COMPLEX AND REAL POLYNOMIAL ROOT APPROXIMATION VIA DOMINANT EIGENSPACES

IVAN RETAMOSO

iretamoso@bmcc.cuny.edu

DEPARTMENT OF MATHEMATICS

BOROUGH OF MANHATTAN COMMUNITY COLLEGE

APPLICATIONS

POLYNOMIAL ROOT –FINDING HAS MODERN APPLICATIONS IN:

- *COMPUTER ALGEBRA*
- *CONTROL THEORY*
- *SIGNAL PROCESSING*
- *GEOMETRIC MODELING*
- *FINANCIAL CALCULATIONS (Internal Rate of Return)*

Another application arises in certain financial calculations, for example, to compute the rate of return on an investment where a company buys a machine for, (say) \$100,000. Assume that they rent it out for 12 months at \$5000/month, and for a further 12 months at \$4000/month. It is predicted that the machine will be worth \$25,000 at the end of this period. The solution goes as follows: the present value of \$1 received n months from now is $\frac{1}{(1+i)^n}$, where i is the monthly interest rate, as yet unknown. Hence

$$100,000 = \sum_{j=1}^{12} \frac{5000}{(1+i)^j} + \sum_{j=13}^{24} \frac{4000}{(1+i)^j} + \frac{25,000}{(1+i)^{24}}$$

Hence

$$100,000(1+i)^{24} - \sum_{j=1}^{12} 5000(1+i)^{24-j} - \sum_{j=13}^{24} 4000(1+i)^{24-j} - 25,000 = 0,$$

a polynomial equation in $1+i$ of degree 24. If the term of the lease was many years, as is often the case, the degree of the polynomial could be in the hundreds.

Solution of Cubic Equation?

Found by Cardano (16th century)

Solution of Quartic Equation?

Found by Ferrari (16th century)

Solution of Quintic equation?

Never found!

Historical Background

Niels Henrik Abel

Évariste Galois

In 1824 Niels Abel showed that there existed polynomials of degree 5, whose roots could not be expressed using radicals and arithmetic operations through their coefficients. Here is an example of such polynomials:

$$x^5 - 4x - 2$$

This discovery made us aware that we are left with iterative methods for the approximation of the roots of a polynomial given its coefficients.

Basic Polynomial

$$p(x) = \sum_{i=0}^{n} p_i x^i = p_n \prod_{j=1}^{n} (x - \lambda_j).$$

$$Where \ p_n \neq 0$$

COMPANION MATRIX OF p(x)

$$C_p = \begin{pmatrix} 0 & & & & -\frac{p_0}{p_n} \\ 1 & \ddots & & & -\frac{p_1}{p_n} \\ & \ddots & \ddots & & \vdots \\ & & \ddots & 0 & -\frac{p_{n-2}}{p_n} \\ & & & 1 & -\frac{p_{n-1}}{p_n} \end{pmatrix}$$

Definition

Eigenpair of a matrix

$Given\ a\ matrix\ M\ and\ a\ vector\ v \neq 0$
$\{\lambda, v\}\ \ is\ an\ eigenpair\ of\ \ M\ \ if\ \ Mv = \lambda v$

$\lambda\ is\ called\ an\ eigenvalue\ of\ M.$
$v\ is\ called\ an\ eigenvector\ of\ \lambda\ .$
$\Lambda(M)\ is\ the\ set\ of\ all\ eigenvalues\ of\ M,$
$called\ the\ spectrum\ of\ M.$

Theorem

Given $p(x)$ *and its companion matrix* C_p.

λ *is an eigenvalue of* $C_p \iff \lambda I - C_p$ *is singular*

$$\iff \det(\lambda I - C_p) = 0 \iff p(\lambda) = 0$$

Thus, we deduce this simple fact:

The roots of a polynomial are precisely the eigenvalues of its associated companion matrix.

