Diagonal Interleaving Poetry: A New Generative Permutational Poetic Form Inspired By Error-Correcting Coding Theory

Susan Gerofsky

Department of Curriculum and Pedagogy, University of British Columbia, Vancouver, Canada;
susan.gerofsky@ubc.ca

Abstract

Diagonal interleaving poetry is a new poetic form that generates poems from an initial matrix of words and phrases, using a periodic permutation inspired by interleaving (a technique from error-correcting coding theory). The author introduces this new form in the context of other permutational poetry and wordplay and describes its formal connections with interleaving techniques in coding theory. Poetic and combinatorial constraints and affordances are discussed. Throughout the paper, several n×m matrices of words and phrases are developed as interleaving poems, offering examples of ways that the form can be used to generate interesting poetry.

Introduction: Permutational Poetry

Permutation patterns offer an interesting creative constraint with the potential to generate new poetic forms and poems. Bridges poets Marian Christie [1] and JoAnne Growney [5] have explored some of these in their mathematical poetry blogs, and I have written about them in an earlier Bridges paper [2]. Kowalska [6] also offers a look at the history of permutational poems in the preamble to her paper on this style of poetry on the South African literary scene.

Permutations can take place on many levels in the creation of poetry, including permutation of letters in anagrammatic poems; permutations of sounds to create puns and Spoonerisms; permutations of lines and verses of poems, in forms like villanelles and triolets; and permutations of words and phrases, as in the PH4 poems described in an earlier Bridges workshop paper [4]. Many of the permutational poems Kowalska cites are examples of 'concrete poetry', where the visual typography and organization of words and letters on the page are of primary importance, but in other permutational poetic forms, sound and metre, imagery and metaphor and other poetic features may take precedence.

Some permutational poetic forms are relatively familar and well-known. These include sestinas, villanelles and pantoums, which formalize a patterning of refrain lines, rhymes and words at the ends of lines. There are poems that support readers' multiple choices to create many possible final versions of the poem like Queneau's *Cent mille milliards de poémes* ('A hundred thousand billion poems'), and Mathew's *Multiple choice*, both part of the 20th C French Oulipo movement [8], and Bridges poet Mike Naylor's similarly-structured *Decision Tree* [9]. My own previously-developed permutational poetic forms include combinatoric patterns based on multi-strand braiding (*Seven Strands of Alphabetical Braided Crows*) [3], bellringing (PH4 poems), and formal poems that incorporate anagrams within some of the alliterative and caesura structures of Anglo Saxon poetry [2].

Mathematical permutation patterns for generating poetry may be partially or wholly deterministic; but where there is a human poet involved in the poetic process, rather than simply an automated algorithmic procedure [7], the process always involves something greater than 'mere generation' of deterministic forms. To write 'good' poetry — poetry that is memorable, playful, meaningful, that creates striking imagery and sound patterns — requires choices that are made through human experience of language and life. As with any constrained poetic form, from sonnets to haikus and 'fibs', the interplay of the formal rules and the poet's judgement is what makes poems that speak to readers. Experimentation with a poetic form is a kind of mathematical-literary inquiry that yields insights into what does and doesn't work, based on the

constraints and affordances of the poetic form and the forms of the language. I will be offering my own advice to poets on some aspects of a 'good' interleaving poem in the concluding part of this paper.

Interleaving: A Permutational Process Inspired By A Process Used With Error-Correcting Codes

The permutational form used in interleaving poems is motivated by a process used in the mathematical areas of error-correcting codes and coding theory. To be clear, this process is not error-correcting in itself, and I use it for the aesthetics of the poems it produces, not as a method for error-correction! I came to this shift-based permutation through coding theory, but it could ostensibly be derived in other contexts as well.

Error detecting and correcting codes are different from secret codes or cryptography; they are not used to hide a message or keep it private, but to reveal and repair it, even when the original message is damaged by noise, static, human copying error or degradation of the medium that contains the message. Here are some examples of the uses of error correcting codes in the communication of data:

- Most tracking numbers contain a check digit at the end that is the sum of all the previous digits, modulo 7 (or some other prime number). If the sum does not match the check digit, that indicates that there is an error in the entry or transmission of the tracking number—or of the check digit. In any case, a discrepancy would indicate a need to double-check the tracking number.
- A hard medium like a CD may have a physical scratch on its surface. Error-correcting codes make it possible to reconstruct the missing data damaged by the scratch and to interpolate digital data that will repair the data loss.
- A Mars lander transmits photos and videos in the form of streams of data to be received and reconstructed on Earth. Interference from radiation in space between Mars and Earth may create noise that obliterates bursts of data. Error-correcting codes make it possible to reconstruct and interpolate missing data and produce images faithful to the far-away source.

