in practical designs has very few nonzero terms, often just one or even none. This is the case for our example design, in fact, as can be seen in Fig. 8. In that example, for each intra-tile offset $\mathbf{B}\boxed{\boldsymbol{n}}$ there is at most one $\boldsymbol{m}$ for which the corresponding **point** in Fig. 8 has a nonzero weight $h_{-(\mathbf{R}_{\text{steer}}\boldsymbol{m}+\boxed{\boldsymbol{n}})}$ in folding sum (37).

5.4 The DFT Output Space and the Beamsteering DFT

Beamspace, as pictured in the example of Fig. 7 and its closeup in the lower right of Fig. 6, can be tiled with identical output tiles (which need not resemble the input tiles) such that every tile contains exactly one

point of array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ and every point of steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ is in exactly one tile.

Suppose an allowable output tiling is in place. The tile containing an arbitrary beam position $\boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ also contains exactly one point $\boldsymbol{j}\mathbf{B}^+$ of the array-factor periodicity lattice (with $\boldsymbol{j}$ not to be confused with $\mathrm{j} = \sqrt{-1}$), from which integer row two-vector $\boxed{\boldsymbol{k}}$ can in principle be solved for (practical approaches to follow) to obtain unique decomposition

$$\boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+ = \boldsymbol{j}\mathbf{B}^+ + \boxed{\boldsymbol{k}}\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+,$$

expressing the beam position as a point $\boldsymbol{j}\mathbf{B}^+$ in the array-factor periodicity lattice plus an intra-tile offset $\boxed{\boldsymbol{k}}\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ from the steering lattice. Multiply through on the right by $\mathbf{B}\mathbf{R}_{\text{steer}}$ and use $\mathbf{B}^+\mathbf{B} = \mathbf{I}$ to obtain the equivalent

$$\boldsymbol{k} = \boldsymbol{j}\mathbf{R}_{\text{steer}} + \boxed{\boldsymbol{k}}. \tag{38}$$

In the common $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$ special case—it applies with integer expansion factor $N_{\text{ex}} = 64$ in the example of Figs. 6 through 8—the ⟨output prototile⟩ anchored at the origin includes beam positions

$$\left\{\boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+ \text{ such that } \boldsymbol{k} = \begin{bmatrix} k_1 & k_2 \end{bmatrix} \text{ with } 0 \le k_1 < N_{\text{ex}} \text{ and } 0 \le k_2 < N_{\text{ex}}\right\}. \tag{39}$$

More generally we can always construct basis-vector tiles by assigning to the ⟨output prototile⟩ the beam positions $\boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ for which $\lfloor \boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1} \rfloor = 0$. Then index decomposition (38) amounts to computing

$$\begin{aligned} \boldsymbol{j} &= \lfloor \boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1} \rfloor, \\ \boxed{\boldsymbol{k}} &= \boldsymbol{k} - \boldsymbol{j}\mathbf{R}_{\text{steer}}. \end{aligned} \tag{40}$$

To realize anchor-point index computation (40) robustly, use

$$\boldsymbol{j} = \left(\boldsymbol{k}\ \mathsf{round}\big(|\mathbf{R}_{\text{steer}}|\mathbf{R}_{\text{steer}}^{-1}\big)\right) \Big/_{\text{int}} |\mathbf{R}_{\text{steer}}|. \tag{41}$$

An alternate approach [1] solves index-vector decomposition (38) for $\boldsymbol{j}$ after first computing $\boxed{\boldsymbol{k}}$ safely using MATLAB's `mod( )` function rather than integer division. Integer matrix $\mathsf{round}(|\mathbf{R}_{\text{steer}}|\mathbf{R}_{\text{steer}}^{-1})$ can be directly constructed as outlined above, at the end of Section 5.3.4, if desired.

Apply the output tiling to beam computation by substituting unique decomposition (38) into multibeam computation (36):

$$\overset{s_0}{}_{\text{beam}\boldsymbol{j}\mathbf{R}_{\text{steer}}+\boxed{\boldsymbol{k}}} = X_{\boxed{\boldsymbol{k}}}, \tag{42}$$

$$X_{\boxed{\boldsymbol{k}}} \triangleq \sum_{\boxed{\boldsymbol{n}} \in \langle \text{input prototile} \rangle} x_{\boxed{\boldsymbol{n}}}\, \mathrm{e}^{-\mathrm{j}2\pi \boxed{\boldsymbol{k}}\mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}}}. \tag{43}$$

Recognizing $\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{j}\mathbf{R}_{\text{steer}}\mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}}} = \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{j}\boxed{\boldsymbol{n}}} = 1$ has simplified the right side of (43), making it in fact independent of $\boldsymbol{j}$. That independence reflects that the phase shifts required to steer unsteered boresight beam to a given position $(\boldsymbol{j}\mathbf{R}_{\text{steer}} + \boxed{\boldsymbol{k}})\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^{+}$ don't depend on $\boldsymbol{j}$, because the array factor's periodicity makes any steering increment of $\boldsymbol{j}\mathbf{B}^{+}$ irrelevant.

The sum on the right in (43) is a generalized 2D discrete Fourier transform (DFT) [2], in recognition of which the traditional DFT output name $X_{\boxed{\boldsymbol{k}}}$ is used alongside a functional designation as such-and-such beam of s_0.

5.5 FFT Realization of the DFT in the Simple Case

When $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$ for some integer expansion factor N_{ex}, the beamsteering DFT can be computed using an ordinary 2D FFT. This is developed here, but this simple special case must not be confused with the general result. More generally a factorable $\mathbf{R}_{\text{steer}}$ leads to the generalized FFT structure developed later in Section 5.6.

In the simplest case, $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$ for some integer expansion factor N_{ex}. If the input and output prototiles are basis-vector tiles as per special-case relationships (31) and (39), we can use expansions

$$\boxed{\boldsymbol{n}} = \begin{bmatrix} \boxed{\boldsymbol{n}}_1 \\ \boxed{\boldsymbol{n}}_2 \end{bmatrix}, \qquad \boxed{\boldsymbol{k}} = \begin{bmatrix} \boxed{\boldsymbol{k}}_1 & \boxed{\boldsymbol{k}}_2 \end{bmatrix}, \qquad \boxed{\boldsymbol{k}}\mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}} = (\boxed{\boldsymbol{k}}_1\boxed{\boldsymbol{n}}_1 + \boxed{\boldsymbol{k}}_2\boxed{\boldsymbol{n}}_2)/N_{\text{ex}} \tag{44}$$

to rewrite DFT (43) as

$$X_{\left[\boxed{\boldsymbol{k}}_1 \;\; \boxed{\boldsymbol{k}}_2\right]} = \sum_{\boxed{\boldsymbol{n}}_1,\, \boxed{\boldsymbol{n}}_2 = 0}^{N_{\text{ex}}-1} x_{\left[\begin{smallmatrix} \boxed{\boldsymbol{n}}_1 \\ \boxed{\boldsymbol{n}}_2 \end{smallmatrix}\right]} \, \mathrm{e}^{-\mathrm{j}2\pi(\boxed{\boldsymbol{k}}_1\boxed{\boldsymbol{n}}_1 + \boxed{\boldsymbol{k}}_2\boxed{\boldsymbol{n}}_2)/N_{\text{ex}}} \tag{45}$$

$$= \sum_{\boxed{\boldsymbol{n}}_2=0}^{N_{\text{ex}}-1} \left(\sum_{\boxed{\boldsymbol{n}}_1=0}^{N_{\text{ex}}-1} x_{\left[\begin{smallmatrix} \boxed{\boldsymbol{n}}_1 \\ \boxed{\boldsymbol{n}}_2 \end{smallmatrix}\right]} \, \mathrm{e}^{-\mathrm{j}2\pi \boxed{\boldsymbol{k}}_1\boxed{\boldsymbol{n}}_1/N_{\text{ex}}} \right) \mathrm{e}^{-\mathrm{j}2\pi \boxed{\boldsymbol{k}}_2\boxed{\boldsymbol{n}}_2/N_{\text{ex}}} \tag{46}$$

$$= \sum_{\boxed{\boldsymbol{n}}_1=0}^{N_{\text{ex}}-1} \left(\sum_{\boxed{\boldsymbol{n}}_2=0}^{N_{\text{ex}}-1} x_{\left[\begin{smallmatrix} \boxed{\boldsymbol{n}}_1 \\ \boxed{\boldsymbol{n}}_2 \end{smallmatrix}\right]} \, \mathrm{e}^{-\mathrm{j}2\pi \boxed{\boldsymbol{k}}_2\boxed{\boldsymbol{n}}_2/N_{\text{ex}}} \right) \mathrm{e}^{-\mathrm{j}2\pi \boxed{\boldsymbol{k}}_1\boxed{\boldsymbol{n}}_1/N_{\text{ex}}} . \tag{47}$$

The form of (45) is that of an ordinary $N_{\text{ex}} \times N_{\text{ex}}$ point DFT in 2D, which is conventionally computed as per either of equivalent forms (46) and (47). Figure 9 illustrates special-case DFT (46) in block-diagram form. Each vertical block on the left realizes the inner sum, a 1D DFT, for a particular value of $\boxed{\boldsymbol{n}}_2$ to transform the dependency on position index $\boxed{\boldsymbol{n}}_1$ into dependency on beamspace index $\boxed{\boldsymbol{k}}_1$. Each horizontal block in Fig. 9 then realizes the outer sum for one value of $\boxed{\boldsymbol{k}}_1$, transforming dependency on position index $\boxed{\boldsymbol{n}}_2$ into a dependency on beamspace index $\boxed{\boldsymbol{k}}_2$. Realizing equivalent special-case DFT (47) instead would transform first on $\boxed{\boldsymbol{n}}_2$ and then on $\boxed{\boldsymbol{n}}_1$. It makes no difference which set of transforms comes first. If N_{ex} is chosen to be highly composite in the usual ways, each 1D DFT block in this structure can be efficiently realized with an FFT algorithm, and it's both appropriate and completely conventional to then consider the overall 2D realization a 2D FFT.

In many radar applications only a cluster of beams is required, and many of the beam outputs $X_{\left[\boxed{\boldsymbol{k}}_1 \;\; \boxed{\boldsymbol{k}}_2\right]}$ are unneeded. This typically allows some **output FFTs** to simply be removed, and of course the **input FFTs**

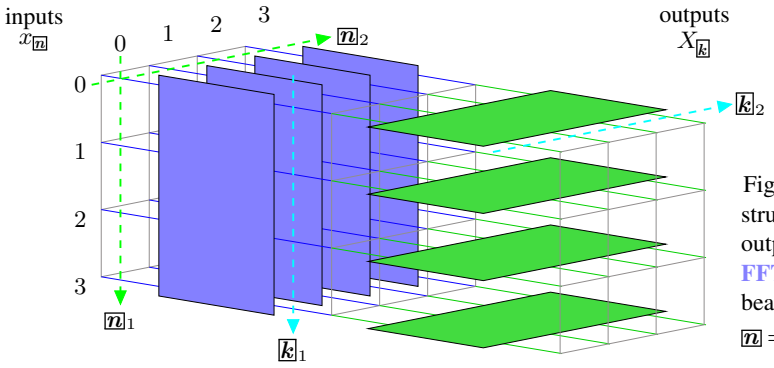


Fig. 9 — Example simplest-case FFT digital steering structure for $N_{\text{ex}} = 4$. Weighted and folded element outputs $x_{\boxed{n}}$ are here processed in turn by a bank of 1D FFTs on $\boxed{n}_1$ and a bank of 1D FFTs on $\boxed{n}_2$ to create beam outputs $X_{\boxed{k}}$. The components of index vectors $\boxed{n} = \left[\begin{smallmatrix}\boxed{n}_1 \\ \boxed{n}_2\end{smallmatrix}\right]$ and $\boxed{k} = \left[\boxed{k}_1 \;\; \boxed{k}_2\right]$ range over $0, \ldots, N_{\text{ex}}{-}1$.

then need not provide those outputs that would have driven the removed FFTs. Ignoring these potential simplifications in favor of the simplicity of using FPGA or VLSI "cores" as-is is apt to be expensive both in chip area and power consumption and may in borderline cases even negate the advantage of the FFT approach over independent brute-force computation of every beam of the cluster.

5.6 FFT Realization of the DFT in the General Case

Typical reasons to consider going beyond the 2D FFT just discussed are outlined next. A suitable general FFT factoring is then developed, beginning with appropriate tiling-based element- and beam-position addressing schemes.

5.6.1 An Extra Small-Determinant Factor in $\mathbf{R}_{\text{steer}}$ Can be Useful

As per the discussion of sublattice density at the end of Section 2.3.1 above, there are $|\mathbf{R}_{\text{steer}}|$ beam positions in steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+$ per period of array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$. The area of that period is generally fixed by the grating-lobe considerations discussed above in Section 5.1.3, so typically beam density is actually set by $|\mathbf{R}_{\text{steer}}|$ alone. The resampling matrix $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$ of the last section has $|\mathbf{R}_{\text{steer}}| = N_{\text{ex}}^2$, so when N_{ex} is limited to powers of two changing N_{ex} cannot effect changes in beam density by less than a factor of four. Design options for $|\mathbf{R}_{\text{steer}}|$ then start at some initial power of four, the lowest to be considered, and go up by factors of four.

$$4^{\text{initial}} \times \text{ one of } \left\{ \begin{array}{c|cccc} & 4^0 & 4^1 & 4^4 & 4^3 \ldots \\ \hline = & 1 & 4 & 16 & 32 \ldots \end{array} \right.$$

Many more possibilities result from allowing a little more flexibility and permitting N_{ex} to have one factor of three as well as arbitrarily many factors of two, as one factor of 9 is now allowed in $|\mathbf{R}_{\text{steer}}|$.

$$4^{\text{initial}} \times \text{ one of } \left\{ \begin{array}{c|ccccc} & 4^0 & 4^1 & 4^2 & 4^3 & 4^4 \ldots \\ \hline \times 1 = & 1 & 4 & 16 & 64 & 256 \ldots \\ \times 9 = & 9 & 36 & 144 & 576 \ldots & \end{array} \right.$$

In effect this is a factor of 9 or, say, 16 times an arbitrary power of four, where the factor of 9 is the new possibility introduced here. A density change of $16/9$ or $9/16$, relative to densities available without the factor of 9, becomes an option.

One attractive feature of the $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$ form is that the steering lattice has the same geometry as the array-factor periodicity lattice. Only the scale is changed. A hexagonal array-factor periodicity lattice yields a hexagonal steering lattice, and a square array-factor periodicity lattice yields a square steering lattice. If we allow a rotation as well as a scaling in obtaining the steering lattice from the array-factor periodicity lattice, we can obtain even more possible $|\mathbf{R}_{\text{steer}}|$ values while otherwise preserving this same-geometry feature. If the array-factor periodicity lattice is hexagonal, for example, the matrix product

$$\mathbf{R}_{\text{steer}} = \begin{bmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{bmatrix} \quad \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} \tag{48}$$

makes the steering lattice hexagonal as well, but scaled and rotated $30°$. Scaling is by

$$|\mathbf{R}_{\text{steer}}| = \left|\begin{bmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{bmatrix}\right| \left|\begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}\right| = 3N_{\text{ex}}^2.$$

This optional factor of three increases the spectrum of $|\mathbf{R}_{\text{steer}}|$ possibilities.

$$4^{\text{initial}} \times \text{ one of } \left\{ \begin{array}{c|ccccc} & 4^0 & 4^1 & 4^2 & 4^3 & 4^4 \ldots \\ \hline \times 1 = & 1 & 4 & 16 & 64 & 256 \ldots \\ \times 3 = & 3 & 12 & 48 & 192 \ldots & \\ \times 9 = & 9 & 36 & 144 & 576 \ldots & \end{array} \right.$$

In effect this is a factor of 9, 12, or 16 times an arbitrary power of four. These are fairly fine steps.

Instead of obtaining a factor of three in $|\mathbf{R}_{\text{steer}}|$ by including $\left[\begin{smallmatrix} 1 & -1 \\ 1 & 2 \end{smallmatrix}\right]$, we could obtain a factor of 7 or 13 or 19 by including $\left[\begin{smallmatrix} 2 & -1 \\ 1 & 3 \end{smallmatrix}\right]$ or $\left[\begin{smallmatrix} 3 & -1 \\ 1 & 4 \end{smallmatrix}\right]$ or $\left[\begin{smallmatrix} 3 & -2 \\ 2 & 5 \end{smallmatrix}\right]$ respectively, with different resulting steering-lattice rotations in each case, though computational efficiency suffers some with larger determinant factors, even with the relatively efficient approach of the next section.

If the array-factor periodicity lattice were square, a factor of 2, 5, 13, or 17 could be included in $|\mathbf{R}_{\text{steer}}|$ by respectively including in $\mathbf{R}_{\text{steer}} = \left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$ an extra factor of $\left[\begin{smallmatrix} 1 & -1 \\ 1 & 1 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 2 & -1 \\ 1 & 2 \end{smallmatrix}\right]$, $\left[\begin{smallmatrix} 3 & -2 \\ 2 & 3 \end{smallmatrix}\right]$, or $\left[\begin{smallmatrix} 4 & -1 \\ 1 & 4 \end{smallmatrix}\right]$. In each case the steering lattice would remain square but would be rotated.

A systematic approach to constructing these geometry-preserving resampling-matrix factors is described in the background material on sublattices in Ref. 6. Other possible reasons to include additional matrix factors in $\mathbf{R}_{\text{steer}}$ are discussed in Ref. 3, most importantly obtaining a nearly square steering lattice from a hexagonal array-factor periodicity lattice and vice versa. In general incorporating a new small-determinant $\mathbf{R}_{\text{steer}}$ factor should be viewed as an available tool for more flexibly customizing the steering lattice.

5.6.2 An Intermediate Lattice

Suppose $\mathbf{R}_{\text{steer}} = \mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}$, for example with one of $\mathbf{R}_{\text{out}}$ and $\mathbf{R}_{\text{in}}$ of the form $\left[\begin{smallmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{smallmatrix}\right]$ and the other a resampling matrix with a small determinant as discussed in the last section. This naturally introduces an *intermediate lattice* $\mathbb{Z}^2\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$ between the array-factor periodicity and steering lattices, so that the three

lattices form a sublattice chain, as do their duals.

$$\begin{array}{rcccl}
\textit{element-position lattice} & \mathbf{B}\mathbb{Z}^2 & \overset{\text{dual}}{\longleftrightarrow} & \mathbb{Z}^2\mathbf{B}^+ & \textit{array-factor periodicity lattice} \\
& \cup & \text{density ratio } |\mathbf{R}_{\text{out}}| & \cap & \\
\textit{dual intermediate lattice} & \mathbf{B}\mathbf{R}_{\text{out}}\mathbb{Z}^2 & \longleftrightarrow & \mathbb{Z}^2\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+ & \textit{intermediate lattice} \\
& \cup & \text{density ratio } |\mathbf{R}_{\text{in}}| & \cap & \\
& \mathbf{B}\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}\mathbb{Z}^2 & & \mathbb{Z}^2\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+ & \\
& || & & || & \\
\textit{dual steering lattice} & \mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2 & \longleftrightarrow & \mathbb{Z}^2\mathbf{R}_{\text{steer}}^{-1}\mathbf{B}^+ & \textit{steering lattice}
\end{array} \tag{49}$$

5.6.3 Element Addressing Using Two Position-Space Tilings

We can use the lattices on the left in relationships (49) to develop an alternate addressing scheme for element positions that extends the position-space tiling ideas developed above, in Section 5.3.

First, the basis vectors of dual intermediate lattice $\mathbf{B}\mathbf{R}_{\text{out}}\mathbb{Z}^2$ can be used to construct small position-space tiles, as pictured in the example of Fig. 10, to tile the plane such that each tile contains one point, its anchor point, from that dual intermediate lattice and such that each point of element-position lattice $\mathbf{B}\mathbb{Z}^2$ is in exactly one tile. This leads to a unique decomposition, in either of the two equivalent forms, of an arbitrary point from that element-position lattice as the position of an anchor point plus an intra-tile offset,

$$\mathbf{B}\boldsymbol{n} = \mathbf{B}\mathbf{R}_{\text{out}}\boldsymbol{n}' + \mathbf{B}\boxed{\boldsymbol{n}}, \tag{50}$$

$$\boldsymbol{n} = \mathbf{R}_{\text{out}}\boldsymbol{n}' + \boxed{\boldsymbol{n}} \tag{51}$$

where, if we use the optional but convenient basis-vector-tile approach developed above in Section 5.3.3,

$$\begin{aligned} \boldsymbol{n}' &= \lfloor \mathbf{R}_{\text{out}}^{-1}\boldsymbol{n} \rfloor, \\ \boxed{\boldsymbol{n}} &= \boldsymbol{n} - \mathbf{R}_{\text{out}}\boldsymbol{n}'. \end{aligned} \tag{52}$$

The decomposition can be determined graphically, using a diagram like that of Fig. 10, or the approach developed above in Section 5.3.4 can be used to compute small-tile anchor-point index (52) robustly. Either way, the values that can be taken by $\boxed{\boldsymbol{n}}$ are the $|\mathbf{R}_{\text{out}}|$ specific integer column two-vectors for which $\lfloor \mathbf{R}_{\text{out}}^{-1}\boxed{\boldsymbol{n}} \rfloor = 0$.

Second, the basis vectors of dual steering lattice $\mathbf{B}\mathbf{R}_{\text{steer}}\mathbb{Z}^2$ can be used in a similar way to construct large position-space tiles, as pictured in the example of Fig. 10, to tile the plane such that each tile contains one point, its anchor point, from that dual steering lattice and such that each point of dual intermediate lattice $\mathbf{B}\mathbf{R}_{\text{out}}\mathbb{Z}^2$ is in exactly one tile. This leads to a unique decomposition of an arbitrary point—point $\mathbf{B}\mathbf{R}_{\text{out}}\boldsymbol{n}'$ from decomposition (50) above is especially interesting—from that dual intermediate lattice as the position of an anchor point plus an intra-tile offset in either of the two equivalent forms

$$\mathbf{B}\mathbf{R}_{\text{out}}\boldsymbol{n}' = \mathbf{B}\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}\boldsymbol{\ell} + \mathbf{B}\mathbf{R}_{\text{out}}\boxed{\boldsymbol{m}}, \tag{53}$$

$$\boldsymbol{n}' = \mathbf{R}_{\text{in}}\boldsymbol{\ell} + \boxed{\boldsymbol{m}} \tag{54}$$

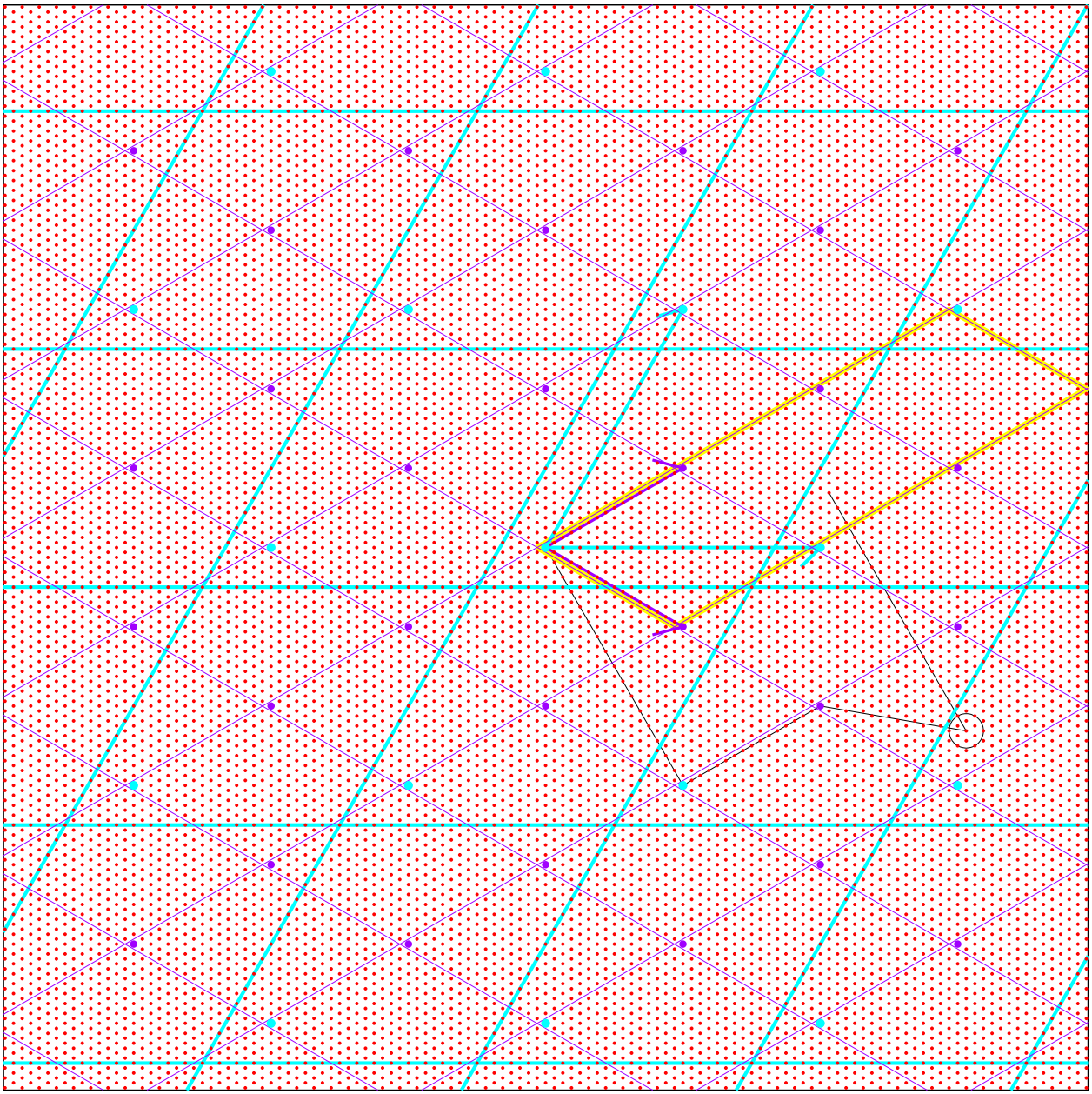


Fig. 10 — To address element positions with three digit vectors, use the Section 5.3.3 basis-vector tiling twice to create, in the example here with $\mathbf{R}_{\text{in}} = \left[\begin{smallmatrix} 1 & -1 \\ 1 & 2 \end{smallmatrix}\right]$ and $\mathbf{R}_{\text{out}} = \left[\begin{smallmatrix} 16 & 0 \\ 0 & 16 \end{smallmatrix}\right]$,

containing an anchor point from small tiles ↗↘ dual intermediate lattice $\mathbf{B}\mathbf{R}_{\text{out}}\mathbb{Z}^2$ and exactly $|\mathbf{R}_{\text{out}}|$ ↗↘ points from element-position lattice $\mathbf{B}\mathbb{Z}^2$ and

large tiles ↗↘ dual steering lattice $\mathbf{B}\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}\mathbb{Z}^2$ ↗↘ $|\mathbf{R}_{\text{in}}|$ ↗↘ dual intermediate lattice $\mathbf{B}\mathbf{R}_{\text{out}}\mathbb{Z}^2$.

