GMRES and Polynomial Algebra Equivalence *

gmresalgebra@optoutofdatabases.33mail.com

August 22, 2020

Abstract

This paper establishes a theoretical link between GMRES and the much simpler problem of polynomial evaluation. In doing so, we define some algebraic structures that synesthize some of the most important elements of the GMRES algorithm. These elements can be used to abstract and provide new ways for thinking about the GMRES algorithm. We use this homomorphism to show the connection between sequential GMRES and Horner's Rule, s-step GMRES and Dorn's rule, and predict future possible GMRES-like algorithms

Keywords. GMRES, s-step GMRES, Krylov space, polynomial evaluation, Horner's rule, Dorn's rule

AMSsubjectclassifications. 65F10, 65F25, 12Y05, 15A06, 16S50, 20G40, 20H30

1 Introduction

Given a matrix A, the GMRES algorithm sets out to find a solution to a linear system $Ax = b$ [11, 13]. The idea behind which is to use repeated multiplications by A in order to iteratively create a search space to help approximate the solution. This iteration is a well-known subroutine of GMRES known as Arnoldi's method, and it includes two key operations in a loop: matrix-vector multiplication by A, and orthonormalization. The occurrence of these two subroutines differ in appearance and implementation, but in all GMRES-like algorithms, we see repeated use of matrix-vector multiplications, and orthonormalization.

Traditionally, GMRES operated as a sequential algorithm; however, recently, there has been interest in applying GMRES in parallel to improve its performance. This has resulted in a number of methods, such as s-step GMRES. Current improvements look at variations of deflation, augmentation, better preconditioners, different polynomial preconditioner, or trying to calculate one iteration's orthogonalization concurrently with the next step's multiplication routine[1, 3, 7, 8, 12, 14]. All such s-step GMRES can be interpreted as a generalization of traditional GMRES[11]. The growing number and complexity of these GMRES algorithms and their implementations of the nontrivial operations of matrix-vector multiplication and orthnormalization make GMRES very difficult to analyze and optimize.

Therefore, there is a huge benefit if we can make an analogy between GMRES and something much more simple like polynomial evaluation. In any polynomial evaluation method, like Horner's rule, Dorn's rule, etc., we trivially have two basic subroutines: multiplication and addition[6]. We aim to show that the matrix-vector multiplications and orthonormalizations in GMRES-like algorithms may be thought in some sense (as to be defined more precisely below) as ordinary multiplication and addition in any standard polynomial evaluation algorithm.

In the following section, we recall some basic properties of GMRES and some common polynomial evaluation algorithms in order to provide intuition for the reader. In section three we use this intuition in order to properly define a ring structure that we argue will properly model the behavior of GMRES, and we use this ring structure to make well-defined a ring homomorphism between GMRES and the polynomial

*This work was supported in part by the EXA2CT Project.

ring $Z_2[x]$. Armed with this homomorphism, in section 4 we show some examples of how this relates and explains currently existing GMRES-like and polynomial evaluation algorithms. Finally, we conclude with some thoughts for future work.

2 Review of GMRES and Polynomial evaluation algorithms

We start by providing a simplified overview of the GMRES algorithm, and then focus solely on the Arnoldi loop by simplifying the following into a skeleton and carving out its core components. As stated before, the aim of GMRES is to solve $Ax = b$. In what follows, r_0 is the residual, from which repeated multiplications serve to form the search space V_m, which is done by constructing an approximate solution in the affine space $x_0 + V_m \cdot y_m$. The key thing to notice is the inner loop in which a matrix-vector multiplication (matvec) is interpolated with a orthonormalization procedure (such as Gram-Schmidt, but one should keep in mind that in practice the choice of an orthonormalization procedure can influence the computation heavily[5]). We show this algorithm explicitly and in full below:

Definition 0.1 (GMRES). $r_0 = b - Ax_0, \beta := ||r_0||_2, v_1 = \frac{r_0}{\beta}$

For $j = 1, 2, ..., m$ } *Arnoldi*
- $w_j := Av_j$ } *Matvec*
- *For* $i = 1, ..., j$ } *Gram-Schmidt*
  - $h_{ij} := (w_j, v_i); w_j := w_j - h_{ij}v_i$
- *End*
- $h_{j+1,j} = ||w_j||_2$
- *If* $h_{j+1,j} = 0$ } *Fortunate Breakdown*
  - $m := j$
  - *break*
- $v_{j+1} = \frac{w_j}{h_{j+1,j}}$

