Journal of Progressive Research in Mathematics(JPRM)
ISSN: 2395-0218

SCITECH
RESEARCH ORGANISATION|

Volume 11, Issue 1

Published online: January 14, 2017|

Journal of Progressive Research in Mathematics
www.scitecresearch.com/journals

Vector Space Ideas Computing A^m

Hamide Dogan
UTEP Mathematics EL Paso, TX
United States

Abstract.

We discuss an algorithm with a simplistic approach to solving systems of linear equations arising from the application of real-valued vector space ideas to the computation of the large powers of square matrices.

Keywords: Matrix Power Algorithm; Linear Algebra; Vector Spaces; Linear Systems.

1. Introduction

Large powers of matrices are commonly applied to many areas such as graph theory models and population prediction models etc. So far, the calculation of powers of matrices however requires computationally taxable algorithms that need and require necessary number of eigenvalues of matrices for diagonalization [4], or algorithms that need to apply computer-based memory-consuming recursive steps [2,3]. In this article, we derive a simple approach of solving systems of linear equations in order to obtain the large powers of square matrices with very little memory usage. Our approach (we named "*Augmented Approach*") makes use of the division algorithm over the ring of polynomials [1], and it does not require any recursive process or the computation of the eigenvalues of matrices.

2. Vector Space Ideas

Using the vector space structure of the set of polynomials of degree m, P_m, over the field of Real numbers, one writes any polynomial of degree m (or less), G(x)=$\sum_0^m z_i x^i$, applying the division algorithm using a basis of the vector space [1] as:

$$G(x)=\sum_0^m z_i x^i = \sum_0^{n-1} a_i x^i + \sum_0^{m-n} b_i x^i F \quad where\ F(x)= \sum_0^n y_i x^i\ for\ n\leq m\ (1)$$

Here, $\{1, x, x^2, \ldots, x^{n-1}, F, xF, x^2F, \ldots, x^{m-n}F\}$ is the basis of P_m, used in (1), which guarantees a unique linear combination of the basis vectors resulting in G(x) [1].

Solving equation (1) for the unknowns b_i and a_i leads to the recognition of two separate systems of linear equations. One of the two systems contains equations with only the unknowns *bi*, and the other includes equations with each having single unknown, a_i.

Equations of the first system are obtained by comparing the coefficients of the terms x^{m-k}. These coefficients form a *((m-n)+1)x((m-n)+2)* system of linear equations(*Here, we set $y_j=0$ when $j<0$*):

$$\sum_{i=0}^{k} b_{m-n-k+i}\, y_{n-i} = z_{m-k},\ k=0,\ldots,m-n \quad (2)$$

It is easy to verify that (2) gives the coefficients of x^{m-k}: $b_{m-n-k+i}$ is the coefficient of $x^{m-n-k+i}F$, and y_{n-i} is the coefficient of the term x^{n-i} in F. Thus, (2) picks the coefficient of the term $x^{m-n-k+i}F$, and matches with the coefficient of x^{n-i} making sure the combined power of x is adding upto the power *m-k*. For instance, for the largest possible power *m* of *x*, equation in (2) considers *k=0*, and picks the coefficient b_{m-n} of $x^{m-n}F$ with the highest power, and pairs it with the coefficient y_n of x^n in *F* with the highest power, which are the only combinations adding upto the largest power x^m. In short, (2) considers the coefficients of *x* ranging from *n* to *m*.

An interesting result emerges when considering the augmented matrix of the system in (2). One obtains a nicely patterned ((m-n)+1)x((m-n)+2) triangular matrix seen in (3):

$$\begin{bmatrix} 0 & 0 & \ldots & 0 & y_n & z_m \\ 0 & \ldots. & & y_n & y_{n-1} & z_{m-1} \\ 0 & & y_n & y_{n-1} & y_{n-2} & z_{m-2} \\ \ldots. & \ldots & & \ldots & \ldots & \ldots \\ y_n & \ldots. & & y_{n-(m-n)-1} & y_{n-(m-n)} & z_n \end{bmatrix} \quad (3)$$

In light of the triangular aspect of the matrix in (3), readers may agree that solving the system in (2) requires only the basic back substitution technique. Also it is easy to verify that this system has a unique solution whenever $y_n \neq 0$.

After obtaining the values of the unknowns, *bi*, it is not hard to evaluate the values of a_i. All one needs is to substitute b_i values to the single unknown equations given in (4) below, and apply basic arithmetic. The values of a_i are in turn forming the remainder polynomial in the division algorithm (1).

It can easily be verified that the equations in (4) are obtained from comparing the coefficients of x^i where the index *i* ranges from *0* to *n-1*:

$$a_i = z_i - \sum_{k=0}^{i} b_{i-k} y_k, \quad i=0,\dots,n-1 \tag{4}$$

We should note here that equations in (2) could also be obtained from the expression in (4) by extending the values of the index *i* from *n-1* to*m*, and adding the following two conditions: $b_t=0$ *if* $t>m-n$ & $y_s=0$ *if* $s>n$.

Example:

Let us consider a polynomial *F* with degree 2, and a polynomial *G* with degree 5. Then,

$F(x)=\sum_{i=0}^{2} y_i x^i$ and $G(x)=\sum_{i=0}^{5} z_i x^i$. Using the equation in (1), we get:

$$z_0 + z_1x + z_2x^2 + z_3x^3 + z_4x^4 + z_5x^5 = a_0 + a_1x + b_0F + b_1xF + b_2x^2F + b_3x^3F \tag{5}$$

Solving (5) via the comparison of the coefficients of x^k, $k=0, \dots, m$, we get the values of the unknowns b_i and a_i.

Equations with unknowns a_i*:*

- Comparing the constant terms in (5) gives the equation: $a_0 = z_0 - b_0 y_0$.

This further verifies the equation in (4) for $i = 0$.

- Comparing the coefficients of the x-terms in (5) gives the equation: $a_1 = z_1 - b_0 y_1 - b_1 y_0$;

This verifies the equation in (4) for $i=1$.

Equations with Unknowns b_i*:*

- Comparing the coefficients of the x^2-terms in (5) gives the equation: $z_2 = b_0 y_2 + b_1 y_1 + b_2 y_0$;

This time, the formula in (2) can be verified by setting *k=m-n=3*: $\sum_{i=0}^{3} b_{5-2-3+i} y_{2-i} = z_{5-3}$.Next, expanding out the summation gives:

$b_0 y_2 + b_1 y_1 + b_2 y_0 + b_3 y_{-1} = z_2$.

Finally, setting y_{-1}=0, gives the linear equation: $b_0 y_2 + b_1 y_1 + b_2 y_0 = z_2$.

- Comparing the coefficients of the x^3-terms in (5) gives the equation: $z_3 = b_1 y_2 + b_2 y_1 + b_3 y_0$.

  Again, one can easily verify the formula in (2) by setting k=2.

- Comparing the coefficients of the x^4-terms in (5) gives the equation: $z_4 = b_2 y_2 + b_3 y_1$.

  The same equation emerging from the expression in (2) can be verified by setting k=1.

- Comparing the coefficients of the x^5-terms in (5) gives the equation: $z_5 = b_3 y_2$. One can easily verify (2) setting k=0.

Notice that the coefficients of x^k, *k=2,3,4,5* forms a 4x5 linear system containing only the

unknowns, b_i, *i=0, 1,2,3*. Furthermore, its augmented matrix is a 4x5 triangular matrix with a nice pattern as seen in (6). The pattern emerged in (6) is also in agreement with the matrix form given in (3).

$$\begin{bmatrix} 0 & 0 & 0 & y_2 & z_5 \\ 0 & 0 & y_2 & y_1 & z_4 \\ 0 & y_2 & y_1 & y_0 & z_3 \\ y_2 & y_1 & y_0 & 0 & z_2 \end{bmatrix} \qquad (6)$$

The triangular and easily recognized pattern in (6) would make it easier to input these matrices into computer algorithms, and furthermore it would be straightforward for the computer-based programs to solve the systems easily simply by applying basic back-substitution techniques thus using drastically less memory than memory used with many of the existing approaches such as eigenvalue approach to diagonalizing matrices [4], and recursive approach to obtaining remainder polynomials in division algorithms [2,3].

3. Computing A^m

In fact, the pattern observed in the matrix (3) has implications for the calculation of higher powers of matrices. It has been known that the Division Algorithm and Cayley-Hamilton Theorem provide means to be able to calculate a large power of a square matrix by evaluating a remainder polynomial, obtained via division algorithm, for the matrix via a recursive process [2, 3]. In this case, F polynomial in the equation (1) takes the role of the characteristic polynomial of a square matrix, and the polynomial G considers only the x^m-type polynomials. The recursive process however can be memory consuming for computer-based algorithms to compute large values of the powers of remainder polynomials. Our *Augmented Approach* drastically cuts down the processes of finding a remainder polynomial thus provides a more effective means to obtain a remainder polynomial and as a result more efficient calculation of A^m for the large values of m.