Error-correcting coding theory uses techniques related to abstract algebra (rings and fields). However there is a simple permutational technique called interleaving that is used in conjunction with error-correcting techniques to minimize and distribute burst errors, so that algebraic error-correction techniques can deal with smaller, random errors rather than large amounts of missing code clustered in one section of the message.

Interleaving rearranges the bits of information before transmitting a message, and then puts them back in the original order after the message is received. The aim is to turn burst errors that affect a large data cluster into smaller, randomly placed errors that are easier to correct through error-correcting techniques. (Imagine transmitting a message by Morse Code over a staticky network: interleaving would make it possible to correct a random scattering of individual missing letters rather than whole missing words.)

In error-correcting codes, data (or 'words' of code) are arrayed in the rows of a rectangular matrix, and then transmitted column by column, so that the first letter of each the words would be transmitted sequentially, then the second letter of each of the words, and so on. (See Figure 1 for an example.) By interleaving the words of the message in this way, a burst error that affected up to three letters in the sequence would only affect one letter in each of the three words.

Table 1: *The original message (3 five-letter words), and the resulting interleaved message, created by rearranging the letters column by column.*

A	N	G	L	E
B	R	O	W	S
C	H	U	M	P

--> A B C N R H G O U L W M E S P

This simple permutation allows for burst errors to be spread over random single letters, rather than wiping out large chunks of whole words to a degree where it might be impossible to reconstitute them through other error-correcting techniques. It is also easy to reconstitute the original words by arranging the letters back into the vertical columns of the 5 x 3 array, and reading the words horizontally along the rows.

Developing a Diagonalized Interleaving Poetic Form

In designing an interleaving permutational poetic form, I wanted to use words and phrases, rather than single letters (or numbers), to create my 'message', and then to permute those words and phrases using a process of interleaving to alter the message and develop extensions and variations on its meaning, while holding the original word choices invariant.

However, if I had used exactly the technique of interleaving described above, where words are arrayed in horizontal rows to form lines of the poem, and then taken in vertical columns to create the interleaved version, there would only be two versions of the poem generated: the original and the columnar version. Interleaving the columnar version would revert the word order back to its original form, and further iterations would just oscillate between these two versions. (See Table 2.)

Table 2: *(a) First lines of Poe's The Raven in a 4 x 4 array (original)*

Once	upon	a midnight	dreary
While	I pondered	weak	and weary
Over	many	a quaint	and curious
Volume	of	forgotten	lore

(b) Interleaved version using vertical columnar rearrangement

Once	while	over	volume
Upon	I pondered	many	of
A midnight	weak	a quaint	forgotten
Dreary	and weary	and curious	lore

Note that repeating the vertical columnar interleaving on (b) would render (a) again.

This two-verse vertically interleaved permutation of the initial lines of Poe's *The Raven* is not without its charms, but it did not suit my poetic purposes. I wanted to create a form that could take the original words and phrases of the first stanza to a larger number of permutations, to explore further interesting rearrangements, sounds, associations, images and meanings. However, I did want the permutations to be constrained, and did not wish to create an exhaustive catalogue of all possible permutations, as that might be too dreary, and leave readers weary...

There is another approach to interleaving that reorganizes the data array along diagonal lines rather than vertical columns. This diagonal approach can create redundancy in the transmission of error-correcting codes by rearranging the code words (or letters or numbers) in several different ways, so that a short message can be retransmitted several times in more than one 'diagonalization'. The redundancy makes it easy to reconstruct the original message even if there are several burst errors, as the message is interleaved in more than one iteration. The diagonal interleaving begins with the first word (or letter or number) in the first line (the upper left box in the array) and proceeds down the main diagonal; then jumps to the first word

in the second line and proceeds down the diagonal from there, looping to the top of the next column as needed; and so on, until all the words have been used. (See Table 3).

Table 3: *Diagonal interleaving of a 4 x 4 array*

(a) Original order

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

(b) First diagonal interleaving

A	F	K	P
E	J	O	D
I	N	C	H
M	B	G	L

Note that this process can be reiterated n times for an n×n square array to create n distinct diagonal interleavings. The (n+1)th interleaving will bring the array back to its original state.

Note that a technique of multiple interleavings allows for error correction simply because sending the same message multiple times makes a communication more robust (as it is unlikely for the same errors to occur in multiple transmissions). From a poetic viewpoint, employing multiple interleavings can be used to make the multifaceted meanings of individual words more robust, as nuances are revealed by placing each word into juxtaposition with different words in each stanza: a tangential but interesting connection to error correction. The effect of diagonal interleaving on an actual poem can be seen below in Table 4, in contrast with vertical columnar interleaving of the same lines shown above in Table 2:

Table 4: *First lines of Poe's The Raven in a diagonalized interleaving form:*

(a) Original 4×4 array:

Once	upon	a midnight	dreary
While	I pondered	weak	and weary
Over	many	a quaint	and curious
Volume	of	forgotten	lore

(b) Diagonalized interleaved version 1:

Once	I pondered	a quaint	lore
While	many	forgotten	dreary
Over	of	a midnight	and weary
Volume	upon	weak	and curious

(c) Diagonalized interleaved version 2:

Once	many	a midnight	and curious
While	of	weak	lore
Over	upon	a quaint	dreary
Volume	I pondered	forgotten	and weary

(d) Diagonalized interleaved version 3:

Once	of	a quaint	and weary
While	upon	forgotten	and curious
Over	I pondered	a midnight	lore
Volume	many	weak	dreary

The fourth diagonalization would reproduce the original 4×4 array/ poem.

The diagonal interleaving keeps the words in their original columns, but leaves the first column in its original order, permutes the second column by one position each time, the third column by two positions each time, etc.

Diagonal interleaving works well for a poem based on an n×n matrix (for example, in my poem *Diagonal Eyes Enter Leaving* below, a 6×6 matrix of six words or phrases and six lines), although any n×m matrix (with n, m>2) could make a potentially interesting poem. However, if n>m, there will be more than one column that will remain in its original order all the way through, with potentially more repetition of words, and less variety in the poem; the first column always remains in its original order, as well as any other column whose number is 1, modulo the number of rows. Table 5 below gives an illustrative example of an original short poem in the form of an n×m matrix, n>m; n, m >2 (in this case, a 5×3 matrix) and its two diagonalized interleavings. Note that column 1 *and* column 4 remain in their original order throughout, as (column number) $4 = 1 \ mod \ 3$ (the number of rows):

Table 5: Diagonal interleaving of an original rectangular (non-square) array poem, **Spring Moon**:

(a) Original 5×3 array:

Wan, pale, and	listless	the moon	wanders	downward
To	the edges	of sky	where waters	shine
Trees	catch it	bright	turbulent	adrift

(b) Diagonalized interleaved version 1

Wan, pale, and	the edges	bright	wanders	shine
To	catch it	the moon	where waters	adrift
Trees	listless	of sky	turbulent	downward

(c) Diagonalized interleaved version 2:

Wan, pale, and	catch it	of sky	wanders	adrift
To	listless	bright	where waters	downward

Trees	the edges	the moon	turbulent	shine

Here is the poem in non-table form:

Spring Moon

Wan, pale, and listless, the moon wanders downward
To the edges of sky where waters shine.
Trees catch it: bright, turbulent, adrift.

Wan, pale, and the edges bright, wanders shine
To catch it: the moon, where waters adrift
Tree, listless, of sky, turbulent, downward,

Wan, pale and catch it of sky, wanders adrift
To listless bright, where waters downward
Trees, the edges, the moon, turbulent shine.

An Example of an Original Diagonalized Interleaving Poem on a Square Array

Now, with the technical description of this periodic permutational poetic form, the proof of its interest is in the poems that it can generate, in dialogue with a living human poet. Here is my first diagonalized, interleaving poem on an n×n array, followed by some notes on its construction:

Diagonal Eyes Enter Leaving

My eyes scan cross-cutting fragments of diagonals;
The codes seem clearly to be error-correcting.
What images fly cornerwise across my vision?
Did birds or bats sneak subtly through the screen?
Are there ghosts lurking in dimly-lit edges of this room?
We mortals intuit, through tears, the persistence of dreams.

My codes fly subtly, edges of dreams;
The images sneak in, dimly-lit, the persistence of diagonals.
What birds or bats, lurking through tears, fragments of error-correcting,
Did ghosts intuit cross-cutting to be my vision?
Are there mortals scan clearly across the screen?
We eyes seem cornerwise through this room.

My images lurking, cross-cutting, across this room;
The birds or bats intuit clearly through dreams.
What ghosts scan cornerwise edges of diagonals?

Did mortals seem, subtly, the persistence of error-correcting?
Are there eyes fly in dimly-lit fragments of my vision?
We codes sneak through tears to be the screen.

My birds or bats scan, subtly, fragments of the screen;
The ghosts seem in dimly-lit to be this room.
What mortals fly through tears across dreams?
Did eyes sneak cross-cutting through diagonals?
Are there codes lurking clearly, edges of error-correcting?
We images intuit cornerwise the persistence of my vision.

My ghosts fly, cross-cutting edges of my vision;
The mortals sneak, clearly the persistence of the screen.
What eyes lurking cornerwise, fragments of this room,
Did codes intuit subtly to be dreams?
Are there images scan in dimly-lit across diagonals?
We birds or bats seem, through tears, through error-correcting.

My mortals lurking subtly across error-correcting;
The eyes intuit in dimly-lit through my vision.
What codes scan, through tears, edges of the screen?
Did images seem cross-cutting the persistence of this room?
Are there birds or bats fly, clearly fragments of dreams?
We ghosts sneak cornerwise to be diagonals.