POLYNOMIAL ROOT-FINDING BY MATRIX ALGORITHMS

WE APPROXIMATE THE ROOTS OF A POLYNOMIAL
$p(x) = p_n \prod_{j=1}^{n}(x - \lambda_j) = p_0 + p_1 x + \ldots + p_{n-1}x^{n-1} + p_n x^n$

AS THE EIGENVALUES OF THE ASSOCIATED COMPANION

MATRIX $C_p = \begin{pmatrix} 0 & \ldots & \ldots & \ldots & \ldots & -p_0/p_n \\ 1 & 0 & \ldots & \ldots & & -p_1/p_n \\ & 1 & \ldots & \ldots & \ldots & -p_2/p_n \\ & & \ldots & \ldots & \ldots & \ldots \\ & & \ldots & \ldots & \ldots & \ldots \\ & & & 1 & 0 & -p_{n-2}/p_n \\ & & & & 1 & -p_{n-1}/p_n \end{pmatrix}$

TOEPLITZ MATRIX

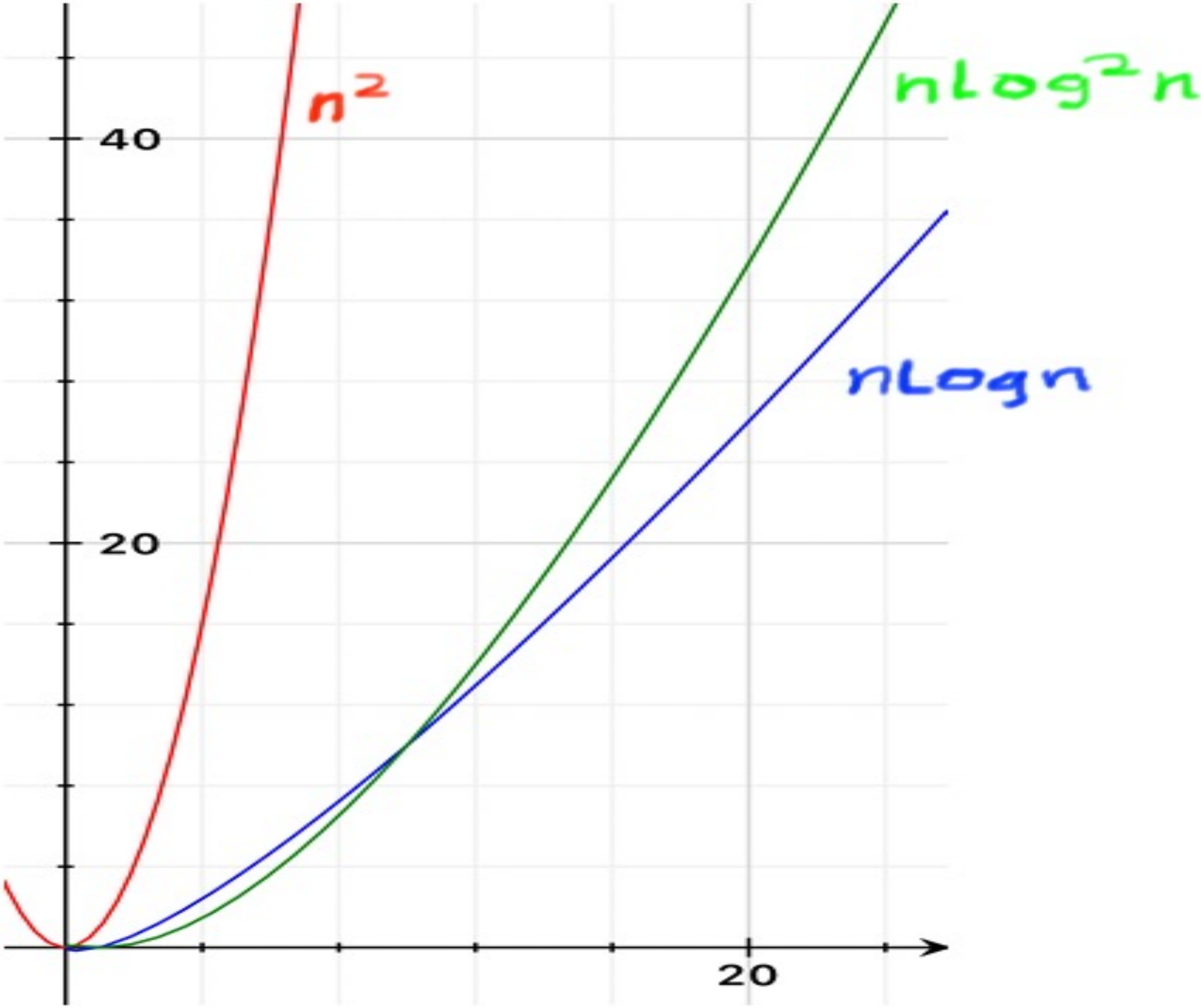
$$T = (t_{i-j})_{i,j=1}^{n,n} = \begin{pmatrix} t_0 & t_{-1} & \dots & t_{1-n} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{pmatrix}$$

THE MATRIX C_p GENERATES AN ALGEBRA OF $n \times n$ MATRICES WITH STRUCTURE OF TOEPLITZ TYPE.