End

$H_m := [h_{ij}]$

Find $y_m = min||\beta e_1 - H_m y||_2$ *(e.g., via Givens orthonormalization [2])*

$x_m := x_0 + V_m y_m$

In particular, in sequential GMRES, the Arnoldi step can be thought of as the following (for completeness, we included the breakdown case above in definition 0.1, but in the rest of this paper, we make the proper assumptions to avoid this case to simplify our discussion):

Definition 0.2 (Arnoldi).
- *Loop j=1,2,...,m*
- $w_j := Av_j$
- *Apply Gram-Schmidt to* $\{w_j, v_{j-1}, \cdots, v_1\}$ *to get* v_{j+1}
- *End*

However, as we have stated, we aim to simplify the problem into one of optimizing polynomial evaluation. As such, in order to compare with the above, we recall Horner's rule as the sequential algorithm for evaluating the given polynomial

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \tag{1}$$

which redistributes the coefficients and variables in to the following expression:

$$(((a_n) \cdot x + a_{n-1}) \cdot x + \cdots) \tag{2}$$

This is efficient because the standard way of evaluating the polynomial as in equation 1 requires adding a_0 then adding the multiplication of a_1 with x then adding the multiplication of a_2 with x^2, and so on, which requires roughly $O(\Sigma_{i=1}^{n} i) = O(n^2)$ operations, whereas Horner's rule only requires an addition and a multiplication for each coefficient, which requires roughly $O(n)$ operations.[6]

We rewrite the skeleton code of the Arnoldi step algorithm of definition 0.2 used in sequential GMRES and we rewrite Horner's rule for evaluating the polynomial that we have just presented in order to better compare the two:

$$(j = n-1 : 0)$$

Arnoldi's Iteration	Horner's Rule
$V_n = \{v_0\}$	$b_n = a_n$
$V_j = \text{orthogonalize}\{V_{j+1}, A \cdot \text{last vector of } V_n\}$	$b_j = a_j + x \cdot b_{j+1}$

From an algebraic standpoint, the only difference between the two algorithms is that Horner's rule relies on a_j whereas Arnoldi relies on the previous step V_{j+1} (keeping in mind that the presentation here counts down to 1). To make the algebraic structure even further apparent, we consider a slightly modified Horner's rule that relies on b_{j+1}, and instead of acting on a general polynomial in $\Re[x]$, we abstract and consider its action on $Z_2[x]$. With these modifications we obtain the following:

$$(j = n-1 : 0)$$

Arnoldi's Iteration	Abstracted, Modified Horner's Rule
$V_n = \{v_0\}$	$b_n = 1$
$V_j = \text{orthogonalize}\{V_{j+1}, A \cdot \text{last vector of } V_n\}$	$b_j = b_{j+1} + x \cdot b_{j+1}$

This mirroring presentation is the central rationale for our intuition between the connection of GMRES and polynomial evaluation and forms the base for our theoretical investigations. However, this similarity is potentially a fluke unless it is replicated in other GMRES and polynomial evaluation methods. Therefore, we define some of these other forms. In particular, s-step GMRES and Dorn's rule for GMRES and polynomial evaluation respectively.

The following is a sketch of the Arnoldi step in a s-step GMRES implementation, and can be compared with the sketch of an Arnoldi step in GMRES implementation above:

Definition 0.3 (ssteparnoldi)**.**

- $w_0 := r_0 = b - Ax_0, \beta := ||r_0||_2, v_1 = \frac{r_0}{\beta}$
- *Loop*
- *For j=1,2,...,s*
- $w_j := Aw_{j-1}$
- *End*
- *Orthogonalize* $\{w_s, w_{m-1}, \cdots, w_1\}$ *to get V*

Likewise, the following can be compared to Horner's rule above, and is the skeleton of Dorn's rule (where every power mod s is evaluated independently):

$$\begin{aligned} b_j &= a_j, j = n, \cdots, n-k-1 \\ b_j &= a_j + x^s \cdot b_{j+1} \end{aligned} \tag{3}$$