3.1 Augmented Approach

Our approach gives the nicely pattern triangular augmented matrix below in (7) obtained by considering the characteristic polynomial, F, of an *nxn* matrix, and applying the division algorithm (1) to the characteristic polynomial and the polynomial $G(x) = x^m$. Moreover, since we have $y_n = \pm 1$ in the characteristic polynomial of any square matrix, the linear system represented by the matrix in (7) always has a unique solution.

$$\begin{bmatrix} 0 & 0 & \dots & 0 & \pm 1 & 1 \\ 0 & \dots & & \pm 1 & y_{n-1} & 0 \\ 0 & & \pm 1 & y_{n-1} & y_{n-2} & 0 \\ \dots & \dots & & \dots & . & \dots \\ \pm 1 & \dots & & y_{n-(m-n)-1} & y_{n-(m-n)} & 0 \end{bmatrix} \qquad (7)$$

***Example*:**

Let's now calculate A^6 using the *Augmented Approach* where A= $\begin{bmatrix} 2 & 1 & 0 \\ 0 & 3 & 2 \\ 0 & 0 & 4 \end{bmatrix}$. Its characteristic polynomial is $F(x) = 24 - 26x + 9x^2 - x^3$. Thus, for this example, the augmented matrix in (7) becomes a 4x5 triangular matrix seen in (8) representing the linear system obtained by applying the equation in (1):

$$\begin{bmatrix} 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 9 & 0 \\ 0 & -1 & 9 & -26 & 0 \\ -1 & 9 & -26 & 24 & 0 \end{bmatrix} \quad (8)$$

The row reduced echelon form of the matrix in (8) or solving the system using back-substitution beginning with the top row in (8), $-b_3=1$, gives the values:

$$b_0 = -285$$
$$b_1 = -55$$
$$b_2 = -9$$
$$b_3 = -1$$

Using these values, we easily evaluate the equations in (4) thus obtaining the remainder polynomial:

$$R(x)=6840-6090x+1351x^2$$

Now, we are ready to compute A^6:

$$A^6=R(A)= 6840I-6090A+1351A^2 = \begin{bmatrix} 64 & 665 & 2702 \\ 0 & 729 & 6734 \\ 0 & 0 & 4096 \end{bmatrix}.$$

As a final remark regarding the higher powers of A in the example above, since y values are the same, the augmented matrix for higher powers will be a spare matrix with many zeros thus requiring about the same number of steps as the lower powers to calculate A^m. As an example, for A^{100}, the augmented matrix would be a $98x99$ triangular matrix, seen in (9),with only non-zero entries coming from the values of y_n of the characteristic polynomial $F$of A.

$$\begin{bmatrix} 0 & 0 & \ldots & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 9 & 0 \\ 0 & \ldots & \ldots & 0 & -1 & 9 & -26 & 0 \\ 0 & 0 & 0 & \ldots. & 9 & -26 & 24 & 0 \\ \ldots & 0 & -1 & \ldots & -26 & 24 & 0 & 0 \\ 0 & -1 & 9 & \ldots & \ldots & \ldots & \ldots & \ldots \\ -1 & 9 & -26 & \ldots & 0 & 0 & 0 & 0 \end{bmatrix} \quad (9)$$

Notice that using our augmented matrix approach we directly obtained the coefficients of the remainder polynomial R for the calculation of A^6 by simply solving a single linear system. That is, we did not need to compute the coefficients of the remainder polynomials for the earlier degrees, A^3, A^4,and A^5. An already existing recursive approach derived from the division algorithm however requires the coefficients of the earlier remainder polynomials with lesser degrees calculated to be able to identify the coefficients of the remainder polynomial for the desired degree hence requiring a recursive process [2, 3].

References

[1] A. Von Sohsten de Medeiros (2002) *Elementary Linear Algebra and the Division Algorithm*. The College Mathematics Journal. The Mathematical Association of America.

[2] R. Abu-Saris and W. Ahmad(2005) *Avoiding Eigenvalues in Computing Matrix Power*, The American Mathematical Monthly , Vol. 112, Number 5, (May 2005) 450-454.

[3] S.N. Elaydi and W. A. Harris, Jr. (1997) *On the computation of* A^n, SIAM Rev. 40 (1998) 965-971.

[4] A. Sheldon (1997). *Linear Algebra Done Right*. Second Edition. Springer.