WE NEED $O(n \log n)$ AR. OPS TO MULTIPLY THEM PAIRWISE OR BY A TOEPLITZ MATRIX, AND $O(n \log^2 n)$ TO INVERT THEM (IF NONSINGULAR).

$O(n \log^2 n)$ COVER THE COST OF AN ITERATION.

Comparing Complexities



There are many methods for the approximation of the eigenvalues of a matrix. They can be found in:

- *Golub, MATRIX COMPUTATION*
- *Stewart, MATRIX ALGORITHM Vol. 2*
- *Watkins, FUNDAMENTALS OF MATRIX COMPUTATIONS*

THE POWER METHOD

Let $A \epsilon \mathbb{C}^{n \times n}$ be a matrix with n simple and distinct eigenvalues.
$(\lambda_1, v_1), (\lambda_2, v_2), \dots, (\lambda_n, v_n)$ *are n eigenpairs of A.*
$|\lambda_1| > |\lambda_i|$ *for all $i \neq 1$*

Choose a random vector v.
Then

$$v = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n \quad , c_1 \neq 0 \; with \; probability \; 1$$

$$Av = c_1 A v_1 + c_2 A v_2 + \cdots + c_n A v_n$$

$$A^k v = c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + \cdots + c_n \lambda_n^k v_n$$

$A^k v \approx c_1 \lambda_1^k v_1$ *for sufficiently large k*

$$A^k v = w$$

$\lambda_1 = \dfrac{w^T A w}{w^T w}$ *(Rayleigh quotient)*

Definition of an eigenspace

$\mathcal{S} \subseteq \mathbb{C}^n$ is an *invariant subspace* or *eigenspace* of a matrix $M \in \mathbb{C}^{n \times n}$ if $M\mathbf{v} \in \mathcal{S}$ for all $\mathbf{v} \in \mathcal{S}$.

An example of an eigenspace is the space spanned by an eigenvector or a set of eigenvectors.

Generalization of Power Method

MORE GENERALLY (E.G., FOR CLUSTERS OF EIGENVALUES), APPROXIMATE AN EIGENSPACE OF EIGENVECTORS OF THE MATRIX C_p AND THEN RECOVER THE ASSOCIATED SET OF EIGENVALUES OF C_p.

Basic Theorem 1

Stewart 2001, Vol. 2, [Theorem 4.1.2].
For all matrix bases $U \in \mathbb{C}^{n \times r}$ of an eigenspace $\mathcal{U}$ of $M \in \mathbb{C}^{n \times n}$ we have $MU = UL$ for unique matrix $L = U^{(I)} MU$.

Where $U^{(I)}$ is a left inverse of U i.e., $U^{(I)} U = I$ and L shares all its eigenvalues with M.

DOMINANT EIGENSPACE

Consider $\mathcal{U}$ as in our *Basic Theorem 1,* let $\Lambda(L)$ be the set of eigenvalues of L, and let $\Lambda(M)$ be the set of eigenvalues of M

$\mathcal{U}$ is a Dominant Eigenspace if :

$$|\lambda| \gg |\mu|$$

as long as $\lambda \in \Lambda(L)$ and $\mu \in \left(\Lambda(M) - \Lambda(L)\right)$

Method to approximate a Dominant Eigenspace

LET $\mathcal{U}$ BE A DOMINANT EIGENSPACE OF M, $\dim(\mathcal{U}) = r$.

CHOOSE A STANDARD GAUSSIAN RANDOM $n \times r$ MATRIX G.

THEN RANGE (COLUMN SPAN) OF $U = MG$ APPROXIMATES $\mathcal{U}$ WITH PROBABILITY ≈ 1.

A matrix M may have no Dominant Eigenspace but $f(M)$ does for appropriate $f(x)$.

So next we define matrix functions.

Matrix Functions

Given $f(z)$ a scalar rational function we naturally define $f(M)$ a matrix function by substituting M for z, replacing division by matrix inversion and replacing 1 by the identity matrix.