For example, if for $s = 2$ one wanted to evaluate $1 + x + x^2 + x^3$ they would instead evaluate it as $x(1 + x^2) + (1 + x^2)$, where each parenthetical component could perform an independent Horner's rule.[6]

If we place the two side-by-side, we obtain a structure as follows.

$$(j = n-1 : 0)$$

Unabstracted s-step	Abstracted, Modified Dorn's Rule
$V_n = \{v_0\}$	$b_j = 1, j = n, \cdots, n-k+1$
$V_j = \text{orthogonalize } \{v_{j+1}, Av_{j+1}, \cdots, A^s v_{j+1}\}$	$b_j = b_{j+1} + x^s \cdot b_{j+1}$

If the x^s multiplication can be identified with the s simultaneous matvec operations, then in both of the abstracted and modified Horner's and Dorn's rules we see that s components are computed independently before and addition operation is called upon. In both methods we see a strikingly similar structure. In order to clarify why this is the case any further, we must now turn to analyzing the underlying algebraic structure of GMRES.

3 GMRES and Polynomial Chain Homomorphisms

We wish to create a homomorphism in order to formalize the algebraic and structural similarities between GMRES and polynomial computations. As such, we simplify the polynomial ring to be considered to be $Z_2[x]$, and we will first need to define a ring structure whose operations mirror the important elements *of* the orthonormalization and matrix-vector product operations in the Arnoldi loop of any GMRES-derivative algorithm. Our aim will be to make what results in the orthonormalization procedure in the context of Arnoldi to be viewed as an addition, and the matrix-vector product procedure viewed algebraically as multiplication by an indeterminate. In order to do so, we will thus first need to ensure that a simplified model of the orthonormalization procedure can form a commutative group, which in turn will require some definitions whose aim will not be fully apparent until the proof of the well-definedness of the group is achieved. Therefore, we set out first to make these definitions, and we must delay the full discussion as to our reasons for these choices until the proof of well-definedness. Then we may analyze each choice's influence on the underlying algebra before generating the ring structure we need in order to rigorously define the polynomial homomorphism between GMRES and polynomial computations.

To this end, we assume that we have a nonsingular matrix A of size n and a vector r whose minimal polynomial w.r.t. A has degree n. Let Q be a fixed and given orthonormalization of the Krylov basis $K := \{r, Ar, A^2r, \ldots, A^{n-1}r\}$. We make the following definitions and notations as a prelude to our ring structure.

Definition 0.4. *a* $\Re^{n\times n} \supseteq \mathbb{W} := \{(v_1, v_2, \ldots, v_n) = V \in \Re^n | \forall v_i, v_i$ *is a column of* Q *(with no repetitions) or* $= 0\}$*, we identify elements of* $\mathbb{W}$ *as equivalent up to permutation.*

b By $[A, B]$ *we mean the column concatenation of two matrices or lists of vectors* A *and* B*.*

c For $A, B \in \mathbb{W}$*, by* $dup_2([A, B]) = U \in \Re^{n\times n}$ *we let* U *be the list parallel nonzero vectors in* $[A, B]$ *that occur an odd number of times, and let the rest of the vectors be* 0*.*

d Our group addition operation will be $A \oplus_Q B := dup_2([A_Q, B_Q])$*.*

We use Q in our definition of $\oplus$ here to emphasize that the space expresses its contents in terms of the vectors of the orthonormal matrix Q, and it is this expression of our elements that we will think of later as a form of orthonormalization proper. In other words, this operation is a simplification meant to express the result of the orthonormalization in the context of Arnoldi (what we mean by this will be formalized by a GMRES evaluation homomorphism presented below later). In order to clarify the actions and meanings of these definitions before verifying that this forms a group, we provide some examples.

Because GMRES generates the Krylov basis, we choose to work on a space that abstracts this and simply keeps track of how much of the basis has been developed by narrowing the scope of our space to $\mathbb{W}$. Elements of $\mathbb{W}$ can really be thought of as binary placeholders for the appropriate vectors of the Krylov basis (a fact that will be formalized as theorem 1 below). However, the space $\mathbb{W}$ may be generalized considerably, and the operation $\oplus$ may also be changed to explicitly include the orthonormalization operator.