Each element position can then be uniquely expressed as

a large-tile anchor point $\mathbf{B}\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}\boldsymbol{\ell}$

$+$ an offset $\mathbf{B}\mathbf{R}_{\text{out}}\boxed{\boldsymbol{m}}$ to one of that large tile's small-tile anchor points

$+$ an offset $\mathbf{B}\boxed{\boldsymbol{n}}$ to one of that small tile's element positions.

The (example) has $\boldsymbol{\ell} = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right]$, $\boxed{\boldsymbol{m}} = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right]$, and $\boxed{\boldsymbol{n}} = \left[\begin{smallmatrix} 11 \\ 6 \end{smallmatrix}\right]$. Element outputs are summed into the FFT input bins in the highlighted region located by setting $\boldsymbol{\ell} = 0$ to address bins with $\boxed{\boldsymbol{m}}$ and $\boxed{\boldsymbol{n}}$ only.

where, if we again choose to use the basis-vector-tile approach of Section 5.3.3 above,

$$\begin{aligned} \boldsymbol{\ell} &= \lfloor \mathbf{R}_{\text{in}}^{-1} \boldsymbol{n}' \rfloor, \\ \boxed{\boldsymbol{m}} &= \boldsymbol{n}' - \mathbf{R}_{\text{in}} \boldsymbol{\ell}. \end{aligned} \tag{55}$$

The decomposition can be determined graphically, using a diagram like that of Fig. 10, or the approach of Section 5.3.4 can be used to compute large-tile anchor-point index (55) robustly. Either way, the values that can be taken by $\boxed{\boldsymbol{m}}$ are the $|\mathbf{R}_{\text{in}}|$ specific integer column two-vectors for which $\lfloor \mathbf{R}_{\text{in}}^{-1} \boxed{\boldsymbol{m}} \rfloor = 0$.

Decompositions (50) and (51) can be combined with decompositions (53) and (54) to yield the decompositions of the element position and the element-position index vector given by

$$\mathbf{B}\boldsymbol{n} = \mathbf{B}\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}\boldsymbol{\ell} + \mathbf{B}\mathbf{R}_{\text{out}}\boxed{\boldsymbol{m}} + \mathbf{B}\boxed{\boldsymbol{n}}, \tag{56}$$

$$\boldsymbol{n} = \mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}\boldsymbol{\ell} + \mathbf{R}_{\text{out}}\boxed{\boldsymbol{m}} + \boxed{\boldsymbol{n}}. \tag{57}$$

In Fig. 10 the three thin line segments leading from the origin to the one (example) element position correspond to the three terms on the right in element-position decomposition (56). (The fourth segment is discussed below.)

An element position is indexed or addressed by column vector $\boldsymbol{n}$ or, using address decomposition (57), the three column vectors $\boldsymbol{\ell}$, $\boxed{\boldsymbol{m}}$, and $\boxed{\boldsymbol{n}}$, so in effect (57) relates two element-position addressing schemes. It also generalizes on mixed-radix representation of an integer. Here $\boldsymbol{\ell}$, $\boxed{\boldsymbol{m}}$, and $\boxed{\boldsymbol{n}}$ are digit vectors from $\mathbb{Z}^2$ and can respectively take an infinite number of values, exactly $|\mathbf{R}_{\text{in}}|$ values, and exactly $|\mathbf{R}_{\text{out}}|$ values. Resampling matrices $\mathbf{R}_{\text{out}}$ and $\mathbf{R}_{\text{in}}$ are radix matrices. The most significant vector (MSV) is $\boldsymbol{\ell}$, and the least significant vector (LSV) is $\boxed{\boldsymbol{n}}$. When the vectors and matrices involved are replaced with integers, the procedure here for obtaining the digit vectors in fact degenerates to the usual procedure for obtaining a mixed-radix representation of an integer.

The key ideas of this position-space tiling are summarized for review convenience in the Fig. 10 caption.

5.6.4 Beam Addressing Using Two Beamspace Tilings

In a similar way, we can use the lattices on the right in the relationships of (49) to develop an alternate addressing scheme for beam positions that extends the beamspace tiling ideas from Section 5.4 above.

First, the basis vectors of intermediate lattice $\mathbb{Z}^2\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$ can be used to construct small beamspace tiles, as pictured in the example of Fig. 11, to tile the plane such that each tile contains one point, its anchor point, from that intermediate lattice and such that each point of steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$ is in exactly one tile. This leads to a unique decomposition of an arbitrary point from that steering lattice as the position of an anchor point plus an intra-tile offset in either of the two equivalent forms

$$\boldsymbol{k}\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+ = \boldsymbol{k}'\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+ + \boxed{\boldsymbol{k}}\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+, \tag{58}$$

$$\boldsymbol{k} = \boldsymbol{k}'\mathbf{R}_{\text{in}} + \boxed{\boldsymbol{k}} \tag{59}$$

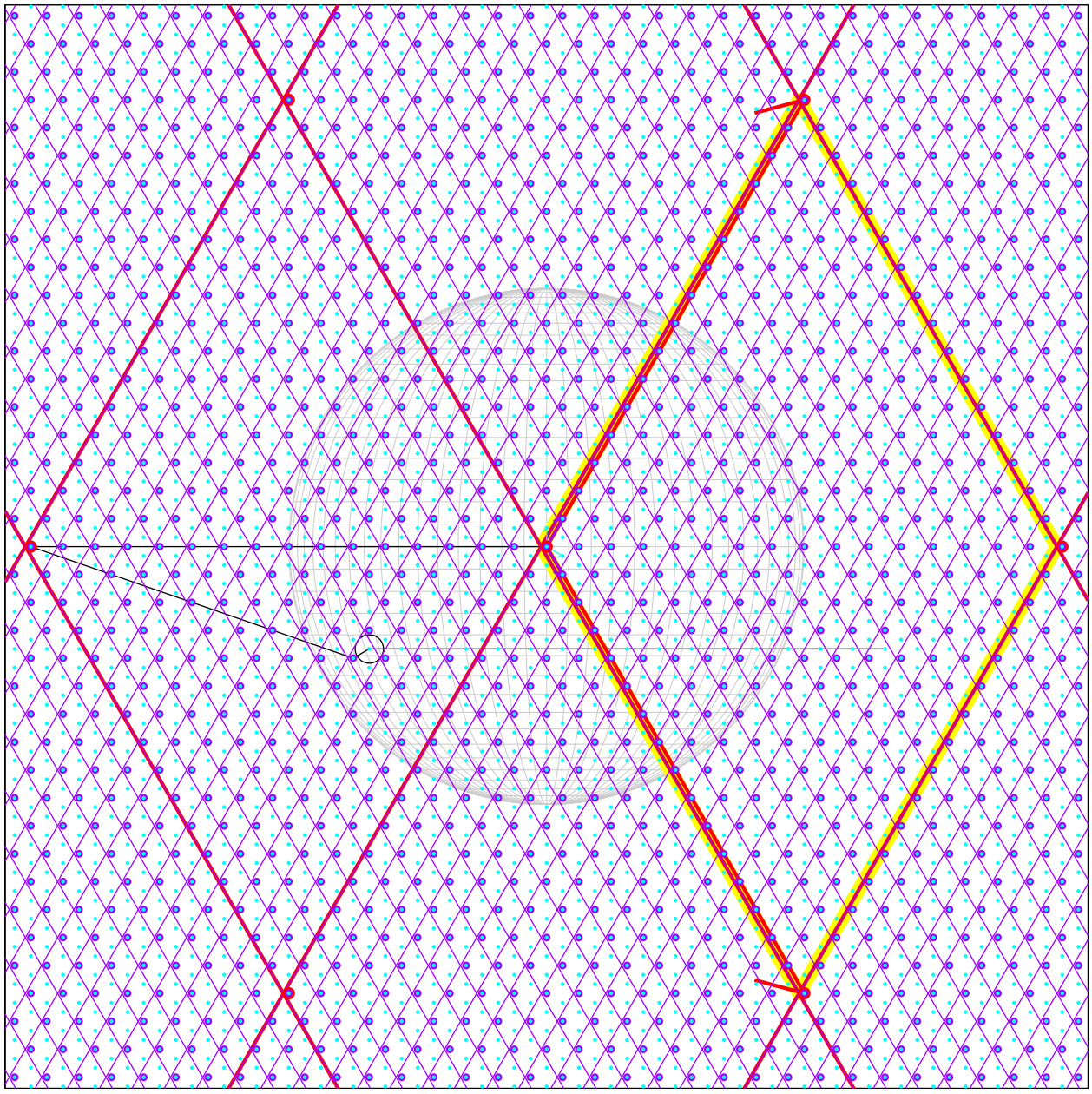


Fig. 11 — To address beam positions with three digit vectors, use the Section 5.4 basis-vector tiling twice to create, in the example here with $\mathbf{R}_{\text{in}} = \left[\begin{smallmatrix} 1 & -1 \\ 1 & 2 \end{smallmatrix}\right]$ and $\mathbf{R}_{\text{out}} = \left[\begin{smallmatrix} 16 & 0 \\ 0 & 16 \end{smallmatrix}\right]$,

containing an anchor point from / and exactly / points from

small tiles ↗↘ intermediate lattice $\mathbb{Z}^2\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$ ↗↘ $|\mathbf{R}_{\text{in}}|$ ↗↘ steering lattice $\mathbb{Z}^2\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$ and

large tiles ↗↘ array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ ↗↘ $|\mathbf{R}_{\text{out}}|$ ↗↘ intermediate lattice $\mathbb{Z}^2\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$.

Each beam position can then be uniquely expressed as

a large-tile anchor point $\boldsymbol{i}\mathbf{B}^+$

$+$ an offset $\boxed{\boldsymbol{j}}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$ to one of that large tile's small-tile anchor points

$+$ an offset $\boxed{\boldsymbol{k}}\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$ to one of that small tile's beam positions.

The (example) has $\boldsymbol{i} = [-1 \;\; -1]$, $\boxed{\boldsymbol{j}} = [12 \;\; 8]$, and $\boxed{\boldsymbol{k}} = [1 \;\; 1]$. Beam outputs are taken from FFT output bins in the highlighted region located by setting $\boldsymbol{i} = 0$ to address bins with $\boxed{\boldsymbol{j}}$ and $\boxed{\boldsymbol{k}}$ only.

where, if we use the optional but convenient basis-vector-tile approach of Section 5.4 above (which approach is itself based on Section 5.3.3),

$$\begin{aligned} \boldsymbol{k}' &= \lfloor \boldsymbol{k}\mathbf{R}_{\text{in}}^{-1} \rfloor, \\ \boxed{\boldsymbol{k}} &= \boldsymbol{k} - \boldsymbol{k}'\mathbf{R}_{\text{in}}. \end{aligned} \tag{60}$$

The decomposition can be determined graphically, using a diagram like that of Fig. 11, or an approach analogous to robust anchor-point index computation (41) can be used to compute small-tile anchor-point index (60) equally robustly. Either way, the values that can be taken by $\boxed{\boldsymbol{k}}$ are the $|\mathbf{R}_{\text{in}}|$ specific integer row two-vectors for which $\lfloor \boxed{\boldsymbol{k}}\mathbf{R}_{\text{in}}^{-1} \rfloor = 0$.

Second, the basis vectors of array-factor periodicity lattice $\mathbb{Z}^2\mathbf{B}^+$ can be used in a similar way to construct large beamspace tiles, as pictured in the example of Fig. 11, to tile the plane such that each tile contains one point, its anchor point, from that array-factor periodicity lattice and such that each point of intermediate lattice $\mathbb{Z}^2\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+$ is in exactly one tile. This leads to a unique decomposition of an arbitrary point from that intermediate lattice as the position of an anchor point plus an intra-tile offset in either of the two equivalent forms

$$\boldsymbol{k}'\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+ = \boldsymbol{i}\mathbf{B}^+ + \boxed{\boldsymbol{j}}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+, \tag{61}$$

$$\boldsymbol{k}' = \boldsymbol{i}\mathbf{R}_{\text{out}} + \boxed{\boldsymbol{j}} \tag{62}$$

where, again using the convenient basis-vector-tile approach of Section 5.4,

$$\begin{aligned} \boldsymbol{i} &= \lfloor \boldsymbol{k}'\mathbf{R}_{\text{out}}^{-1} \rfloor, \\ \boxed{\boldsymbol{j}} &= \boldsymbol{k}' - \boldsymbol{i}\mathbf{R}_{\text{out}}. \end{aligned} \tag{63}$$

The decomposition can be determined graphically, using a diagram like that of Fig. 11, or an approach analogous to robust anchor-point index computation (41) can be used to compute large-tile anchor-point index (63) equally robustly. Either way, the values that can be taken by $\boxed{\boldsymbol{j}}$ are the $|\mathbf{R}_{\text{out}}|$ specific integer row two-vectors for which $\lfloor \boxed{\boldsymbol{j}}\mathbf{R}_{\text{out}}^{-1} \rfloor = 0$.

Small-tile decompositions (58) and (59) can be combined with large-tile decompositions (61) and (62) to yield the decompositions of the beam position and the beam-position index vector given by

$$\boldsymbol{k}\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+ = \boldsymbol{i}\mathbf{B}^+ + \boxed{\boldsymbol{j}}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+ + \boxed{\boldsymbol{k}}\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\mathbf{B}^+, \tag{64}$$

$$\boldsymbol{k} = \boldsymbol{i}\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}} + \boxed{\boldsymbol{j}}\mathbf{R}_{\text{in}} + \boxed{\boldsymbol{k}}. \tag{65}$$

In Fig. 11 the three thin line segments leading from the origin to the one (example) beam position correspond to the three terms on the right in beam-position decomposition (64). (The fourth segment is discussed below.)

A beam position is indexed or addressed by row vector $\boldsymbol{k}$ or, using address decomposition (65), the three row vectors $\boldsymbol{i}$, $\boxed{\boldsymbol{j}}$, and $\boxed{\boldsymbol{k}}$, so in effect (65) relates two beam-position addressing schemes. It also generalizes on mixed-radix representation of an integer. Here $\boldsymbol{i}$, $\boxed{\boldsymbol{j}}$, and $\boxed{\boldsymbol{k}}$ are digit vectors from $\mathbb{Z}^2$ and can respectively take an infinite number of values, exactly $|\mathbf{R}_{\text{out}}|$ values, and exactly $|\mathbf{R}_{\text{in}}|$ values. Resampling matrices $\mathbf{R}_{\text{in}}$

and $\mathbf{R}_{\text{out}}$ are radix matrices. The most significant vector (MSV) is $\boldsymbol{i}$, and the least significant vector (LSV) is $\boxed{\boldsymbol{k}}$. When the vectors and matrices involved are replaced with integers, the procedure here for obtaining the digit vectors in fact degenerates to the usual procedure for obtaining a mixed-radix representation of an integer.

The key ideas of this beamspace tiling are summarized for review convenience in the Fig. 11 caption.

5.6.5 Two-Tiling Element and Beam Addresses Factor a Generalized DFT

The General Form

Decompositions (57) and (65) respectively express the element-position and beam-position index vectors as the three terms above and left of the double lines in Table 2, making the product $\boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n} = \boldsymbol{k}\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\boldsymbol{n}$ just the sum of the nine terms on the lower right of the double lines. The highlighted terms are integers and so do not contribute to the complex exponential

$$\mathrm{e}^{-\mathrm{j}2\pi\boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n}} = \mathrm{e}^{-\mathrm{j}2\pi\boxed{\boldsymbol{k}}\mathbf{R}_{\text{in}}^{-1}\boxed{\boldsymbol{m}}}\,\mathrm{e}^{-\mathrm{j}2\pi\boxed{\boldsymbol{k}}\underbrace{\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}}_{\mathbf{R}_{\text{steer}}^{-1}}\boxed{\boldsymbol{n}}}\,\mathrm{e}^{-\mathrm{j}2\pi\boxed{\boldsymbol{j}}\mathbf{R}_{\text{out}}^{-1}\boxed{\boldsymbol{n}}}\,.$$

Substituting this complex exponential, element-position decomposition (57), and beam-position decomposition (65) into beam computation (28) yields, after some reordering and three quite optional uses of $\mathbf{R}_{\text{steer}} = \mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}$ that simply affect how the result is written,

$$\underset{\text{beam } \boldsymbol{i}\mathbf{R}_{\text{steer}}+\boxed{\boldsymbol{j}}\mathbf{R}_{\text{in}}+\boxed{\boldsymbol{k}}}{s_0} = X_{\boxed{\boldsymbol{j}},\boxed{\boldsymbol{k}}}, \tag{66}$$

$$X_{\boxed{\boldsymbol{j}},\boxed{\boldsymbol{k}}} \triangleq \sum_{\substack{\boxed{\boldsymbol{n}} \text{ such that} \\ \lfloor \mathbf{R}_{\text{out}}^{-1}\boxed{\boldsymbol{n}} \rfloor = 0}} \left(\mathrm{e}^{-\mathrm{j}2\pi\boxed{\boldsymbol{k}}\mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}}} \sum_{\substack{\boxed{\boldsymbol{m}} \text{ such that} \\ \lfloor \mathbf{R}_{\text{in}}^{-1}\boxed{\boldsymbol{m}} \rfloor = 0}} x_{\boxed{\boldsymbol{m}},\boxed{\boldsymbol{n}}}\,\mathrm{e}^{-\mathrm{j}2\pi\boxed{\boldsymbol{k}}\mathbf{R}_{\text{in}}^{-1}\boxed{\boldsymbol{m}}} \right) \mathrm{e}^{-\mathrm{j}2\pi\boxed{\boldsymbol{j}}\mathbf{R}_{\text{out}}^{-1}\boxed{\boldsymbol{n}}}, \tag{67}$$

($|\mathbf{R}_{\text{in}}|$ generalized output DFTs (one for each $\boxed{\boldsymbol{k}}$); twiddle factor; $|\mathbf{R}_{\text{out}}|$ generalized input DFTs (one for each $\boxed{\boldsymbol{n}}$))

$$x_{\boxed{\boldsymbol{m}},\boxed{\boldsymbol{n}}} \triangleq \sum_{\boldsymbol{\ell}} \left[h_{-\boldsymbol{n}} s_{\boldsymbol{n}}^0 \right]_{\boldsymbol{n}=\mathbf{R}_{\text{steer}}\boldsymbol{\ell}+\mathbf{R}_{\text{out}}\boxed{\boldsymbol{m}}+\boxed{\boldsymbol{n}}}. \tag{68}$$

Table 2 — Terms of $\boldsymbol{k}\mathbf{R}_{\text{steer}}^{-1}\boldsymbol{n}$ When Two Tilings are Used in Each Space

$\langle$one of $\downarrow\rangle\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\langle$one of $\rightarrow\rangle$	$\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}\boldsymbol{\ell}$	$\mathbf{R}_{\text{out}}\boxed{\boldsymbol{m}}$	$\boxed{\boldsymbol{n}}$
$\boldsymbol{i}\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}$	$\boldsymbol{i}\mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}\boldsymbol{\ell}$	$\boldsymbol{i}\mathbf{R}_{\text{out}}\boxed{\boldsymbol{m}}$	$\boldsymbol{i}\boxed{\boldsymbol{n}}$
$\boxed{\boldsymbol{j}}\mathbf{R}_{\text{in}}$	$\boxed{\boldsymbol{j}}\mathbf{R}_{\text{in}}\boldsymbol{\ell}$	$\boxed{\boldsymbol{j}}\boxed{\boldsymbol{m}}$	$\boxed{\boldsymbol{j}}\mathbf{R}_{\text{out}}^{-1}\boxed{\boldsymbol{n}}$
$\boxed{\boldsymbol{k}}$	$\boxed{\boldsymbol{k}}\boldsymbol{\ell}$	$\boxed{\boldsymbol{k}}\mathbf{R}_{\text{in}}^{-1}\boxed{\boldsymbol{m}}$	$\boxed{\boldsymbol{k}}\mathbf{R}_{\text{in}}^{-1}\mathbf{R}_{\text{out}}^{-1}\boxed{\boldsymbol{n}}$

Here (68) in effect addresses elements using two column-vector digits to generalize relationship (37) governing the folding (data turning) of weighted element outputs into FFT input bins. Very general FFT factorization (67) then transforms the doubly indexed input-bin signals $x_{\boxed{m},\boxed{n}}$ of (68) to create doubly indexed output-bin signals $X_{\boxed{j},\boxed{k}}$. The latter formally become beam outputs in (66), which is a generalization of (42) to address beams with two row-vector digits.

It is natural to suppose that the choice of tiling strategies somehow is key here, but it isn't actually so. Factorization $\mathbf{R}_{\text{steer}} = \mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}$ implies dual intermediate lattices in position space and beamspace, and the selection of that dual pair of lattices indirectly designs FFT factorization (67). Choosing tilings helps us get from the intermediate lattice to this factorization, but the factorization that results doesn't actually depend on the specific tilings chosen. Using basis-vector tilings led to expressions $\lfloor \mathbf{R}_{\text{out}}^{-1}\boxed{n} \rfloor = 0$ and $\lfloor \mathbf{R}_{\text{in}}^{-1}\boxed{m} \rfloor = 0$ in FFT (67), but that tiling choice was purely for convenience. It gave us a simple way to compute and label indices. Prototile references more general in form could have been used, however, without changing the resulting FFT structure in any way, because the specific tilings chosen actually affect bin labeling only. It is the intermediate lattice used that determines the FFT factorization itself.

The Block Diagram

FFT expression (67) is presented as a 3D block diagram in Fig. 12 for the example case in which

$$|\mathbf{R}_{\text{in}}| = 3 \text{ so that } \begin{cases} \text{input MSV } \boxed{m} \in \{0, \boxed{m}^1, \boxed{m}^2\} \subset \mathbb{Z}^2, \\ \text{output LSV } \boxed{k} \in \{0, \boxed{k}^1, \boxed{k}^2\} \subset \mathbb{Z}^2, \end{cases}$$

$$|\mathbf{R}_{\text{out}}| = 4 \text{ so that } \begin{cases} \text{input LSV } \boxed{n} \in \{0, \boxed{n}^1, \boxed{n}^2, \boxed{n}^3\} \subset \mathbb{Z}^2, \\ \text{output MSV } \boxed{j} \in \{0, \boxed{j}^1, \boxed{j}^2, \boxed{j}^3\} \subset \mathbb{Z}^2. \end{cases}$$

This diagram is very similar in appearance to Fig. 9, and the two systems have in common that they express a large DFT in terms of smaller ones. In other ways the two systems are quite different, at least if Fig. 12 is considered in its full generality.

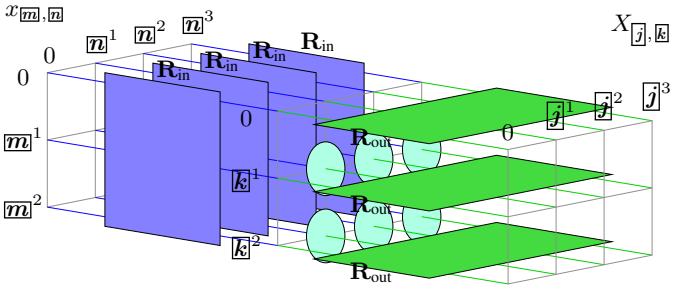


Fig. 12 — General FFT digital steering structure, here with $|\mathbf{R}_{\text{in}}| = 3$ and $|\mathbf{R}_{\text{out}}| = 4$, transforms weighted and folded element outputs $x_{\boxed{m},\boxed{n}}$ into beam outputs $X_{\boxed{j},\boxed{k}}$ in steps:

1. a **generalized DFT** for each $\boxed{n}$ transforms position index $\boxed{m}$ to beamspace index $\boxed{k}$ using $\mathbf{R}_{\text{in}}$,
2. a twiddle-factor multiply per $\boxed{k}$, $\boxed{n}$ combination using $\mathbf{R}_{\text{in}}$ and $\mathbf{R}_{\text{out}}$, a no-op when $\boxed{k}=0$ or $\boxed{n}=0$, and
3. **a generalized output DFT** for each $\boxed{k}$ transforms position index $\boxed{n}$ to beamspace index $\boxed{j}$ using $\mathbf{R}_{\text{out}}$.

In Fig. 9 all component transforms were 1D FFTs with integer-indexed bins, but here in Fig. 12 each component transform is a generalized DFT with inputs and outputs indexed by vectors from $\mathbb{Z}^2$ and $\mathbb{Z}^2$ respectively. In the example application addressed by the maps of position space and beamspace in Figs. 10

and 11, index vectors $\boxed{\boldsymbol{m}}$ and $\boxed{\boldsymbol{j}}$ select small tiles related to an intermediate lattice, and index vectors $\boxed{\boldsymbol{n}}$ and $\boxed{\boldsymbol{k}}$ select bins from inside those tiles. This is very different in principle from the row/column indexing of Fig. 9.

Here the component transforms' vertical and horizontal orientations in the diagram and the arrangement of input and output bins on grids need bear no particular relation to the geometrical positioning of input and output bins in position space and beamspace. In Fig. 12 the row-column arrangement of inputs and outputs is largely just a drawing convenience to help make the factorization comprehensible, though there is a pattern to it: for both input and output transforms changing the least significant digit vector steps from one component transform to another, while changing the more significant digit vector steps from bin to bin within a single component transform.

Finally, in FFT (67) and in Fig. 12, multiplications by twiddle factors —this is actually the traditional name in the FFT literature—come between the two layers of smaller DFTs. This twiddle factor is not shown when $\boxed{\boldsymbol{k}}=0$ or $\boxed{\boldsymbol{n}}=0$ because in those cases its value $\mathrm{e}^{-\mathrm{j}2\pi \boxed{\boldsymbol{k}}\mathbf{R}_{\text{steer}}^{-1}\boxed{\boldsymbol{n}}} = 1$.

5.7 Three Key Special Cases

Technically the DFT factorization into smaller DFTs of Fig. 12 can only be called an FFT if each of its component DFTs is either itself an FFT, of one kind or another, or a very small DFT, small enough that using an FFT approach to calculating it would be unwarranted. Three specializations of particular interest correspond to three specific factorizations.

5.7.1 The Simple Case

The simple case of Section 5.5 above results from substitutions

$$\mathbf{R}_{\text{out}} = \begin{bmatrix} 1 & 0 \\ 0 & N_{\text{ex}} \end{bmatrix}, \qquad \boxed{\boldsymbol{j}} = \begin{bmatrix} 0 & r_2 \end{bmatrix},$$
$$\boxed{\boldsymbol{m}} = \begin{bmatrix} m_1 \\ 0 \end{bmatrix}, \qquad \mathbf{R}_{\text{in}} = \begin{bmatrix} N_{\text{ex}} & 0 \\ 0 & 1 \end{bmatrix},$$
$$\boxed{\boldsymbol{n}} = \begin{bmatrix} 0 \\ m_2 \end{bmatrix}, \qquad \boxed{\boldsymbol{k}} = \begin{bmatrix} r_1 & 0 \end{bmatrix}$$

where each of r_1, r_2, m_1, and m_2 range over the integers $0, \ldots, N_{\text{ex}} - 1$. These turn the general FFT realization presented here in (67) and Fig. 12 into Section 5.5's special-case FFT realization of (45) and Fig. 9, where input and output FFTs are just ordinary 1D FFTs of N_{ex} points each.

5.7.2 Coarse and Fine Beam-Spacing Modes

Coarse and fine beam-spacing modes can be realized using an extra small-determinant factor as the rightmost factor in $\mathbf{R}_{\text{steer}}$ as in the example of (48). These substitutions make $\mathbf{R}_{\text{in}}$ such an "extra" factor

with $\mathbf{R}_{\text{out}}$ the "main portion" of $\mathbf{R}_{\text{steer}}$.

$$\mathbf{R}_{\text{out}} = \begin{bmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{bmatrix}, \qquad \boxed{\boldsymbol{j}} = \begin{bmatrix} r_1 & r_2 \end{bmatrix},$$
$$\boxed{\boldsymbol{m}} \in \left\{0, \boxed{\boldsymbol{m}}^1, \ldots, \boxed{\boldsymbol{m}}^{|\mathbf{R}_{\text{in}}|-1}\right\}, \qquad |\mathbf{R}_{\text{in}}| \quad \text{small},$$
$$\boxed{\boldsymbol{n}} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \qquad \boxed{\boldsymbol{k}} \in \left\{0, \boxed{\boldsymbol{k}}^1, \ldots, \boxed{\boldsymbol{k}}^{|\mathbf{R}_{\text{in}}|-1}\right\}.$$

This gives the general FFT structure of Fig. 12 a large number $|\mathbf{R}_{\text{out}}| = N_{\text{ex}}^2$ of small input FFTs and a small number $|\mathbf{R}_{\text{in}}|$ of large output FFTs. Each of the latter large FFTs are implemented as in Fig. 9 or, equivalently, using **the simple case** described above. Each of r_1, r_2, m_1, and m_2 here again range over the integers $0, \ldots, N_{\text{ex}}-1$, and the mismatch here between these variable names and the names of the vectors they make up is simply for compatibility with Fig. 9.

Here digit vectors $\boxed{\boldsymbol{m}}$ and $\boxed{\boldsymbol{k}}$ each take $|\mathbf{R}_{\text{in}}|$ specific values, one of which is a zero vector. The other specific values can be determined graphically, as illustrated in Figs. 10 and 11 for $|\mathbf{R}_{\text{in}}| = 3$, or a somewhat larger set determined graphically can be thinned down computationally to keep only those $\boxed{\boldsymbol{m}}$ and $\boxed{\boldsymbol{k}}$ for which $\lfloor \mathbf{R}_{\text{out}}^{-1} \boxed{\boldsymbol{m}} \rfloor = 0$ and $\lfloor \boxed{\boldsymbol{k}} \mathbf{R}_{\text{in}}^{-1} \rfloor = 0$ respectively, with these floor tests computed using the robust approach of (35) and (41) that was derived above in Section 5.3.4.

The example design presented Figs. 10 and 11, for example, would be realized using Fig. 12 with $16^2 = 256$ three-input, three-output generalized DFT blocks as input FFTs and three 16×16 point 2D FFTs as output FFTs. Each of those three 2D output FFTs would be realized using 32 1D FFTs of 16 points each using the architecture of Fig. 9. The Appendix shows that the three-point generalized DFTs are actually ordinary 1D DFT (or FFT) blocks.

5.7.3 Efficient Computation of Beam Clusters

Efficient computation of beam clusters is often possible using either the first, **simple case** above or using an $\mathbf{R}_{\text{steer}} = \mathbf{R}_{\text{out}}\mathbf{R}_{\text{in}}$ factorization in which $\mathbf{R}_{\text{out}}$ is the small "extra" factor, basically reversing the roles of $\mathbf{R}_{\text{in}}$ and $\mathbf{R}_{\text{out}}$ relative to the case just discussed. Now

$$|\mathbf{R}_{\text{out}}| \quad \text{small}, \qquad \boxed{\boldsymbol{j}} \in \left\{0, \boxed{\boldsymbol{j}}^1, \ldots, \boxed{\boldsymbol{j}}^{|\mathbf{R}_{\text{out}}|-1}\right\},$$
$$\boxed{\boldsymbol{m}} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \qquad \mathbf{R}_{\text{in}} = \begin{bmatrix} N_{\text{ex}} & 0 \\ 0 & N_{\text{ex}} \end{bmatrix},$$
$$\boxed{\boldsymbol{n}} \in \left\{0, \boxed{\boldsymbol{n}}^1, \ldots, \boxed{\boldsymbol{n}}^{|\mathbf{R}_{\text{out}}|-1}\right\}, \qquad \boxed{\boldsymbol{k}} = \begin{bmatrix} r_1 & r_2 \end{bmatrix}.$$

The difference between this case and the last is one of input-bin and output-bin geometry in position space and beamspace. The last case corresponded to Figs. 10 and 11, in which input bins were structured into a few giant blocks and output bins were structured into many small ones. Here that is reversed so that input bins are structured into many small blocks and output bins into a few large blocks. This is convenient for computing a cluster of beams, because beams near each other in beamspace, such as those in a cluster, are often computed near each other in the same large block of output bins. This makes it easy to improve

computational efficiency by deleting entire unused rows or columns of output bins in the large-block FFTs. Examples of such systems are presented in Ref. 3.

6. SUMMARY AND OBSERVATIONS

This document used a simple theoretical formulation of a planar array of identical elements positioned on points of a lattice and surrounded by guard elements to develop many-beam phase-shift steering using a generalized Cooley-Tukey FFT structure to realize a generalized 2D DFT, one in which the usual number of DFT points has been replaced with a factorable 2×2 resampling matrix $\mathbf{R}_{\text{steer}}$.

In discussions of FFT beamsteering, four specific questions typically arise, two that are about the FFT itself and two that concern its application to radar systems.

1. *Are these "decimation in time" or "decimation in frequency" FFT structures?*

   Such names only apply here beginning in Section 5.6, when an intermediate lattice is first used, and even then only when one of $|\mathbf{R}_{\text{in}}|$ and $|\mathbf{R}_{\text{out}}|$ is small enough that the corresponding resampling matrix is used without further factorization, when the associated DFTs are realized directly from general expression (43). In that case the classification is according to which of $|\mathbf{R}_{\text{in}}|$ or $|\mathbf{R}_{\text{out}}|$ is large and which is small. In Table 3 position corresponds to the usual "time," and beamspace corresponds to the usual "frequency."

Table 3 — Decimation in. . . What?

structure	$\lvert\mathbf{R}_{\text{in}}\rvert$	$\lvert\mathbf{R}_{\text{out}}\rvert$	*examples*
decimation in position	large	small	Section 5.7.3: Efficient Computation of Beam Clusters
decimation in beamspace	small	large	Section 5.7.2: Coarse and Fine Beam-Spacing Modes, example factorization (48), example tilings of Figs. 10 and 11

2. *What is the purpose of the tiles?*

   The short version is that large input tiles are about input folding (data turning), large output tiles are about beam-position periodicity, and small input and output tiles are what enable the DFT to be turned into an FFT through factorization. The general tiling rules retain the full design flexibility inherent in the DFT and FFT mathematics while organizing matters graphically to make comprehension less arduous.

   The longer version goes something like this.

   Large input tiles structure the discrete-space Fourier transform into a finite DFT sum and an infinite input-folding sum, in the process giving the DFT a finite number of input bins, by making product $\mathbf{km}$ an integer in the derivation of the multibeam computation of (36) and (37). Further, they represent the most general way of making that happen. The tiles then imply an MSV-LSV addressing scheme that maps any element location to a combination of an DFT input bin and an input-folding (data turning) term.

Large output tiles eliminate redundant DFT output bins by making product $\boldsymbol{j}\boxed{\boldsymbol{n}}$ an integer in the derivation of the finite set of beams computed in (42) and (43). They represent the most general way of making that happen. The tiles then imply an MSV-LSV addressing scheme that maps any beam position to a combination of an DFT output bin and a steering-vector period.

Small tiles add "intermediate-significance vectors" to create MSV-ISV-LSV element-location addressing and MSV-ISV-LSV beam-position addressing in such a way that the DFT factors into LSV-labeled input FFT blocks with ISV input-bin labeling followed by (twiddle-factor multiplications and) LSV-labeled output FFT blocks with ISV output-bin labeling. Fundamentally, the tiling scheme enables this factorization by making $\boxed{\boldsymbol{j}}\boxed{\boldsymbol{m}}$, the middle term of the nine tabulated in Table 2, an integer. It is the most general way of making that happen. The specific FFT factorization realized is determined by the choice of the intermediate lattice and its dual used in the construction of the small tiles.

3. *What are the main limitations of the approach?*

   The key limitation of the approach is that the positions of the synthesized beams in beamspace (sine space, give or take a scale factor) must be drawn from a superlattice of the array-factor periodicity lattice, with the two lattices related by $\mathbf{R}_{\text{steer}}$. Further, for an FFT structure to be viable $\mathbf{R}_{\text{steer}}$ must be factorable into resampling matrices with determinants small in absolute value. The FFT incorporates a specific choice of $\mathbf{R}_{\text{steer}}$ in a deep, structural way, so generally $\mathbf{R}_{\text{steer}}$ cannot be arbitrarily changed in real time and therefore neither can the steering lattice of beam positions. At most there might be flexibility to switch between steering lattices, particularly between related ones, for example as in the case of "coarse and fine beam-spacing modes" discussed near the end of the last section. But these restrictions do mean that not even the beamspace spacing of the steering-lattice beams can be freely chosen. Enough viable choices of $\mathbf{R}_{\text{steer}}$ are available, however, that respecting these limits seldom cripples system design, as long as specific beam positions are less important than having the spacing between them be small enough without being too small.