Example:

$$f(z) = \frac{1+z^2}{1-z} \quad \Longrightarrow \quad f(M) = (I + M^2)(I - M)^{-1}$$

Basic Theorem 2

$Let\ f\ be\ a\ function\ defined\ on\ \Lambda(M).$

$$If\ \ Mv = \lambda v,\ \ then\ \ f(M)v = f(\lambda)v$$

MATRIX FUNCTIONS PRESERVE EIGENSPACES OF M

REPEATED SQUARING WITH SCALING

$$F_0 = M, \qquad F_{h+1} = \frac{F_h^2}{\left\| F_h^2 \right\|} \quad for\ \ h = 0,1, \ldots$$

Every squaring of a matrix squares its eigenvalues. This strengthens the domination of the dominant eigenvalues.

Basic Steps

The eigenvalues of C_p will be approximated in 5 steps.

1. Apply an appropriate function f to the matrix C_p to generate a dominant eigenspace of $f(C_p)$.

2. Compute matrix basis U for the dominant eigenspace of $f(C_p)$.

3. Compute the matrix L as:
   $L = U^{(I)} C_p U$.

4. Approximate eigenvalues of L, which L shares with C_p.

5. Deflate C_p to find more eigenvalues and repeat steps 1-5 until we find all desired eigenvalues.

NUMERICAL TEST FOR REPEATED SQUARING (CODES)

```
clearvars
clc
Degree=64
p=rand(1,Degree+1);
C=compan(p);
M=C;
for i=1:10
    M=(M^2)/norm(M^2,2);
    temp=rank(M);
    variation_of_rank(1,i)=temp;
end
variation_of_rank
G=randn(Degree,rank(M));
U=M*G;
L=pinv(U)*C*U;
eigenvalues_of_L=sort(eig(L),'descend')
eigenvalues_of_C=sort(eig(C),'descend')
```

NUMERICAL TEST FOR REPEATED SQUARING

```
Degree =

    64

variation_of_numerical rank =

    64    64    64    64    64     8     2     2     2     2

eigenvalues_of_L =

   0.2054 + 1.6678i
   0.2054 - 1.6678i

eigenvalues_of_C =

   0.2054 + 1.6678i
   0.2054 - 1.6678i
  -0.3737 + 1.0324i
  -0.3737 - 1.0324i
  -1.0523 + 0.2863i
             .
             .
             .
```

Numerical Test
Table for Repeated squaring

n	numerical rank/squarings	min	max	mean	std
64	numerical rank	1	10	5.31	2.79
128	numerical rank	1	10	3.69	2.51
256	numerical rank	1	10	4.25	2.67
64	squarings	6	10	7.33	0.83
128	squarings	5	10	7.37	1.16
256	squarings	5	11	7.13	1.17

Approximating Real roots of a polynomial

MOTIVATION FOR APPROXIMATING REAL ROOTS OF A POLYNOMIAL

IN VARIOUS APPLICATIONS, E.G., TO OPTIMIZATION OF COMPUTATIONS IN ALGEBRAIC GEOMETRY, ONLY THE r REAL ROOTS ARE OF INTEREST, AND TYPICALLY THEY ARE MUCH LESS NUMEROUS THAN ALL n ROOTS.

Main idea

Are the r real eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_r$ of C_p dominant?
NO!
Find a function f such that the eigenvalues
$f(\lambda_1), f(\lambda_2), \dots, f(\lambda_r)$ of $f(C_p)$ are dominant.

Matrix Method to Approximate Real Roots of a polynomial

Separate Real from Nonreal roots. use a Matrix Cayley map.

Cayley maps

for complex numbers:

$w: z \longmapsto \frac{z-\sqrt{-1}}{z+\sqrt{-1}}$

as a matrix function:

$W: X \longmapsto (X - I\sqrt{-1})(X + I\sqrt{-1})^{-1}$

Let $X = C_p$

The map above transforms: "the real axis into the unit circle". Hence the real eigenvalues of the input matrix will be mapped into the unit circle and the other comple: eigenvalues will be mapped either inside or outside of the unit circle.