However, doing so makes our presentation needlessly complex for our purpose, as changing the definitions presented to accomodate such an expansion amounts to acting trivially on the matrices outside of $\mathbb{W}$.

To clarify the definition of this $\mathbb{W}$ space we have made, we give an example of what one of these elements looks like.

Example 3.1.

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad r = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$
$$K = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix} \quad Q = \begin{pmatrix} \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{4}{3\sqrt{2}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & -\frac{10}{3\sqrt{2}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{10}{3\sqrt{2}} \end{pmatrix}$$
$$\begin{pmatrix} \frac{1}{\sqrt{3}} & 0 & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{3}} & 0 & 0 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & 0 & 0 \end{pmatrix} \in \mathbb{W}$$

The equality in the last line is meant to demonstrate and emphasize that we view the elements as equivalent up to permutation.

Because the dup_2 operation must be thought of as an addition and therefore be thought of as a binary operator, we also make the notation $[A, B]$ to mean the column concatenation of the nonzero columns vectors of the matrices A and B.

Because $dup_2([A_Q, B_Q]) = V$ lists the parallel vectors in $[A, B]$ that occur an odd number of times in a given orthonormal basis Q once and replace the ones that occur an even number of times with an all 0 vector, it can be thought of as analogous to a mod_2 operation (see theorem 1).

Example 3.2.

We denote e_i as the standard basis vectors.

$$Q = \{e_1, e_2, \cdots, e_n\}$$
$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$
$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$
$$[A, B] = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
$$dup_2([A, B]) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

With this, it is clear what our group addition $A \oplus B := dup_2([A, B])$ means.

Definition 0.5 (GMRES Ring)**.** *Note that for the space $A, B \in \mathbb{W}$, if we define $A \oplus B := dup_2([A, B])$, then this forms a group. By $Z_2[G]$ we mean the group ring taken with the field Z_2 and the group G, and we denote this by $R := (Z_2[G])$. Finally, we consider $C_s, 1 \leq s \leq n$ to be indeterminates, and we define $GMR := R[C_s]$ to be the polynomial ring taken over R with these indeterminates to be the* **GMRES ring**.

Proof. To show that this is well defined we only need to show that G forms a commutative group. To this end, we show the well-definedness, commutativity, closure, and inverse properties (associativity being trivial).

Note that since there are only n vectors in Q and $2n$ vectors in $[A, B]$, then by the definition of dup_2, the output can only have at most n nonzero vectors, so the output is of the same size as the elements of $\mathbb{W}$, and by definition of dup_2, is $\in \mathbb{W}$. So, the operation is closed, and since dup_2 only counts the total number of occurrences of a given vector, permuting their positions in $[A, B]$ does not influence this operation, so it is well-defined.

As emphasized in the example above the elements of $\mathbb{W}$ are equivalent up to permutation, so since the output of dup_2 is in $\mathbb{W}$, we have $A \oplus B = dup_2([A, B]) = dup_2([B, A]) = B \oplus A$.

We now show the existence of inverses. Given $A \in \mathbb{W}$, we need to find some $B \in \mathbb{W}$ so that $A \oplus B = 0$, but due to the dup_2 operation, $A \oplus A = 0$.

Thus $G = (\mathbb{W}, \oplus)$ forms an Abelian group. □

Before continuing on to discuss GMR, we emphasize that, in fact, the structure of G is very simple, and already quite well-known.

Theorem 1. *G is isomorphic to Z_2^n.*

Proof. To show this, we simply define the identifying isomorphism $[x_1, x_2, \ldots, x_n] = \psi(A) : G \to Z_2^n$, as follows. We let x_i be the number of occurrences of the ith column of the given matrix Q modulo 2. □

We have defined the structure as we have done here in order, instead of as Z_2^n directly, in order to emphasize the connection to the Krylov basis.

It is thus well-defined to refer to $R := Z_2[G]$, which is the group ring taken with Z_2 considered as a ring and G as the Abelian group. Finally, we consider $X_s, 1 \leq s \leq n$ to be n indeterminate variables, and we define $GMR := R[X_s]$ to be the polynomial ring taken over R with these indeterminates as the *$GMRES$ ring*.

