# On a Nonlinear Matrix Equation Arising in Nano Research

*Chun-Hua Guo, Yueh-Cheng Kuo, and Wen-Wei Lin*

Abstract

The matrix equation $X + A^\top X^{-1} A = Q$ arises in Green's function calculations in nano research, where $A$ is a real $n \times n$ matrix and $Q$ is a real symmetric matrix dependent on a parameter and is usually indefinite. In practice one is mainly interested in those values of the parameter for which the matrix equation has no stabilizing solutions. The solution of interest in this case is a special weakly stabilizing complex symmetric solution $X_*$, which is the limit of the unique stabilizing solution $X_\eta$ of the perturbed equation $X + A^\top X^{-1} A = Q + i\eta I$, as $\eta \to 0^+$. It has been shown in [1] that a doubling algorithm can be used to compute $X_\eta$ efficiently even for very small values of $\eta$, thus providing good approximations to $X_*$. It has been observed by nano scientists that a modified fixed-point method can sometimes be very efficient, particularly for computing $X_\eta$ for many different values of the parameter. We provide a rigorous analysis of this modified fixed-point method and its variant, and of their generalizations. We also show that the imaginary part $X_I$ of the matrix $X_*$ is positive semidefinite and determine the rank of $X_I$ in terms of the number of unimodular eigenvalues of the quadratic pencil $P(\lambda) = \lambda^2 A^\top - \lambda Q + A$. Finally we present a structure-preserving algorithm that can find $X_*$ directly from the equation $X + A^\top X^{-1} A = Q$, under the assumption that all unimodular eigenvalues $\lambda \neq \pm 1$ of $P(\lambda)$ are semisimple and the eigenvalues $\pm 1$ (if any) have partial multiplicities 2. The algorithm computes $X_*$ by finding a suitable invariant subspace of a symplectic pencil $\mathcal{M} - \lambda\mathcal{L}$ that is a linearization of $P(\lambda)$. It starts with using the $(S + S^{-1})$-transform in [2] to transform $\mathcal{M} - \lambda\mathcal{L}$ into a new pencil $\mathcal{K} - \lambda\mathcal{N}$, where $\mathcal{K}$ and $\mathcal{N}$ are both skew-Hamiltonian. This new pencil is then reduced using the Patel approach in [3]. A suitable invariant subspace of $\mathcal{K} - \lambda\mathcal{N}$ is then obtained by performing eigen computations for an $n \times n$ matrix pencil, and is converted to the required invariant subspace of the pencil $\mathcal{M} - \lambda\mathcal{L}$. The computational work of this algorithm is roughly 1/4 of that for the QZ algorithm applied directly on the $2n \times 2n$ pencil $\mathcal{M} - \lambda\mathcal{L}$. Numerical results show that the accuracy of $X_*$ from the structure-preserving algorithm is as good as that from the QZ algorithm.

## References

[1] C.-H. Guo and W.-W. Lin, *The matrix equation $X + A^\top X^{-1} A = Q$ and its application in nano research*, SIAM J. Sci. Comput., 32 (2010), pp. 3020–3038.

[2] W.-W. Lin, *A new method for computing the closed-loop eigenvalues of a discrete-time algebraic Riccati equation*, Linear Algebra Appl., 96 (1987), pp. 157–180.

[3] R. V. Patel, *On computing the eigenvalues of a symplectic pencil*, Linear Algebra Appl., 188-189 (1993), pp. 591–611.

# A Fast Algorithm for Approximating the Distance to Instability

*Mert Gurbuzbalaban and Michael L. Overton*

Abstract

An $n \times n$ complex matrix $A$ is *stable* if and only if the *spectral abscissa* of $A$ (the maximum of the real parts of the eigenvalues, denoted $\alpha$), is negative. However, $\alpha$ is not a robust measure of stability, as small perturbations to the matrix can result in large changes to the eigenvalues. A natural alternative measure is the *distance to (continuous) instability* of $A$, that is, the distance from $A$ to the set of unstable matrices:

$$d_{\rm CI}(A) \quad = \quad \sup\{\epsilon : A + E \text{ is stable for all } E \text{ with } \|E\| \le \epsilon\} \tag{1}$$

