

The Tridiagonal Polynomial Eigenvalue Problem

Thomas R. Cameron

December 2018

Abstract

This note introduces Hyman's method for matrix polynomials and outlines our potential research project. In particular, we are interested in applying Hyman's method to solve for the eigenvalues and eigenvectors of a tridiagonal matrix polynomial. Our research will entail the development and analysis of this method, including cost analysis and backward stability analysis, initial conditions, and computation of eigenvectors.

1 Hyman's Method for Matrix Polynomials

Hyman's method, a method for evaluating the characteristic polynomial and its derivatives at a point, is attributed to a conference presentation given by M.A. Hyman of the Naval Ordnance Laboratory in 1957 [9]. The backward stability of this method has been shown in [5, 9], and this method has been used to evaluate the characteristic polynomial of a matrix [7] and matrix pencil [4]. In this section, we derive Hyman's method for matrix polynomials.

We denote an upper Hessenberg matrix polynomial as follows

$$H(\lambda) = \begin{bmatrix} h_{11}(\lambda) & h_{12}(\lambda) & \cdots & h_{1n}(\lambda) \\ h_{21}(\lambda) & h_{22}(\lambda) & \cdots & h_{2n}(\lambda) \\ & \ddots & \ddots & \vdots \\ & & h_{n,n-1}(\lambda) & h_{nn}(\lambda) \end{bmatrix},$$

where $h_{ij}(\lambda)$ is a scalar polynomial of degree at most m . Let $\mu \in \mathbb{C}$ and suppose that $H(\mu)$ is a proper upper Hessenberg matrix, i.e., $h_{i+1,i}(\mu) \neq 0$ for all $i \in \{1, 2, \dots, n-1\}$. Following the development in [5, Section 14.6.1], denote $H(\mu)$ in block form as follows

$$H(\mu) = \begin{bmatrix} h^T(\mu) & \eta(\mu) \\ R(\mu) & y(\mu) \end{bmatrix},$$

where $h(\mu), y(\mu) \in \mathbb{C}^{n-1}$, $\eta(\mu) \in \mathbb{C}$, and T denotes the transpose.

Now, consider the cyclically permuted matrix

$$\hat{H}(\mu) = \begin{bmatrix} R(\mu) & y(\mu) \\ h^T(\mu) & \eta(\mu) \end{bmatrix},$$

and note that $\det(\hat{H}(\mu)) = (-1)^{n-1} \det(H(\mu))$. Furthermore, since $H(\mu)$ is a proper upper Hessenberg matrix, it follows that $R(\mu)$ is an invertible upper triangular matrix. We have the following LU factorization

$$\hat{H}(\mu) = \begin{bmatrix} I & 0 \\ h^T(\mu)R^{-1}(\mu) & 1 \end{bmatrix} \begin{bmatrix} R(\mu) & y(\mu) \\ 0 & \eta(\mu) - h^T(\mu)R^{-1}(\mu)y(\mu) \end{bmatrix}.$$

Let $p(\mu) = \det(H(\mu))$, then we have

$$p(\mu) = (-1)^{n-1} \det(R(\mu)) \left(\eta(\mu) - h^T(\mu)R^{-1}(\mu)y(\mu) \right).$$

Hyman's method consists of evaluating the above equation in the natural way: Solving the triangular system $R(\mu)x(\mu) = y(\mu)$, then forming $q(\mu) = \eta(\mu) - h^T(\mu)x(\mu)$ and its product with $r(\mu) = \det(R(\mu))$. Note that Hyman's method is prone to overflow and underflow in the computation of $\det(R(\mu))$. However, we can avoid this problem by computing the Laguerre correction terms directly.

To this end, note that

$$p'(\mu) = (-1)^{n-1} (r'(\mu)q(\mu) + r(\mu)q'(\mu)).$$

Therefore,

$$\frac{p'(\mu)}{p(\mu)} = \frac{r'(\mu)}{r(\mu)} + \frac{q'(\mu)}{q(\mu)}$$

Let $R(\mu) = [r_{ij}(\mu)]$, then

$$\frac{r'(\mu)}{r(\mu)} = \sum_{i=1}^n \frac{r'_{ii}(\mu)}{r_{ii}(\mu)}.$$

Furthermore,

$$q'(\mu) = \eta'(\mu) - (h'^T(\mu)x(\mu) + h^T x'(\mu)),$$

where

$$R(\mu)x'(\mu) = y'(\mu) - R'(\mu)x(\mu).$$

Similarly, we have

$$-\left(\frac{p'(\mu)}{p(\mu)}\right) = \left(\frac{r'(\mu)}{r(\mu)}\right)^2 - \frac{r''(\mu)}{r(\mu)} + \left(\frac{q'(\mu)}{q(\mu)}\right)^2 - \frac{q''(\mu)}{q(\mu)},$$

where

$$\frac{r''(\mu)}{r(\mu)} = \left(\frac{r'(\mu)}{r(\mu)}\right)' + \left(\frac{r'(\mu)}{r(\mu)}\right)^2.$$

Therefore, we have

$$\begin{aligned} -\left(\frac{p'(\mu)}{p(\mu)}\right) &= \left(\frac{q'(\mu)}{q(\mu)}\right)^2 - \frac{q''(\mu)}{q(\mu)} - \left(\frac{r'(\mu)}{r(\mu)}\right)' \\ &= \left(\frac{q'(\mu)}{q(\mu)}\right)^2 - \frac{q''(\mu)}{q(\mu)} - \sum_{i=1}^n \left(\frac{r''_{ii}(\mu)}{r_{ii}(\mu)} - \left(\frac{r'_{ii}(\mu)}{r_{ii}(\mu)}\right)^2\right). \end{aligned}$$

Finally, note that

$$q''(\mu) = \eta''(\mu) - (h''^T(\mu)x(\mu) + 2h'^T(\mu)x'(\mu) + h^T x''(\mu)),$$

where

$$R(\mu)x''(\mu) = y''(\mu) - (R''(\mu)x(\mu) + 2R'(\mu)x'(\mu)).$$

Algorithm 1 Hyman's Method

Solve $R(\mu)x(\mu) = y(\mu)$ using backward substitution

Solve $R(\mu)x'(\mu) = y'(\mu) - R'(\mu)x(\mu)$ using backward substitution

Solve $R(\mu)x''(\mu) = y''(\mu) - (R''(\mu)x(\mu) + 2R'(\mu)x'(\mu))$ using backward substitution

Compute

$$q(\mu) = \eta(\mu) - h^T(\mu)x(\mu)$$

$$q'(\mu) = \eta'(\mu) - (h'^T(\mu)x(\mu) + h^T x'(\mu))$$

$$q''(\mu) = \eta''(\mu) - (h''^T(\mu)x(\mu) + 2h'^T(\mu)x'(\mu) + h^T x''(\mu))$$

Compute $\frac{r'(\mu)}{r(\mu)}$ and $\left(\frac{r'(\mu)}{r(\mu)}\right)'$

return $\frac{r'(\mu)}{r(\mu)} + \frac{q'(\mu)}{q(\mu)}$ and $\left(\frac{q'(\mu)}{q(\mu)}\right)^2 - \frac{q''(\mu)}{q(\mu)} - \left(\frac{r'(\mu)}{r(\mu)}\right)'$

2 Outline of Research Project

Algorithm 1 provides an effective way for computing the Laguerre correction terms necessary to use Laguerre’s method to compute all the eigenvalues of a matrix polynomial. This method is cost effective and backward stable, but there are still several points that are worth investigation. Below is an itemized list of points that we should consider during our research project.

- We developed Hyman’s method for upper Hessenberg matrix polynomials, but we are likely to have the most applications with tridiagonal matrix polynomials. We will need to specialize Algorithm 1 for tridiagonal matrix polynomials. In particular, the cost analysis and backward stability will be different.
- Algorithm 1 only works if $H(\mu)$ is proper. There are several ways for dealing with the non-proper case. We should develop some rudimentary code to test these cases in order to make the best decision.
- On a similar note to the previous item, we should compare the Laguerre method to the Ehrlich-Aberth method when the correction terms are computed using Hyman’s method. The modified Laguerre method I published in [3] is faster than the Ehrlich-Aberth method for scalar polynomials, it will be interesting to see if the same holds true for matrix polynomials.
- The reduction of matrix polynomials to simpler forms is a difficult problem, see [6]. However, I have seen matrix polynomials come in Hessenberg form. For instance, the *bilby* problem in [1]. It would be interesting to find out if there are applications where upper Hessenberg matrix polynomials occur naturally. Whether or not we find applications for upper Hessenberg form will dictate the development of our paper and software. For instance, if we find applications we will write software for both the upper Hessenberg and tridiagonal structures; if no application for upper Hessenberg form exists, then we will only write software for the tridiagonal structure. In either case, I plan on developing the theory through the Hessenberg case in the paper as there is no benefit of simplicity gained from jumping to the tridiagonal case immediately.
- The tridiagonal eigenvalue problem is a famous problem. We should investigate the literature to see what else has been used for this problem and what software exists today, for example see [2, 8]. It would be a good start to see what papers have cited these references and collect any software that is still online.
- We also want to investigate the literature for test problems. Again, a good place to start is the tests in [2, 8]. As we write our software we will want to do thorough testing. The publishing of our work will depend greatly on how much testing we do and the quality thereof.

My goal over the next few weeks is to start putting together a backward error analysis of Algorithm 1. I want to start with this problem since most of it has been done before and it will give us a good understanding of what to put in our paper for the analysis of our algorithm, especially in the tridiagonal case. Also, I want you to see a backward error analysis since it will likely be your first time. As I work on the analysis of Algorithm 1, I will begin developing code for the Hessenberg and tridiagonal case.

You will be able to help in the software development and testing once you are on campus and I can give you an overview of Fortran and how I am developing the software. In the meantime you can help by collecting data, whether its old articles, software, applications, or test problems. The more resources we have the better we can make our software and paper.

If you have some time to study, I highly recommend reading from [5]. In particular, Chapter 2 and 3 of [5] would be most helpful. In addition, any Fortran tutorial you can find would likely give you an idea of some of the basics you will need. You can install the Fortran compiler, simply type `brew install gcc` in the terminal, assuming you have homebrew installed.

References

- [1] T. BETCKE, N. J. HIGHAM, V. MEHRMANN, C. SCHRÖDER, AND F. TISSEUR, *NLEVP: a collection of nonlinear eigenvalue problem*, ACM Trans. Math. Softw., 39 (2013).
- [2] D. A. BINI, L. GEMIGNANI, AND F. TISSEUR, *The Ehrlich-Aberth method for the nonsymmetric tridiagonal eigenvalue problem*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 153–175.
- [3] T. R. CAMERON, *An effective implementation of a modified laguerre method for the roots of a polynomial*, Numer. Algor., (2018), pp. 1–20.

- [4] J. GARY, *Hyman's method applied to the general eigenvalue problem*, Math. Comp., 19 (1965), pp. 314–316.
- [5] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 2002.
- [6] Y. NAKATSUKASA, L. TASLAMAM, F. TISSEUR, AND I. ZABALLA, *Reduction of matrix polynomials to simpler forms*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 148–177.
- [7] B. PARLETT, *Laguerre's method applied to the matrix eigenvalue problem*, Math. Comp., 18 (1964), pp. 464–485.
- [8] B. PLESTENJAK, *Numerical methods for the tridiagonal hyperbolic quadratic eigenvalue problem*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1157–1172.
- [9] J. H. WILKINSON, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1963.