4. *When this approach is used for a shipboard array, can it be modified to include electronic ship-motion compensation?*

   Yes and no. Yes, it is easy enough to replace element outputs s^0 with $s^0 \curvearrowright \boldsymbol{f}_\Delta^{\text{comp}}$ everywhere, so that specific element output $s^0_{\boldsymbol{n}}$ is replaced with $s^0_{\boldsymbol{n}}\, \mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{f}_\Delta^{\text{comp}} \boldsymbol{n}}$ using one extra complex multiply per element to effectively translate the entire lattice of beam positions by $\mathbf{k}_\Delta^{\text{comp}} = \boldsymbol{f}_\Delta^{\text{comp}} \mathbf{B}^+$ to compensate for motion. But no, this is not complete compensation. One beam, at least if it is circularly symmetric in beamspace, can be compensated by a simple translation in this way. But translation is not enough when a family of beams on lattice points is involved, because there is no way to compensate for ship rotation, for example ship roll when a radar face's boresight vector points forward or ship pitch when a radar face's boresight vector points abeam. Even when motions are small enough that the trigonometric distortions involved in mapping from ship-motion angles to changes in beamspace position $\mathbf{k}$ are not significant, there is a problem: rotation in beamspace of the entire set of beams would be needed for complete motion compensation, and this is simply not possible. Scatterer positions must simply be estimated in ship coordinates, converted to Earth coordinates, and passed in that form to the tracking system.

REFERENCES

1. J. O. Coleman, "A generalized FFT for many simultaneous receive beams," Naval Research Laboratory, NRL Report MR/5320--07-9029, June 29, 2007.

2. D. E. Dudgeon and R. M. Mersereau, *Multidimensional Digital Signal Processing*, 3rd ed., ser. Prentice-Hall Signal Processing (Prentice-Hall, http://www.prenhall.com/, Upper Saddle River, NJ, 1983), out of print.

3. J. O. Coleman, "Multi-layer overlapped subarrays for phased-array volume-search radars with FFT-realized clustered beams," Naval Research Laboratory, NRL Report (number TBD), est. mid-2011, in preparation.

4. R. Mersereau and T. Speake, "A unified treatment of Cooley-Tukey algorithms for the evaluation of the multidimensional DFT," *IEEE Trans. Acoust., Speech, Signal Process.*, **29**(5), 1011–1018, Oct. 1981.

5. A. Guessoum and R. Mersereau, "Fast algorithms for the multidimensional discrete Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, **34**(4), 937–943, Aug. 1986.

6. J. O. Coleman, K. R. McPhail, P. E. Cahill, and D. P. Scholnik, "Efficient subarray realization through layering," in *Proc. 2005 Antenna Applications Symp.*, Monticello IL, USA, Sept. 21–23, 2005, org. by U. Massachusetts Amherst (http://www.ecs.umass.edu/ece/allerton/). Obtain at http://www.dtic.mil/

7. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. (McGraw-Hill, http://mcgraw-hill.com/, New York, 2002).

Appendix

A PRIME |R| MAKES A GENERALIZED 2D DFT INTO AN ORDINARY 1D DFT

INTRODUCTION

A DFT Configuration Seen Throughout This Document

Every DFT seen heretofore in this document, whether a "top level" DFT or a component DFT inside an FFT expression, is of the form

$$X_{\boxed{\boldsymbol{k}}} = \sum_{\substack{\boxed{\boldsymbol{n}} \in \mathbb{Z}^2 \text{ such that} \\ \mathbf{B}_{\text{loc}}\boxed{\boldsymbol{n}} \in \langle \text{input prototile} \rangle}} x_{\boxed{\boldsymbol{n}}} \, \mathrm{e}^{-\mathrm{j}2\pi \boxed{\boldsymbol{k}} \mathbf{R}^{-1} \boxed{\boldsymbol{n}}} \tag{A1}$$

for some basis matrix $\mathbf{B}_{\text{loc}}$ and some resampling matrix $\mathbf{R}$, where $\boxed{\boldsymbol{k}}$, $x_{\boxed{\boldsymbol{n}}}$, and $X_{\boxed{\boldsymbol{k}}}$ are variables local to this discussion and may or may not be the specific array-related values from before. Here $\langle$input prototile$\rangle$ is a position-space tile containing the origin such that tiling all of position space with its translates gives each tile exactly one anchor point from sublattice $\mathbf{B}_{\text{loc}}\mathbf{R}\mathbb{Z}^2$ and exactly $|\mathbf{R}|$ points from lattice $\mathbf{B}_{\text{loc}}\mathbb{Z}^2$. Likewise, we have supposed $\boxed{\boldsymbol{k}} \in \mathbb{Z}^2$ such that $\boxed{\boldsymbol{k}}\mathbf{B}_{\text{loc}}^{+} \in \langle$output prototile$\rangle$, where $\langle$output prototile$\rangle$ is a beamspace tile containing the origin and where tiling all of beamspace with its translates gives each tile exactly one anchor point from $\mathbb{Z}^2\mathbf{B}_{\text{loc}}^{+}$ and exactly $|\mathbf{R}|$ possible intra-tile offsets from superlattice $\mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^{+}$. We often used basis-vector tiling in particular, which leads to prototile descriptions

$$\boxed{\boldsymbol{n}} \text{ such that } \lfloor \quad \mathbf{R}^{-1}\boxed{\boldsymbol{n}} \rfloor = 0 \, , \tag{A2}$$

$$\boxed{\boldsymbol{k}} \text{ such that } \lfloor \boxed{\boldsymbol{k}}\mathbf{R}^{-1} \quad \rfloor = 0 \tag{A3}$$

but those particular structures are completely optional.

The Result to be Proved

When $|\mathbf{R}|$ is prime, the sets of allowed index vectors $\boxed{\boldsymbol{n}}$ and $\boxed{\boldsymbol{k}}$ for the intra-tile offsets can each be put in some order

$$\{\text{allowed } \boxed{\boldsymbol{n}}\} = \{\boxed{\boldsymbol{n}}^0, \boxed{\boldsymbol{n}}^1, \boxed{\boldsymbol{n}}^2, \ldots, \boxed{\boldsymbol{n}}^{|\mathbf{R}|-1}\}, \tag{A4}$$

$$\{\text{allowed } \boxed{\boldsymbol{k}}\} = \{\boxed{\boldsymbol{k}}^0, \boxed{\boldsymbol{k}}^1, \boxed{\boldsymbol{k}}^2, \ldots, \boxed{\boldsymbol{k}}^{|\mathbf{R}|-1}\} \tag{A5}$$

such that, if we use x_n for $x_{\boxed{\boldsymbol{n}}^n}$, use X_k for $X_{\boxed{\boldsymbol{k}}^k}$, and let $N = |\mathbf{R}|$, the generalized 2D DFT in (A1) becomes the ordinary one-dimensional DFT

$$X_k = \sum_{n=0}^{N-1} x_n \, \mathrm{e}^{-\mathrm{j}2\pi \frac{kn}{N}} \, .$$

This allows the implementer the convenience of drawing on canned FFT routines, since for a prime number of points the FFT is just a straightforward DFT sum.

The proof offered below is constructive, proving not only that such orderings exist but providing a method for determining them. In fact $|\mathbf{R}|-1$ choices of these orderings are provided.

A BACKGROUND FACT

The proof below hinges on this result from elementary number theory.

Bézout's identity. If k_1 and k_2 are nonzero integers with $\gcd(k_1, k_2)$ their greatest common divisor, there exist integers n_1 and n_2 such that

$$k_1 n_1 + k_2 n_2 = \gcd(k_1, k_2).$$

One special case is of particular interest. If integer $M > 1$ with M prime, then $\gcd(k, M) = 1$ for any k that is not a multiple of M or, equivalently, for which $k \bmod M \neq 0$. Bézout's identity then tells us that we can find integers $-\ell$ and n such that these three equivalent statements hold.

$$\begin{aligned} kn - \ell M &= 1, \\ kn &= \ell M + 1, \\ kn \bmod M &= 1. \end{aligned}$$

The latter reveals that Bézout's identity is telling us that every k that is nonzero modulo M has a multiplicative inverse modulo M. Of course this also follows from another algebraic fact, that the integers modulo a prime form a field.

Multiplicative inverses or, more generally, Bézout coefficients can always be found with the extended Euclidean algorithm [7], but when the solutions sought are restricted to the modulo M integers with M small, as is generally the case in this application, a brute-force computational search—just try all possible solutions—is a simpler and quite reasonable alternative.

THE PROOF

First: Express the Elements of $\mathbf{R}^{-1}$ as Rational Numbers

Recall that the inverse of a nonsingular matrix $\mathbf{R}$ can be expressed in terms of its adjugate matrix and its determinant as

$$\mathbf{R}^{-1} = \frac{1}{\det(\mathbf{R})} \operatorname{adj}(\mathbf{R}). \tag{A6}$$

Of course any number is the product of its absolute value and sign, so $\det(\mathbf{R}) = |\mathbf{R}| \operatorname{sgn}(\det(\mathbf{R})) = |\mathbf{R}| / \operatorname{sgn}(\det(\mathbf{R}))$. Combining with (A6) allows us to write

$$\begin{aligned} \mathbf{R}^{-1} &= \frac{1}{|\mathbf{R}|}\mathbf{R}_i, \\ \mathbf{R}_i &\triangleq \operatorname{sgn}(\det(\mathbf{R})) \operatorname{adj}(\mathbf{R}). \end{aligned} \tag{A7}$$

Resampling matrix $\mathbf{R}$ is an integer matrix and therefore so is $\mathbf{R}_i$.

Second: Tile Beamspace, Pick a Seed Point in Beamspace, and Analyze That Point's Coordinates

Lattice $\mathbb{Z}^2\mathbf{B}_{\text{loc}}^+$ has a superlattice $\mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^+$. Partition the plane into identical tiles, each containing one anchor point of lattice $\mathbb{Z}^2\mathbf{B}_{\text{loc}}^+$. Then each tile contains exactly $|\mathbf{R}|$ points of superlattice $\mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^+$. Choose any "seed" point $\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^+$ in $\mathbb{Z}^2\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^+$ that is not in $\mathbb{Z}^2\mathbf{B}_{\text{loc}}^+$. Of course there are $|\mathbf{R}|-1$ possible choices of seed point in each tile, and this is ultimately why we end up with $|\mathbf{R}|-1$ choices of orderings (A4) and (A5). Using this new seed, let

$$\boldsymbol{k}^{\text{mod}} = \boldsymbol{k}^{\text{seed}}\mathbf{R}_i \bmod |\mathbf{R}|, \tag{A8}$$

with the modulo operation taken on each element of the row vector $\boldsymbol{k}^{\text{seed}}\mathbf{R}_i$ separately. That definition of $\boldsymbol{k}^{\text{mod}}$ is equivalent to the first of these, and the second then follows from (A7):

$$\begin{aligned} \boldsymbol{k}^{\text{seed}}\mathbf{R}_i &= \boldsymbol{j}|\mathbf{R}| + \boldsymbol{k}^{\text{mod}} \quad \text{for some } \boldsymbol{j} \in \mathbb{Z}^2, \\ \boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1} &= \boldsymbol{j} + \frac{1}{|\mathbf{R}|}\boldsymbol{k}^{\text{mod}} \quad \text{(for that same } \boldsymbol{j}\text{)}. \end{aligned} \tag{A9}$$

In the latter we express the elements of row vector $\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}$ as the sum of their integer and fractional parts. All-integer equivalent (A9) is easier to work with below, however. While we could easily compute row vector $\boldsymbol{j}$, actually doing so is unnecessary. It's enough to know that such a vector exists.

One useful fact about $\boldsymbol{k}^{\text{mod}}$ is easily established. Since $\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^+$ is not in $\mathbb{Z}^2\mathbf{B}_{\text{loc}}^+$, equivalent statements

$$\begin{aligned} \boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^+ &\neq \boldsymbol{j}\mathbf{B}_{\text{loc}}^+, \\ \boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1} &\neq \boldsymbol{j}, \\ \boldsymbol{k}^{\text{seed}}\mathbf{R}_i &\neq \boldsymbol{j}|\mathbf{R}| \end{aligned}$$

all hold. Comparing the last of these to (A9) reveals that $\boldsymbol{k}^{\text{mod}} \neq [0 \;\; 0]$.

Third: Construct a Seed Point in Position Space with a Required Property

We can now construct a nonzero seed point $\mathbf{B}_{\text{loc}}\boldsymbol{n}^{\text{seed}}$ in position space in such a way that these equivalent statements are true.

$$\boldsymbol{k}^{\text{seed}}\mathbf{R}_i \;\; \boldsymbol{n}^{\text{seed}} = (\boldsymbol{j}\,\boldsymbol{n}^{\text{seed}} + \ell)|\mathbf{R}| + \;\; 1 \quad \text{for some integer } \ell, \tag{A10}$$

$$\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\boldsymbol{n}^{\text{seed}} = \;\; \boldsymbol{j}\,\boldsymbol{n}^{\text{seed}} + \ell \qquad + \frac{1}{|\mathbf{R}|} \;\; \text{(for that same } \ell\text{)}. \tag{A11}$$

The second statement decomposes the scalar $\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\boldsymbol{n}^{\text{seed}}$ into integer and fractional parts. What is important to us is the particular value $1/|\mathbf{R}|$ of that fractional part. It is easiest in establishing this, however, to aim for (A10), the all-integer statement of the same result. Consider two distinct cases, according to the form of $\boldsymbol{k}^{\text{mod}}$ from (A8).

Case I: $\boldsymbol{k}^{\text{mod}} = [k_1 \;\; k_2]$ with exactly one of k_1 and k_2 equal to zero.

Suppose one of k_1 and k_2 has value 0 and the other has value k with $0 < k < |\mathbf{R}|$. Let n with $0 < n < |\mathbf{R}|$ be the multiplicative inverse of k modulo $|\mathbf{R}|$ so that $kn = 1 + \ell|\mathbf{R}|$ for some integer ℓ. Let $\boldsymbol{n}^{\text{seed}} = \left[\begin{smallmatrix} n \\ 0 \end{smallmatrix}\right]$ or $\boldsymbol{n}^{\text{seed}} = \left[\begin{smallmatrix} 0 \\ n \end{smallmatrix}\right]$ according as $\boldsymbol{k}^{\text{mod}} = [k \;\; 0]$ or $\boldsymbol{k}^{\text{mod}} = [0 \;\; k]$. Using (A9) establishes (A10):

$$\begin{aligned} \boldsymbol{k}^{\text{seed}} \mathbf{R}_i \boldsymbol{n}^{\text{seed}} &= \boldsymbol{j}\, \boldsymbol{n}^{\text{seed}} |\mathbf{R}| + \boldsymbol{k}^{\text{mod}} \boldsymbol{n}^{\text{seed}} \\ &= \boldsymbol{j}\, \boldsymbol{n}^{\text{seed}} |\mathbf{R}| + kn \\ &= \boldsymbol{j}\, \boldsymbol{n}^{\text{seed}} |\mathbf{R}| + (1 + \ell|\mathbf{R}|). \end{aligned}$$

Case II: $\boldsymbol{k}^{\text{mod}} = [k_1 \;\; k_2]$ with both k_1 and k_2 nonzero.

Special case: $\gcd(k_1, k_2) = 1$. This is a special case of the general case considered next, and its proof is included only as an easier-to-follow preview of the proof of the general case to follow.

By Bézout's identity there are integers n_1 and n_2, readily found in fact, such that $k_1 n_1 + k_2 n_2 = 1$. Let $\boldsymbol{n}^{\text{seed}} = \left[\begin{smallmatrix} n_1 \\ n_2 \end{smallmatrix}\right]$. Then, by (A9),

$$\boldsymbol{k}^{\text{seed}} \mathbf{R}_i \boldsymbol{n}^{\text{seed}} = \boldsymbol{j}\, \boldsymbol{n}^{\text{seed}} |\mathbf{R}| + k_1 n_1 + k_2 n_2$$

establishing (A10) with $\ell = 0$.

General case: Let

$$\begin{aligned} k_1' &= k_1 / \gcd(k_1, k_2), \\ k_2' &= k_2 / \gcd(k_1, k_2) \end{aligned}$$

so that integers k_1' and k_2' have $\gcd(k_1', k_2') = 1$ and

$$\begin{aligned} k_1 &= k_1' \gcd(k_1, k_2), \\ k_2 &= k_2' \gcd(k_1, k_2). \end{aligned}$$

Let n^{gcd} be the multiplicative inverse of $\gcd(k_1, k_2)$ modulo $|\mathbf{R}|$. Then there is some integer ℓ such that $\gcd(k_1, k_2) n^{\text{gcd}} = 1 + \ell|\mathbf{R}|$, which with the above gives

$$\begin{aligned} k_1 n^{\text{gcd}} &= k_1' \gcd(k_1, k_2) n^{\text{gcd}} = k_1'(1 + \ell|\mathbf{R}|), \\ k_2 n^{\text{gcd}} &= k_2' \gcd(k_1, k_2) n^{\text{gcd}} = k_2'(1 + \ell|\mathbf{R}|). \end{aligned}$$

By Bézout's identity there are integers n_1 and n_2, readily found in fact, such that $k_1' n_1 + k_2' n_2 = 1$. Let $\boldsymbol{n}^{\text{seed}} = \left[\begin{smallmatrix} n^{\text{gcd}} n_1 \\ n^{\text{gcd}} n_2 \end{smallmatrix}\right]$. By (A9) and then the above,

$$\begin{aligned} \boldsymbol{k}^{\text{seed}} \mathbf{R}_i \boldsymbol{n}^{\text{seed}} &= \boldsymbol{j}\, \boldsymbol{n}^{\text{seed}} |\mathbf{R}| + k_1 n^{\text{gcd}} n_1 + k_2 n^{\text{gcd}} n_2 \\ &= \boldsymbol{j}\, \boldsymbol{n}^{\text{seed}} |\mathbf{R}| + (k_1' n_1 + k_2' n_2)(1 + \ell|\mathbf{R}|) \end{aligned}$$

establishing (A10).

Fourth: Construct Invertible Functions from the Modulo $|\mathbf{R}|$ Integers to the Vectors in the Prototype Tiles

For each $n \in \{0, \ldots, |\mathbf{R}|-1\}$ find the anchor point $\mathbf{B}_{\text{loc}}\mathbf{R}\boldsymbol{m}$ and the intra-tile offset $\mathbf{B}_{\text{loc}}\boxed{\boldsymbol{n}}$ such that

$$n\mathbf{B}_{\text{loc}}\boldsymbol{n}^{\text{seed}} = \mathbf{B}_{\text{loc}}\mathbf{R}\boldsymbol{m} + \mathbf{B}_{\text{loc}}\boxed{\boldsymbol{n}}, \tag{A12}$$

$$n\boldsymbol{n}^{\text{seed}} = \quad \mathbf{R}\boldsymbol{m} + \quad \boxed{\boldsymbol{n}} \tag{A13}$$

where the second is of course obtained by the first by premultiplying by $\mathbf{B}_{\text{loc}}^{+}$ and using identity $\mathbf{B}_{\text{loc}}^{+}\mathbf{B}_{\text{loc}} = \mathbf{I}$. Suppose the value of function $\phi(n)$ is given by $\boxed{\boldsymbol{n}}$ of the above decomposition.

Is function $\phi(n)$ invertible? Given some allowed $\boxed{\boldsymbol{n}}$ and relationship $\phi(n) = \boxed{\boldsymbol{n}}$, is n established unambiguously? It is if we can show that $\phi(n^1) = \phi(n^2)$ implies $n^1 = n^2$. Suppose $\phi(n^1) = \boxed{\boldsymbol{n}} = \phi(n^2)$ so that

$$n^1\boldsymbol{n}^{\text{seed}} = \mathbf{R}\boldsymbol{m}^1 + \boxed{\boldsymbol{n}},$$
$$n^2\boldsymbol{n}^{\text{seed}} = \mathbf{R}\boldsymbol{m}^2 + \boxed{\boldsymbol{n}}.$$

Subtract the two equations to obtain, in three equivalent forms,

$$(n^1 - n^2) \quad \boldsymbol{n}^{\text{seed}} = \mathbf{R}\,(\boldsymbol{m}^1 - \boldsymbol{m}^2),$$
$$(n^1 - n^2)\mathbf{R}^{-1}\boldsymbol{n}^{\text{seed}} = \quad (\boldsymbol{m}^1 - \boldsymbol{m}^2),$$
$$(n^1 - n^2)\mathbf{R}_i \quad \boldsymbol{n}^{\text{seed}} = |\mathbf{R}|(\boldsymbol{m}^1 - \boldsymbol{m}^2).$$

By the last of these $|\mathbf{R}|$ is a factor of both elements of vector $(n^1 - n^2)\mathbf{R}_i\boldsymbol{n}^{\text{seed}}$, which means, since $|\mathbf{R}|$ is prime and cannot be split, it must either be a factor of integer $n^1 - n^2$ or of both elements of vector $\mathbf{R}_i\boldsymbol{n}^{\text{seed}}$. But the latter would make integers of both elements of $\frac{1}{|\mathbf{R}|}\mathbf{R}_i\boldsymbol{n}^{\text{seed}} = \mathbf{R}^{-1}\boldsymbol{n}^{\text{seed}}$, and this would in turn require $\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\boldsymbol{n}^{\text{seed}}$ to be an integer and so contradict (A11). Therefore $|\mathbf{R}|$ divides integer $n^1 - n^2$. But n^1 and n^2 are each restricted to $0, \ldots, |\mathbf{R}|-1$, so $n^1 - n^2 \in \{-(|\mathbf{R}|-1), \ldots, |\mathbf{R}|-1\}$. The only multiple of $|\mathbf{R}|$ in the latter set is zero, so $n^1 - n^2 = 0$ or $n^1 = n^2$.

The correspondence between n and position-space vector $\boxed{\boldsymbol{n}}$ just established has a counterpart in a relationship between k and beamspace vector $\boxed{\boldsymbol{k}}$. Suppose that for each $k \in \{0, \ldots, |\mathbf{R}|-1\}$ anchor point $\boldsymbol{\ell}\mathbf{B}_{\text{loc}}^{+}$ and intra-tile offset $\boxed{\boldsymbol{k}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^{+}$ are the unique choices such that these equivalent relationships hold:

$$k\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^{+} = \boldsymbol{\ell}\mathbf{B}_{\text{loc}}^{+} + \boxed{\boldsymbol{k}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^{+}, \tag{A14}$$

$$k\boldsymbol{k}^{\text{seed}} \quad\quad = \boldsymbol{\ell}\mathbf{R} \quad + \boxed{\boldsymbol{k}}. \tag{A15}$$

Suppose the value of function $\psi(k)$ is given by $\boxed{\boldsymbol{k}}$ of the above decomposition. By an argument essentially identical to that used above for $\phi(n)$, the function $\psi(k)$ is invertible. Given some allowed $\boxed{\boldsymbol{k}}$ and relationship $\psi(k) = \boxed{\boldsymbol{k}}$, integer k is established unambiguously.

Fifth: Assemble the Final Result

Given an allowed $\boxed{\boldsymbol{n}}$ there is exactly one integer n with $0 \le n < |\mathbf{R}|$ such that $\phi(n) = \boxed{\boldsymbol{n}}$, so the sum in (A1) can just as well be taken over those n. The same terms are summed. Indeed, we may as well let x_n refer to $x_{\boxed{\boldsymbol{n}}}$ using this same correspondence. Similarly, given an allowed $\boxed{\boldsymbol{k}}$ there is exactly one integer k with $0 \le k < |\mathbf{R}|$ such that $\psi(k) = \boxed{\boldsymbol{k}}$, so we can let X_k refer to $X_{\boxed{\boldsymbol{k}}}$ using that relationship. Making these substitutions in (A1) along with substitutions

$$\begin{aligned} \boxed{\boldsymbol{n}} &= \boldsymbol{n}^{\text{seed}} n - \mathbf{R}\boldsymbol{m}^n, \\ \boxed{\boldsymbol{k}} &= k\boldsymbol{k}^{\text{seed}} - \boldsymbol{\ell}^k \mathbf{R} \end{aligned}$$

from (A13) and (A15), where $\boldsymbol{m}$ and $\boldsymbol{\ell}$ have been replaced with $\boldsymbol{m}^n$ and $\boldsymbol{\ell}^k$ respectively as a reminder of their dependency on n and k,

$$X_k = \sum_{n=0}^{|\mathbf{R}|-1} x_n \cancelto{1}{\mathrm{e}^{-\mathrm{j}2\pi \boldsymbol{\ell}^k \mathbf{R} \boldsymbol{m}^n}} \cancelto{1}{\mathrm{e}^{\mathrm{j}2\pi k \boldsymbol{k}^{\text{seed}} \boldsymbol{m}^n}} \cancelto{1}{\mathrm{e}^{\mathrm{j}2\pi \boldsymbol{\ell}^k \boldsymbol{n}^{\text{seed}} n}} \mathrm{e}^{-\mathrm{j}2\pi k \boldsymbol{k}^{\text{seed}} \mathbf{R}^{-1} \boldsymbol{n}^{\text{seed}} n}. \tag{A16}$$

But by (A11),

$$\boldsymbol{k}^{\text{seed}} \mathbf{R}^{-1} \boldsymbol{n}^{\text{seed}} = M + \frac{1}{|\mathbf{R}|}$$

for some integer M, so the remaining complex exponential in (A16)

$$\mathrm{e}^{-\mathrm{j}2\pi k \boldsymbol{k}^{\text{seed}} \mathbf{R}^{-1} \boldsymbol{n}^{\text{seed}} n} = \cancelto{1}{\mathrm{e}^{-\mathrm{j}2\pi k M n}} \mathrm{e}^{-\mathrm{j}2\pi \frac{kn}{|\mathbf{R}|}},$$

so (A16) itself becomes

$$X_k = \sum_{n=0}^{|\mathbf{R}|-1} x_n \, \mathrm{e}^{-\mathrm{j}2\pi \frac{kn}{|\mathbf{R}|}},$$

a completely conventional DFT of $|\mathbf{R}|$ points in one dimension. Q.E.D.