where $\|\cdot\|$ denotes the matrix 2-norm. In the systems and control theory community, $d_{\rm CI}$ is known as the complex stability radius [4]. Byers [1] gave the first algorithm for computing $d_{\rm CI}$ to arbitrary accuracy; this bisection method introduced the key idea of determining whether or not $d_{\rm CI}(A) \ge \delta$ for a given $\delta$ by verifying whether any of the eigenvalues of a certain Hamiltonian matrix are imaginary. Soon afterwards, quadratically convergent variations appeared; these were developed for the more general problem of computing the $H_\infty$ norm of the transfer function for a linear dynamical system with input control and output measurement. These algorithms depend on computing eigenvalue decompositions of Hamiltonian matrices of order $2n$ and SVDs of order $n$, so the cost is effectively $O(n^3)$ and they are applicable only to small or moderate-sized systems.

In order to handle larger matrices, we propose a fast algorithm that avoids Hamiltonian eigenvalue decompositions and SVDs. The only allowable matrix operation is computing the rightmost eigenvalue of a given matrix $X$ and its corresponding right and left eigenvectors, which is done by an iterative algorithm such as implicitly restarted Arnoldi or biorthogonal Lanczos using matrix-vector products. (A rightmost eigenvalue $\lambda$ of $X$ is one for which $\mathrm{Re}\,\lambda = \alpha(X)$.) Our method is inspired by the following recent algorithm [2] developed to compute the *$\epsilon$-pseudospectral abscissa*, denoted $\alpha_\epsilon$. The quantity $\alpha_\epsilon(A)$ is the largest of the real parts of the elements in the $\epsilon$-pseudospectrum $\Lambda_\epsilon(A)$ (the set of points in the complex plane which are eigenvalues of $A + \Delta A$ for some $\Delta A$ with $\|\Delta A\| \le \epsilon$).

**Algorithm 1**

1. Let $z_0$ be an eigenvalue of $A$, with corresponding right and left eigenvectors $x_0$ and $y_0$ normalized so $\|x_0\| = \|y_0\| = 1$ and $y_0^* x_0 \in (0, 1]$. If $\alpha(A + \epsilon\, y_0\, x_0^*) \ge \mathrm{Re}(z_0) + \epsilon y_0^* x_0$, set $E_1 = y_0\, x_0^*$; otherwise, set $E_1 = x_0\, y_0^*$. Set $B_1 = A + \epsilon E_1$.

2. For $k = 1, 2, \ldots$: let $z_k$ be the rightmost eigenvalue of $B_k$, with right and left eigenvectors $x_k$ and $y_k$ normalized so that $\|x_k\| = \|y_k\| = 1$ and $y_k^* x_k \in (0, 1]$. Set $E_{k+1} = y_k\, x_k^*$ and $B_{k+1} = A + \epsilon E_{k+1}$. If $|\alpha(B_{k+1}) - \alpha(B_k)| \le \tau$, stop.

Algorithm 1 is guaranteed to generate a sequence of lower bounds for $\alpha_\epsilon(A)$. A detailed analysis of sufficient conditions for these lower bounds to converge to $\alpha_\epsilon(A)$ is given in [2].

One approach to computing $d_{\rm CI}(A)$ would be to repeatedly use Algorithm 1 to compute $\alpha_{\epsilon_j}(A)$ for a sequence $\epsilon_j$, using bisection on $\epsilon$, but this would introduce a third nested iteration and also requires the assumption that each $\alpha_{\epsilon_j}(A)$ is computed accurately. Instead, we wish to compute $d_{\rm CI}$ more directly by generating a sequence $B_k = A + \epsilon_k E_k$ with $\epsilon_k$ converging to $d_{\rm CI}(A)$. This necessitates varying $\epsilon_k$ at each step $k$ in a subtle way. We set $\epsilon_{k+1} = \epsilon_k - (y_k{}^* x_k)\alpha(B_k)$ and motivate this choice by a local analysis near the optimizer under some regularity assumptions on the abscissa. This leads to the following algorithm: