Записки научных
семинаров ПОМИ
Том 373, 2009 г.

V. Y. Pan

ROOT-SQUARING WITH DPR1 MATRICES

Abstract. Recent progress in polynomial root-finding relies on employing the associated companion and generalized companion DPR1 matrices. ("DPR1" stands for "diagonal plus rank-one.") We propose an algorithm that uses nearly linear arithmetic time to square a DPR1 matrix. Consequently the algorithm squares the roots of the associated characteristic polynomial. This incorporates the classical techniques of polynomial root-finding by means of root-squaring into new effective framework. Our approach is distinct from the earlier fast methods for squaring companion matrices.

1. Introduction

The classical root-squaring algorithm proposed about 1830 independently by Dandelin and Lobachevsky and also known as Gräffe's [6] is still an important technique for root-finding for a polynomial $p(x) = p_0(x) = \prod_{j=1}^{n}(x - \lambda_j)$ (cf., e.g., [11]). The algorithm is fundamental but easily defined. One just recursively computes the polynomials

$$p_{i+1}(x) = (-1)^n p_i(\sqrt{x}) p_i(\sqrt{-x}), \quad i = 0, 1, \dots, k-1,$$

and observes that

$$p_k(x) = \prod_{j=1}^{n}(x - \lambda_j^{2^k}).$$

With FFT one can use $O(n \log n)$ arithmetic operations per squaring. (Hereafter we refer to such operations as $\widehat{\text{op}}$s.)

Recent progress in polynomial root-finding largely relied on the transition to eigen-solving for the associated companion and generalized companion matrices, in particular for diagonal plus rank-one matrices (hereafter referred to as *DPR1 matrices*) [8, 9, 5, 1, 3, 2, 13]. The algorithms benefit from performing them entirely in terms of matrices, without back

Key words and phrases. polynomial root-squaring, DPR1 matrices.

Supported by PSC CUNY Awards 61406-0039 and 62230-0040.

and forth transition to the coefficients of the polynomial. Indeed such transitions could easily lead to numerical stability problems.

Squaring matrices squares their eigenvalues, in our case the roots of the associated polynomial. Effective root-finders based on squaring companion matrices in $O(n \log n)$ ops have been proposed in [4] and [12]. In this note, we propose a distinct recipe for squaring DPR1 matrices, also in nearly linear time.

Hereafter M^T and M^H denote the transpose and the Hermitian (that is complex conjugate) transpose of a matrix M, respectively, ($M^H = M^T$ if M is a real matrix). $\det(M)$ denotes its determinant. (A, B) denotes the 1×2 block matrix with the blocks A and B. I_r denotes the $r \times r$ identity matrix.

2. DPR1 generalized companion matrices: definition

With a polynomial $p(x) = \sum_{i=0}^{n} p_i x^i = p_n \prod_{j=1}^{n} (x - \lambda_j)$, $p_n \neq 0$, we associate its DPR1 generalized companion matrix as follows,

$$C = C_{\mathbf{s},\mathbf{u},\mathbf{v}} = D_{\mathbf{s}} - \mathbf{u}\mathbf{v}^H \tag{2.1}$$

for $\mathbf{s} = (s_i)_{i=1}^n$, $\mathbf{u} = (u_i)_{i=1}^n$, $\mathbf{v} = (v_i)_{i=1}^n$,

$$D_{\mathbf{s}} = \operatorname{diag}(s_i)_{i=1}^n, \tag{2.2}$$

$$d_i = u_i v_i = \frac{p(s_i)}{q_i(s_i)}, \quad q_i(x) = \prod_{j \neq i} (x - s_i), \quad i = 1, \ldots, n, \tag{2.3}$$

$$q_i(s_i) = q'(s_i), \quad i = 1, \ldots, n, \quad q(x) = \prod_{j=1}^{n} (x - s_i). \tag{2.4}$$

Theorem 2.1. *The DPR1 matrix C in Eq.* (2.1) *has the eigenvalues* $\lambda_1, \ldots, \lambda_n$.

Proof. See, e.g., [5] or [1, Theorem 4.4]. □

Unlike the companion matrices, DPR1 matrices are defined by the values of the associated polynomial on a fixed set of points, rather than by the coefficients.

We refer to [3] on the association of this matrix with an important secular equation.

3. The Sherman–Morrison–Woodbury formula for determinants

We need the following Sherman–Morrison–Woodbury formula for determinants,

$$\det(C - UV^H) = (\det C)\det(I_r - V^H C^{-1} U), \tag{3.1}$$

which holds provided that C is an $n \times n$ nonsingular matrix and U and V are $n \times r$ matrices (cf., e.g., [7]).

4. Squaring DPR1 matrices

Next we employ the DPR1 matrix structure to obtain fast squaring algorithm, although, unlike the algorithms for companion matrices in [4] and [12], in this case squaring is indirect and does not preserve the eigenvectors.

Theorem 4.1. (a) *Given three vectors* $\mathbf{s} = (s_i)_{i=1}^n$, $\mathbf{u}$, *and* $\mathbf{v}$ *defining the* DPR1 *matrix* C *in Eq.* (2.1) *and* n *scalars* $\mu_1, \dots, \mu_n$ *such that* $s_i^2 = \mu_h$ *for none pair* $\{i, h\}$, *we can compute the values* $\det(C^2 - \mu_h I)$ *for* $h = 1, \dots, n$ *by using* $O(n \log^2 n)$ *ops.*

(b) *Furthermore* $O(n \log n)$ *ops are sufficient if* $s_i = a\omega_n^{i-1}$ *and* $\mu_h = b\omega_k^{h-1}$ *for* $h, i = 1, \dots, n$ *and two nonzero scalars* a *and* b, *where* ω_q *denotes a primitive* q*-th root of unity and* $k = O(n)$.

Proof. Represent C^2 as $D + UV^H$ where $U = (u_{ik})_{i=1,k=0}^{n,1}$ and $V = (v_{ij})_{i=1,j=0}^{n,1}$ are $n \times 2$ matrices and $D = \mathrm{diag}\,(s_i^2)_{i=1}^n$. Equation (3.1) implies that $\det(C^2 - \mu_j I) = (\det(D^2 - \mu_h I))\det(I_2 - V^H(D^2 - \mu_h I)^{-1}U)$. The values $\det(D^2 - \mu_h I) = \prod_{j=1}^n (s_j^2 - \mu_h)$ are the values of the polynomial $\prod_{j=1}^n (s_j^2 - x)$ at the n points $x = \mu_h$, $h = 1, \dots, n$ and thus can be computed within the claimed cost bounds [10, Section 3.1]. It remains to compute the 2×2 matrices $V^H(D^2 - \mu_h I)^{-1}U = (\sum_{j=1}^n \frac{v_{ij} u_{jk}}{s_j^2 - \mu_h})_{i,k=0}^{1,1}$ for $h = 1, \dots, n$, and then (in $3n$ ops) the n values of their determinants. In $4n$ ops we compute the $4n$ products $v_{ij} u_{jk}$ for all i, j, and k. The computation of the (i, k)th entry of all the n matrices amounts to multiplication of the Cauchy matrix $(\frac{1}{s_j^2 - \mu_h})_{h,j=1}^n$ by the vector $(v_{ij} u_{jk})_{j=1}^n$. This can be done within the required bounds of $O(n \log^2 n)$ or, in case (b), $O(n \log n)$ ops (see, e.g., [10, Section 3.6]). $\square$

The theorem bounds the cost of computing the values of the characteristic polynomial $\det(C^2 - \mu_h I)$ of the matrix C^2 at n points $\mu_1, \ldots, \mu_n$. Within the same cost bound we can compute the coefficients of the polynomial $q(x) = \prod_{h=1}^{n}(x - \mu_h)$ and the values $q'(\mu_h)$ for all h (cf. [10, Section 3.1]), thus defining a DPR1 matrix whose eigenvalues $\lambda_1^2, \ldots, \lambda_n^2$ are shared with the matrix C^2.

In many applications we can define DPR1 matrices by choosing the values s_i and μ_h to our advantage, and the theorem shows some benefits in choosing $s_i = a\omega_n^{i-1}$, $\mu_h = b\omega_k^{h-1}$, $i, h = 1, 2, \ldots, n$, which allows squaring in $O(n \log n)$ ops.

Finally we note that the ith squaring step of a DPR1 matrix $C_0 = C - \mu I = D + \mathbf{u}\mathbf{v}_n^T$ can be performed in at most $2^{2i+1}n$ multiplications and $4^i(2n-1)$ additions for $i \leqslant \log_2 n$. Furthermore in this case squaring preserves the eigenvectors. Indeed write $U_0 = \mathbf{u}$, $V_0 = \mathbf{v}$,

$$C_{i+1} = D^{2^{i+1}} + D^{2^i} U_i V_i^T + U_i V_i^T D^{2^i} + U_i V_i^T U_i V_i^T, \quad i = 0, 1, \ldots,$$

and represent the output matrix as $C_{i+1} = D^{2^{i+1}} + U_{i+1} V_{i+1}^T$ where $U_{i+1} = (D^{2^i} U_i, U_i)$, $V_{i+1}^T = (V_i^T, V_i^T D^{2^i} + V_i^T U_i V_i^T)$.

References

1. D. A. Bini, L. Gemignani, V. Y. Pan, *Inverse Power and Durand/Kerner Iteration for Univariate Polynomial Root-finding.* — Computers and Mathematics (with Applications) **47**, **2/3** (2004), 447–459.
2. D. A. Bini, L. Gemignani, V. Y. Pan, *Improved Initialization of the Accelerated and Robust QR-like Polynomial Root-finding.* — Electronic Transactions on Numerical Analysis **17** (2004), 195–205.
3. D. A. Bini, L. Gemignani, V. Y. Pan, *Fast and Stable QR Eigenvalue Algorithms for Generalized Companion Matrices and Secular Equation.* — Numerische Math. **3** (2005), 373–408.
4. J. P. Cardinal, *On Two Iterative Methods for Approximating the Roots of a Polynomial.* — Lect. Appl. Math. **32** (Proceedings of AMS-SIAM Summer Seminar: Mathematics of Numerical Analysis: Real Number Algorithms (J. Renegar, M. Shub, and S. Smale, editors), Park City, Utah, 1995), pp. 165–188, Amer. Math. Soc., Providence, Rhode Island (1996).
5. S. Fortune, *An Iterated Eigenvalue Algorithm for Approximating Roots of Univariate Polynomials.* — J. Symb. Computat. **33**, **5** (2002), 627–646.
6. A. S. Householder, *Dandelin, Lobatchevskii or Gräffe?.* — Amer. Math. Monthly **66** (1959), 464–466.
7. A. S. Householder, *The Theory of Matrices in Numerical Analysis.* Dover, New York, 1964.
8. F. Malek, R. Vaillancourt, *Polynomial Zerofinding Iterative Matrix Algorithms.* — Computers and Math. with Applications **29**, **1** (1995), 1–13.

9. F. Malek, R. Vaillancourt, *A Composite Polynomial Zerofinding Matrix Algorithm.* — Computers and Math. with Applications **30**, **2** (1995), 37–47.
10. V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms.* — Birkhäuser/Springer, Boston/New York, 2001.
11. V. Y. Pan, *Univariate Polynomials: Nearly Optimal Algorithms for Factorization and Rootfinding.* — J. Symb. Computat. **33, 5** (2002), 701–733.
12. V. Y. Pan, *Amended DSeSC Power Method for Polynomial Root-finding.* — Computers and Math. with Applications **49, 9–10** (2005), 1515–1524.
13. V. Y. Pan, D. Ivolgin, B. Murphy, R. E. Rosholt, Y. Tang, X. Wang, X. Yan, *Rootfinding with Eigen-solving.* Chapter 14, pp. 219–245 in *Symbolic-Numeric Computation*, (Dongming Wang and Li-Hong Zhi, editors), Birkhäuser, Basel/Boston, 2007.

Department of Mathematics
and Computer Science Lehman
College of the City University
of New York Bronx, USA

E-mail: victor.pan@lehman.cuny.edu

Поступило 11 сентября 2009 г.