A SIMPLE EXAMPLE

The key step was the third, finding a suitable $\boldsymbol{n}^{\text{seed}}$ to pair with the chosen $\boldsymbol{k}^{\text{seed}}$ so that $\boldsymbol{k}^{\text{seed}} \mathbf{R}_i \boldsymbol{n}^{\text{seed}} \bmod |\mathbf{R}| = 1$. Having established that such an $\boldsymbol{n}^{\text{seed}}$ exists, we can actually find it in practice with a simple brute-force search. Consider a MATLAB example with

```
>> R =[2,-1;...
>>     1, 3];
```

Now let us walk through the four steps of the proof but computing only what is essential.

1. *Express the elements of* $\mathbf{R}^{-1}$ *as rational numbers.*

```
>> adR=abs(det(R))
adR =
     7
>> Ri=round(inv(R)*adR)
Ri =
     3     1
    -1     2
>> R*Ri  % a check
ans =
     7     0
     0     7
```

2. *Tile beamspace, pick a seed point in beamspace, and analyze that point's coordinates.*
Begin either by choosing a beamspace seed point $\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^{+}$ that is not in $\mathbb{Z}^2\mathbf{B}_{\text{loc}}^{+}$ or, equivalently, choosing integer vector $\boldsymbol{k}^{\text{seed}}$ with $\boldsymbol{k}^{\text{seed}}\mathbf{R}_i \bmod |\mathbf{R}| \neq \begin{bmatrix}0 & 0\end{bmatrix}$. We can just guess and test the result.

```
>> kseed=[-1,-1];
>> mod(kseed*Ri,adR)
ans =
     5     4
```

3. *Construct a seed point in position space with a required property.*
Choose an $\boldsymbol{n}^{\text{seed}}$ to make $\boldsymbol{k}^{\text{seed}}\mathbf{R}_i\boldsymbol{n}^{\text{seed}} \bmod |\mathbf{R}| = 1$. Because this test is done modulo $|\mathbf{R}|$, integer multiples of $|\mathbf{R}|$ can be added to the elements of $\boldsymbol{n}^{\text{seed}}$ at will without changing the result of the test. This means it's enough to just test $\boldsymbol{n}^{\text{seed}} = \left[\begin{smallmatrix}n_1\\ n_2\end{smallmatrix}\right]$ for each $n_1, n_2 \in \{0, \ldots, |\mathbf{R}|-1\}$. (Or use any other set of integers no two of which differ by a multiple of $|\mathbf{R}|$.)

```
>> range=(0:(adR-1));
>> n1=range;
>> n2=range;
>> [i,j]=find(mod(repmat(kseed*Ri(:,1)*n1.',1,adR)+...
>>                repmat(kseed*Ri(:,2)*n2,adR,1)...
>>                ,adR)==1);
>> nseedInit=[n1(i);n2(j)]
nseedInit =
     3     5     0     2     4     6     1
     0     1     2     3     4     5     6
```

Of course $\boldsymbol{n}^{\text{seed}}$ here really represents position-space point $\mathbf{B}_{\text{loc}}\boldsymbol{n}^{\text{seed}}$, and only this point's intra-tile offset actually matters. There are $|\mathbf{R}|$ points in a tile, but this calculation tests $|\mathbf{R}|^2$ choices of $\boldsymbol{n}^{\text{seed}}$, so it should be no great surprise that this test returns $|\mathbf{R}|$ suitable choices of $\boldsymbol{n}^{\text{seed}}$.

If each column output above is taken to be a choice of $\boldsymbol{n}^{\text{seedInit}}$, it is easily verified that each of the associated points $\mathbf{B}_{\text{loc}}\boldsymbol{n}^{\text{seedInit}}$ has the same intra-tile offset.

```
>> (R*mod(Ri*nseedInit,adR))/adR
ans =
     0     0     0     0     0     0     0
     2     2     2     2     2     2     2
```

If we take $\boldsymbol{n}^{\text{seed}}$ to be one of the columns of `nseedInit` plus a vector $|\mathbf{R}|\left[\begin{smallmatrix}m_1\\ m_2\end{smallmatrix}\right]$, then choosing the column and m_1 and m_2 chooses the tile. The results do not depend on which tile is chosen, so this is only a matter of plotting convenience (as in Fig. A1 below). The chosen $\boldsymbol{n}^{\text{seed}}$ should be tested to be sure.

```
>> nseed=nseedInit(:,2)+adR*[-1;0]
nseed =
    -2
     1
```

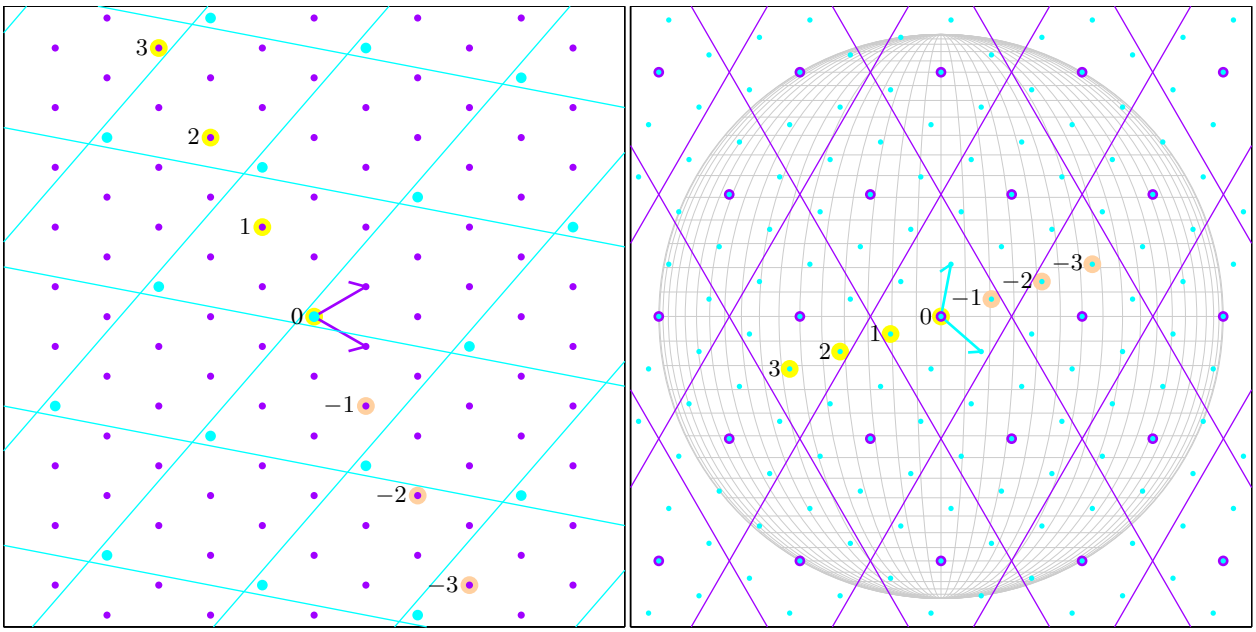


Fig. A1 — Graphical approach to ordering intra-tile offsets, which correspond to FFT input (left) and output (right) bins, so that a 2D DFT with $|\mathbf{R}|$ prime becomes an ordinary 1D DFT. Highlighting flags whether numbers indicate 1D bin indexes or 1D bin indexes less $|\mathbf{R}|$. Here $|\mathbf{R}| = 7$, corresponding to seven points per tile.

```
>> mod(kseed*Ri*nseed,adR)
ans =
     1
```

4. *Construct invertible functions from the modulo-$|\mathbf{R}|$ integers to the vectors in the prototype tiles.*

According to (A12) and (A14) we can obtain the ordered $\boxed{\boldsymbol{n}}$ and $\boxed{\boldsymbol{k}}$ in our $|\mathbf{R}| = 7$ example from the intra-tile offsets of $\mathbf{B}_{\text{loc}}\boldsymbol{n}^{\text{seed}}n$ and $k\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^{+}$ with n and k each taking values in turn of $0, 1, 2, 3, 4, 5, 6$. If we assume basis-vector tiling has been used in both position space and beamspace, we can easily compute the index vectors for the intra-tile offsets.

```
>> nBin=(R*mod(Ri*nseed*range,adR))/adR
nBin =
     0     0     1     1     0     0     1
     0     2     1     3     1     3     2
>> kBin=(mod(range.'*kseed*Ri,adR))*R/adR
kBin =
     0     0
     2     1
     1     0
     1     2
     2     0
     2     2
     1     1
```

Did it work? We can just test all the $\boldsymbol{k}^{\text{bin}}\mathbf{R}_i\boldsymbol{n}^{\text{bin}}$ products and see if they are the same, as hoped, as the kn products.

```
>> kbinRinbin=mod(kBin*Ri*nBin,adR)
```

```
kbinRinbin =
     0     0     0     0     0     0     0
     0     1     2     3     4     5     6
     0     2     4     6     1     3     5
     0     3     6     2     5     1     4
     0     4     1     5     2     6     3
     0     5     3     1     6     4     2
     0     6     5     4     3     2     1
>> all(all(kbinRinbin==mod(range.'*range,adR)))
ans =
     1
```

One other tiling scheme that is nearly as simple uses basis-vector tiles offset by a vector. For example, on the right in Fig. A1 the beamspace tiles are basis-vector tiles translated by $-\frac{1}{2}\mathbf{B}_{\text{loc}}^{+}$. Computing index vectors for the intra-tile offsets is simple, because no points lie on tile boundaries:

```
>> offset=ones(length(range),2)*.5;
>> kBin=round((mod(range.'*kseed/R+offset,1)-offset)*R)
kBin =
     0     0
    -1    -1
     1     0
     0    -1
     0     1
    -1     0
     1     1
```

A geometric view of the correspondence the above calculations establish between scalar indices n and k and the position-space and beamspace bins they are discovered to represent is especially illuminating. One could plot left sides $n\mathbf{B}_{\text{loc}}\boldsymbol{n}^{\text{seed}}$ and $k\boldsymbol{k}^{\text{seed}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^{+}$ of (A12) and (A14) over position-space and beamspace tile diagrams and to obtain or verify intra-tile offsets $\mathbf{B}_{\text{loc}}\boxed{\boldsymbol{n}}$ and $\boxed{\boldsymbol{k}}\mathbf{R}^{-1}\mathbf{B}_{\text{loc}}^{+}$ and associated index vectors $\boxed{\boldsymbol{n}}$ and $\boxed{\boldsymbol{k}}$ graphically. This is almost what has been done in Fig. A1. "Almost" is because instead of n and k taking values of $0, 1, 2, 3, 4, 5, 6$, in the figure they have been given values $0, 1, 2, 3, -3, -2, -1$ for plotting convenience. This is acceptable because it is harmless to change n and k by multiples of $|\mathbf{R}|$. The values of $\boxed{\boldsymbol{n}}$ and $\boxed{\boldsymbol{k}}$ are not affected. To see this we use relationship $\mathbf{I} = \mathbf{R}\mathbf{R}^{-1} = \mathbf{R}\mathbf{R}_i\frac{1}{|\mathbf{R}|}$ or its equivalent $\mathbf{R}\mathbf{R}_i = |\mathbf{R}|\mathbf{I}$ to write

$$\begin{aligned}\boldsymbol{n}^{\text{seed}}(n + m|\mathbf{R}|) &= \boldsymbol{n}^{\text{seed}}n + |\mathbf{R}|\mathbf{I}\boldsymbol{n}^{\text{seed}}m \\ &= \boldsymbol{n}^{\text{seed}}n + \mathbf{R}\mathbf{R}_i\boldsymbol{n}^{\text{seed}}m \\ &= \mathbf{R}(\boldsymbol{m} + \mathbf{R}_i\boldsymbol{n}^{\text{seed}}m) + \boxed{\boldsymbol{n}},\end{aligned} \tag{A17}$$

where (A13) has been used in the last step. Similarly, add a multiple of $|\mathbf{R}|$ to k on the left side of (A15) and use $\mathbf{R}_i\mathbf{R} = \mathbf{I}|\mathbf{R}|$ and (A15) to obtain

$$\begin{aligned}\boldsymbol{k}^{\text{seed}}(j|\mathbf{R}| + k) &= j\boldsymbol{k}^{\text{seed}}\mathbf{I}|\mathbf{R}| + k\boldsymbol{k}^{\text{seed}} \\ &= j\boldsymbol{k}^{\text{seed}}\mathbf{R}_i\mathbf{R} + k\boldsymbol{k}^{\text{seed}} \\ &= (j\boldsymbol{k}^{\text{seed}}\mathbf{R}_i + \boldsymbol{\ell})\mathbf{R} + \boxed{\boldsymbol{k}}.\end{aligned} \tag{A18}$$