We take a moment to clarify the various operations in the group ring structure of GMR with another example. Since the group ring has its own $+$ and $\cdot$, the notation for the operation $\oplus_Q$ may cause some confusion when distribution occurs. We first clarify this possible confusion before justifying our decision for $\oplus_Q$.

Example 3.3.

$$(A + B) \cdot (A) = ((A \oplus A) + (A \oplus B)) \tag{4}$$

The notational issue, as one can see, is that when distribution occurs over a $\cdot$, it uses the group operation which is not a multiplication but an $\oplus$, and the $+$ is a formal placeholding summation, whereas the $\oplus$ is not. The reason for keeping this is a rather unfortunate consequence of the homomorphisms below. Below, we will map the $+$ to $\oplus$ via a homomorphism. Even though GMR is meant as an algebraic infrastructure that is purely meant for means of maintaining rigour, the homomorphism below is the *ultimate* link of this structure's utility in reality. Thus we have chosen our notation to confer to reality.

As such, we shall now describe these relevant homomorphisms, as we can now make the link between GMRES and polynomial evaluation completely rigorous. One of these important homomorphims clarifies the link between GMR and the sequence of operations in GMRES by defining an evaluation homomorphism:

Definition 1.1 (GMRES Evaluation function)**.**

$$f(B[X_1, X_2, \cdots, X_n]) : GMR \to G \in \Re^{n \times n}$$

Given a fixed A and Q corresponding to A, f is defined by first applying the dup_2 operation to all of the $g \in G$, then fixing the columns so that the jth column corresponds to the jth column of Q, and then interpreting $X_s^i \cdot g, g \in G$ taken the sth vector of g and moving it over $j + i$th column and addending the necessary 0 vectors to make it an element of $\Re^{n \times n}$ ($+$ in R gets mapped to $\oplus$ in G, rendering $\cdot$ in R as trivial).

A colloquial way to understand this homomorphism is that it describes the elements of GMR by using X_s^i as referring to multiplication of a given column by A^i, and $\oplus$ as orthonormalization of a given vector with the previous vectors.

Example 3.4.

$$f(X_1 r + r) = A_1 r \oplus r$$

Is equivalent to one loop of Arnoldi, as the $A_1 r$ refers to multiplying the vector r by A and the $\oplus$ refers to the action resulting in obtaining the orthonormalized Krylov vectors associated to Ar and r. Likewise,

$$f(X_2(X_1 r + r) + (X_1 r + r)) = A_2(A_1 r \oplus r) \oplus (A_1 r \oplus r) = A_2(A_1(r)) \oplus (A_1(r)) \oplus r$$

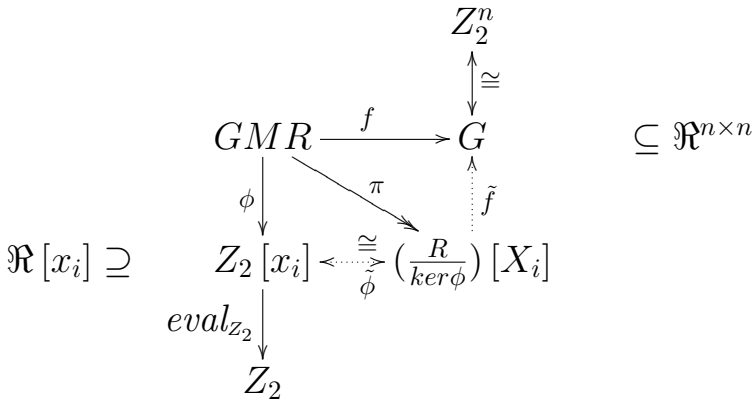
would be equivalent to two steps of Arnoldi.

Now we may clarify the link between GMR and $Z_2[x_i]$ that was alluded to previously.

Definition 1.2 (Polynomial Evaluation Homomorphism)**.** *We define the polynomial evaluation homomorphism as $\phi : GMR \to Z_2[x_{i,1\leq i\leq n}]$ where the indeterminate variable $X^k_{n-s+1+k}$ is mapped to x^k and the coefficient as part of the group ring is simply mapped to* 1*.*

We summarize and elucidate the connections between f, ϕ and theorem 1 with the following lemma.

Lemma 1.1. *For fixed A and Q corresponding to A, the following diagram commutes, where $\tilde{\phi}, \tilde{f}, \pi$ are defined in the proof, and $eval_{Z_2}$ refers to simply evaluating the a polynomial in $Z_2[x_i]$ with chosen elements in Z_2:*



Proof. Where we define π to be the surjection that takes an element in R and maps it to 0 if it's 0, otherwise it maps to 1, the morphisms of $\tilde{\phi}$ and $\tilde{f}$ are those induced by the isomorphism theorems[4]. □

As a final remark before showing some examples, we should note that the $\tilde{\phi}$ function we generated is useful as a tool for theoretical exploration. In a sense, it forms a 'pullback' due to the fact that it is an isomorphism with which we may use to take algorithms generated on $Z_2[x_i]$ into $\frac{R}{ker\phi}[X_i]$. In practice, these can typically be directly applied to GMR with no further modification.

4 Examples of Moving to and from GMRES and Polynomial Algorithms

We will now show some examples of the utility of explaining previous results with this homomorphism.

As a somewhat simpler example, we recall our motivating example, modified to be well-defined in the following spaces:

$$
\begin{array}{c|c}
(j = n-1:0) & \\
\text{Unabstracted Arnoldi's Iteration} & \text{Abstracted, Modified Horner's Rule} \\
V_n = \{v_0\} & b_n = 1 \\
V_j = \text{orthogonalize}\{V_{j+1}, A \cdot \text{last vector of } V_n\} & b_j = b_{j+1} + x \cdot b_{j+1}
\end{array}
$$

We make particular note that Arnoldi's iteration is unabstracted in the sense that none of the underlying algebras are yet applied. We define abstracted Arnoldi to be an expression which under f results in the unabstracted form of Arnoldi. We claim that such an expression exists, and confers to lemma 1.1 above.

Theorem 1. *Under ϕ, abstracted Arnoldi in sequential GMRES is equivalent to modified Horner's Rule.*

Proof. We apply the above definitions appropriately, where we let Q now be the the orthonormalization of the Krylov matrix relating to A and the vector $r_0 := b - Ax_0$ for some given vector x_0 (here we assume that the original equation $Ax = b$ satisfies the same requirements as in the theory above. I.e., that A is nonsingular, and the the degree of the minimum polynomial of r_0 with respect to A is n).

Because writing the vectors with respect to Q is equivalent to orthonormalization step, we may abstract Arnoldi's iteration by replacing the orthonormalization step with $+$, this is 'abstract Arnoldi's iteration', because under f it results in the same answer as j steps of Arnoldi.

Likewise, because ϕ maps to polynomials with coefficients in Z_2, we simplify the coefficients in Horner's rule to simply be 1, and denote this as 'abstracted Horner's rule'.

In summary, by applying ϕ to the elements on the left, we immediately obtain the elements on the right.

$$
\begin{array}{c|c}
(j = n-1:0) & \\
\text{Abstracted Arnoldi's Iteration} & \text{Abstracted Modified Horner's Rule} \\
V_n = expand(\{v_0\}) & b_n = 1 \\
V_j = V_{j+1} + (C_{n-j+1} \cdot V_{j+1})\} & b_j = b_{j+1} + x \cdot b_{j+1}
\end{array}
$$

□

Therefore, we have achieved our goal of making precise what we mean between thinking of GMRES as modified Horner's Rule, but we should note that this can be extended to other currently existing forms of GMRES and polynomial evaluation methods. In particular, modified Dorn's method, which we show as simply as we have done with modified Horner's rule.

Theorem 2. *Under ϕ, abstracted s-step GMRES is equivalent to modified Dorn's method[6].*

Proof. The proof runs similar as the case of modified Horner's rule. Only, we need to rewrite and express unabstracted s-step slightly differently to make the application of ϕ clearer.

We compare modified Dorn with s-step.

$$
\begin{array}{c|c}
(j = n-1:0) & \\
\text{Unabstracted s-step} & \text{Abstracted Modified Dorn's Rule} \\
V_n = \{v_0\} & b_j = 1, j = n, \cdots, n-k+1 \\
V_j = \text{orthogonalize } \{v_{j+1}, Av_{j+1}, \cdots, A^s v_{j+1}\} & b_j = b_{j+1} + x^k \cdot b_{j+1}
\end{array}
$$

However, we may rewrite $V_j = \{v_{j+1}, Av_{j+1}, \cdots, A^s v_{j+1}\} = \{A^s v_{j-k+1}, \cdots, A^s v_{j+1}\}$, using this we may now state an appropriately abstracted s-step

$$(j = n-1:0)$$

Abstracted s-step	Abstracted Modified Dorn's Rule
$V_j = A^{n-j} \cdot expand(\{v_0\}), j = n, \cdots, n-k+1$	$b_j = 1, j = n, \cdots, n-k+1$
$V_j = V_{j+1} + (C^k_{n-j+1+k} V_{j+1})$	$b_j = b_{j+1} + x^k \cdot b_{j+1}$

Keeping carefully in mind that ϕ was constructed so that $X^k_{n-s+1+k}$ maps to x^k.

□

So far, we have shown the great utility behind this ϕ homomorphism. However to be able to use this device to explain already currently existing GMRES and polynomial evaluation algorithms merely makes this a tool of hindsight as opposed to foresight. In this way, we will now see the true power of this framework, and that it allows us to view any GMRES algorithm as a much simpler polynomial evaluation algorithm, and, indeed, allows us to use this method to invent new ones. We include such an demonstration of how this is so, and how to use $\tilde{\phi}$ and $\tilde{f}$ which were induced in lemma 1.1 for such theoretical explorations.

Example 4.1.

One such idea is to allow the k powers in our modified Dorn's Method to vary along a Fibonacci pattern. Such as is done in Muraoka's algorithm[9, 10]. In particular, the following is a modified, abstracted Dorn's method following a Fibonacci pattern:

$$(j = n-1:0)$$
$$\text{Abstracted Modified Fibonacci Dorn's rule}$$
$$b_j = 1, j = n, \cdots, n-s(k)+1$$
$$b_j = b_{j+1} + x^{s(k)} \cdot b_{j+1}$$

(where $s(k)$ is a function that gives the kth Fibonacci number)
Under application of $\tilde{\phi}$ we obtain:

$$(j = n-1:0)$$
$$\text{Abstracted variable s-step}$$
$$V_j = A^{n-j} \cdot expand(\{v_0\}), j = n, \cdots, n-s(k)+1$$
$$V_j = V_{j+1} + (C^{s(k)}_{n-j+1+s(k)} V_{j+1})$$

And then under application of $\tilde{f}$, we can obtain the following is a skeleton for such a variable s-step GMRES implementation:

- $w_0 := r_0 = b - Ax_0, \beta := ||r_0||_2, v_1 = \frac{r_0}{\beta}$
- For j=1,2,3,5,8,...,s(j)
- $w_j := AM^{-1} w_{j-1}$
- End
- Orthogonalize $\{w_{s(j)}, w_{m-1}, \cdots, w_1\}$ to get V, store coefficients in H.
- $H_m := [h_{ij}]; y_m = min||\beta e_1 - H_m y||_2$
- $x_m := x_0 + W y_m$

It should be emphasized that after this exploration, we may *a posteriori* explain the abstracted variable s-step and abstracted modified Fibonacci Dorn's rule using f and ϕ, as this algorithm can work on GMR proper. We utilized the $\tilde{\phi}$ and $\tilde{f}$ here simply in order to explain their function in research.

However, in creating such an example we are forced to point out a central flaw in the above analysis: information about the inherent stability is not carried over. Repeated multiplication is necessary in order to perform the evaluation in polynomial methods, but with matrices repeated matvecs pose serious limitations. However, if one poses a somewhat *ad hoc* constraint–such as placing a cap on $s(j)$. We do not fully analyze such an algorithm here, but merely present it as a possible theoretical application of the infrastructure developed above.

5 Conclusion

We have simplified the actions of GMRES down to the actions of a polynomial. By constructing a binary representation to keep track of the orthonormalization process in GMRES, one can now think of the orthonormalization steps in GMRES as a sort of primitive addition operator and the matvec steps as a sort of primitive multiplication operator. Once this framework is set up, it allows the comparison of various GMRES methods to much simpler–and already well known and well studied–polynomial evaluation algorithms.

But, the algebra introduced includes other possibilities for improvement of a different calibre as well that would do well to be noted here. Namely, to re-analyze the polynomial evaluation literature in the context of significantly changing multiplication and addition costs, so that under the polynomial evaluation homomorphism of definition 1.2, we could obtain a new optimal algorithm similar to what was undertaken in [9]; and then to pull back this polynomial evaluation algorithm to a GMRES-like algorithm similar to what was done in theorem 2.

We should also note that in order to make such research plausable, it is necessary for the homomorphisms presented in this paper to not only carry information regarding the sequence of matrix operations, but also the cost of such operations. The algorithm developed from such an investigation might further improve or possibly help clarify a singular optimal (in terms of computational and communication costs) GMRES algorithm.

If we wish to specify what the most theoretically optimal GMRES algorithm is, the following work is needed for the future.

1. The homomorphism must be changed to also pass information on the orthogonalization and matvec cost and accuracy.

2. The literature of parallel polynomial evaluation assumes that the computational cost, communication cost, and flop error associated to multiplication and addition is the same. All of the literature on parallel polynomial evaluation would have to be reworked with this new assumption.

3. Apply the previous polynomial evaluation analysis to this new set of literature to obtain an optimal parallel polynomial evaluation algorithm where the multiplication and addition's cost and accuracy vary.

4. Use the homomorphism $\tilde{\phi}$ to help 'pullback' to obtain the corresponding optimal GMRES algorithm.

Another interesting theoretical detail overlooked in the above, but should be noted for future work is the dup_2 operation. This dup function can be generalized so that dup_i would list parallel vectors modulo i instead of modulo 2 as occurs in this paper. We choose dup_2 because we believe it effectively replicates the breakdown condition of GMRES. Although we should note it might also be possible to define such a dup_i algebra, and then trace its influences back to its relation on GMRES in order to obtain a different breakdown condition of GMRES. Doing so may be theoretically interesting, and may add to the theory

regarding the underlying algebraic substructures along with a better view of the algebra of the breakdown case. However, we did not do so as this would have served to complicate the definition beyond our aims in this paper.

References

[1] Chronopoulos, A. T. (1991). s-Step iterative methods for (non) symmetric (in) definite linear systems. *SIAM journal on numerical analysis*, 28(6), 1776-1789.

[2] Gentleman, W. M. (1973). Least squares computations by Givens transformations without square roots. *IMA Journal of Applied Mathematics*, 12(3), 329-336.

[3] Hoemmen, M. (2010). Communication-avoiding Krylov subspace methods.

[4] Hungerford, T. (2012). *Abstract algebra: an introduction*. Cengage Learning.

[5] Jalby, W., & Philippe, B. (1991). Stability analysis and improvement of the block Gram-Schmidt algorithm. *SIAM journal on scientific and statistical computing*, 12(5), 1058-1073.

[6] Knuth, D. E. (2014). Art of Computer Programming, Volume 2: Seminumerical Algorithms, The. Addison-Wesley Professional.

[7] Li, G. (1997). A block variant of the GMRES method on massively parallel processors. Parallel Computing, 23(8), 1005-1019.

[8] Morgan, R. B. (2002). GMRES with deflated restarting. *SIAM Journal on Scientific Computing*, 24(1), 20-37.

[9] Muraoka, Y. (1971). Parallelism exposure and exploitation in programs.

[10] Munro, I., & Paterson, M. (1973). Optimal algorithms for parallel polynomial evaluation. *Journal of Computer and System Sciences*, 7(2), 189-198.

[11] Saad, Y., & Schultz, M. H. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on scientific and statistical computing, 7(3), 856-869.

[12] De Sturler, E., & van der Vorst, H. A. (1995). Reducing the effect of global communication in GMRES(m) and CG on parallel distributed memory computers. *Applied Numerical Mathematics*, 18(4), 441-459.

[13] Van der Vorst, H. A., & Vuik, C. (1994). GMRESR: A family of nested GMRES methods. Numerical linear algebra with applications, 1(4), 369-386.

[14] Wakam, D. N., & Erhel, J. (2013). Parallelism and robustness in GMRES with a Newton basis and deflated restarting. *Electronic Transactions on Numerical Analysis*, 40, 381-406.