About the Structure of *Diagonal Eyes Enter Leaving*

Table 6: *(a) Original 6×6 array*

My	eyes	scan	cross-cutting	fragments of	diagonals
The	codes	seem	clearly	to be	error-correcting
What	images	fly	cornerwise	across	my vision
Did	birds or bats	sneak	subtly	through	the screen
Are there	ghosts	lurking	in dimly-lit	edges of	this room
We	mortals	intuit	through tears	the persistence of	dreams

(b) Diagonalized interleaved version 1 (of 5)

My	codes	fly	subtly	edges of	dreams
The	images	sneak	in dimly-lit	the persistence of	diagonals
What	birds or bats	lurking	through tears	fragments of	error-correcting
Did	ghosts	intuit	cross-cutting	to be	my vision

Are there	mortals	scan	clearly	across	the screen
We	eyes	seem	cornerwise	through	this room

The second stanza is obtained reading this table 'on the diagonal'. That is, the first line of the second stanza is the diagonal of the first stanza, and so on. The third stanza is obtained from the second by the same process of diagonalized interleaving. The process is continued for as long as each new stanza is different from all the stanzas that came before it. Note that the diagonalization of stanza 6 yields stanza 1.

The title and theme of this poem were drawn from a shameless pun on 'diagonalize interleaving'. The thought of 'diagonal eyes' reminded me of the eerie and disconcerting floaters that crossed my peripheral vision cornerwise for several weeks after experiencing the detachment of the vitreous gel from the retina in my eye a few years ago. Although it was winter, I would get the impression that a fly had somehow entered the room—through the screen?—and was flying by on the edges of my vision. The choice of words like 'screen' and 'tears' with their multiple meanings, was deliberate; ambiguous words offer depth and interpretive possibilities in the permuted versions of the original stanza.

For mathematical poets ready to try writing a diagonalized interleaving poem, I would advise against writing each line of the original stanza with an identical, rigid grammatical structure; although that would guarantee that the lines would reliably permute, it may sound stilted. The choice of words and word/phrase breaks is crucial in setting the tone of the interleaved verses, and might take several tries as you place these breaks where you like them best. It's worthwhile to set and revise your initial word choices and array-setting in dialogue with the interleaving process until you are satisfied with the results. For me, this interactive process with the permutational form was both illuminating and surprising, and shed light on unexpected meanings of my original stanza through this theme-and-variation approach.

Acknowledgements

Thanks always to Sarah Glaz for her enthusiasm, encouragement and support for all the mathematical poets.

References

[1] M. Christie (April 2021). "Mathematical Forms in Poetry: Permutations." marianchristiepoetry.net/mathematical-forms-in-poetry-4-permutations/

[2] S. Gerofsky. "Two New Combinatoric Poetry Forms: Braided Bellringing PH4 Poems & Anagrammatic, Anglo Saxon-inspired Poems." *Bridges Conference Proceedings,* online, Aug 1-5, 2020, pp. 273-280. https://archive.bridgesmathart.org/2020/bridges2020-273.html

[3] S. Gerofsky. "Seven Strands of Alphabetical Braided Crows." *Bridges Short Film Festival*, online, Aug 1-3, 2021. http://gallery.bridgesmathart.org/exhibitions/2021-bridges-conference-short-film-festival/susan-gerofsky

[4] S. Gerofsky, E. Knoll, T. Taylor and A. Campbell-Cousins. "Experiencing Group Structure: Observing, Creating and Performing the Plain Hunt on 4 Via Music, Poetry, Visual and Culinary Arts." *Bridges Conference Proceedings*, Stockholm, 25-29 July 2018, pp. 659-666. https://archive.bridgesmathart.org/2018/bridges2018-659.html

[5] J. Growney (May 2017). "Intersections: Poetry with Mathematics: Poems with Permutations." poetrywithmathematics.blogspot.com/2011/05/poems-with-permutations.html

[6] E. Kowalska. "All Down the Line: Permutation Poetry in Three South African Journals, 1965–1975." *English in Africa*, *44*, 1, April 2017, pp. 55-71.

[7] C. Lamb, D.G. Brown, and C.L.A. Clarke. "A Taxonomy of Generative Poetry Techniques." *Bridges Conference Proceedings* Jyväskylä, Finland, Aug. 9–13, 2016, pp. 195–202. https://archive.bridgesmathart.org/2016/bridges2016-195.html

[8] H. Mathews and A. Brotchie (Eds.) *Oulipo Compendium*. Atlas Press, 1998.

[9] M. Naylor. "Decision Tree." http://mike-naylor.blogspot.com/2011/09/decision-tree.html

[10] S. A.Vanstone and P. C. van Oorschot. *An Introduction to Error Correcting Codes with Applications.* Springer, 1989.