Now let $M = (X - I\sqrt{-1})(X + I\sqrt{-1})^{-1}$

and let's apply repeated squaring to the following two matrices:

Apply repeated squaring to the following two matrices:

$M_0 = M$

and

$\widehat{M_0} = M^{-1}$,

obtaning the sequences

M_h

and

$\widehat{M_h}$

for $h = 1,2,3, \ldots$

Then for large integers h the images of the real eigenvalues of the input matrix (for us the companion matrix) will strongly dominate all other eigenvalues of the matrix:

$$S = \left(M_h + \widehat{M_h}\right)^{-1}$$

Test for approximating real roots

Let

$$p(x) = (7x^6 + 6x^5 + 5x^4 + 4x^3 + 3x^2 + 2x + 1)(2x^2 - 1)$$

$$p(x) = 14\,x^8 + 12x^7 + 3x^6 + 2x^5 + x^4 - x^2\ - 2x - 1$$

After a Cayley map and 5 repeated squarings we output:

eigenvalues_of_L =

0.7047
-0.6696

eigenvalues_of_C =

0.4107 + 0.6399i
0.4107 - 0.6399i
-0.2051 + 0.6838i
-0.2051 - 0.6838i
0.7071 + 0.0000i
-0.7071 + 0.0000i
-0.6341 + 0.2877i
-0.6341 - 0.2877i

Table for Cayley map and Root-squaring Algorithm

Degree(n)	Number of real roots (r)	Number of Iterations	Error Bound
16	4	5	1.22E-15
16	6	4	2.00E-15
16	8	4	8.10E-15
32	4	6	2.26E-13
32	6	6	9.88E-14
32	8	6	2.35E-13

3.2 “Matrix free” algorithm to approximate real roots

Cayley Map and Root-Squaring

Cayley Maps

The map

$$y = \frac{x + \sqrt{-1}}{x - \sqrt{-1}}$$

and its inverse

$$x = \sqrt{-1}\left(\frac{y + 1}{y - 1}\right)$$

send the real line onto the unit circle and vice versa.

Cayley Map and Root-Squaring Algorithm

INPUT: *two integers n and r, $0 < r < n$, and the coefficients of a polynomial $p(x)$ where $p(0) \neq 0$ and $p(1)p(\sqrt{-1}) \neq 0$.*

OUTPUT: *Approximations of the real roots $x_1, ..., x_r$ of the polynomial $p(x)$.*

COMPUTATIONS:

1. *Compute* $q(x) = (x-1)^n p(\sqrt{-1}\ \frac{x+1}{x-1}) = \sum_{i=0}^{n} q_i x^i$. *(This Cayley map moves the real axis, in particular the real roots of* $p(x)$*, into the unit circle.)*

2. $q_0(x) = q(x)/q_n$*, apply the* k *squaring steps via* $q_{h+1}(x) = (-1)^n q_h(\sqrt{x}) q_h(-\sqrt{x})$ *for* $h = 0, 1, \ldots, k-1$*. then divide* $q_k(x)$ *by* $\|q_k(x)\|$ *(These steps keep the images of the real roots of* $p(x)$ *on the init circle for all* k*, while sending the images of every other root of* $p(x)$ *towards either the origin or the infinity.)*

COMPUTATIONS:

3. *For a sufficiently large integer* k*, the polynomial* $q_k(x)$ *approximates the polynomial* $x^s u_k(x)$ *where* $u_k(x) = \sum_{i=0}^{r} u_i x^i$ *and has all roots lying on the unit circle. Obtain* $u_k(x)$.

4. *Compute the polynomial* $w_k(x) = u_k(\frac{x+\sqrt{-1}}{x-\sqrt{-1}})$. *(This Cayley map sends the images of the real roots of the polynomial* $p(x)$ *lying on the unit circle* $C(0,1)$ *back to the real line.)*

COMPUTATIONS:

5. *Apply one of the algorithms of [BT90], [BP98], and [DJLZ97] to approximate the* r *real roots* $z_1, \ldots, z_r$ *of the polynomial* $w_k(x)$ *(cf. Theorem 3.1.4).*

6. *Apply the Cayley map* $w_j^{(k)} = (z_j + \sqrt{-1})/(z_j - \sqrt{-1})$ *for* $j = 1, \ldots, r$ *to extend Stage 5 to approximating the* r *roots* $x_1^{(k)}, \ldots, x_r^{(k)}$ *of the polynomials* $u_k(x)$ *and* $y_k(x) = x^s u_k(x)$ *lying on the unit circle* $C(0, 1)$.

COMPUTATIONS:

7. *Apply the descending process (similar to the ones of [P95] and [P02])) to approximate the r roots $x_1^{(h)}, \ldots, x_r^{(h)}$ of the polynomials $q_h(x)$ lying on the unit circle $C(0, 1)$ for $h = k - 1, \ldots, 0$.*

8. *Approximate the r real roots $x_j = \sqrt{-1}(x_j^{(0)} + 1)/(x_j^{(0)} - 1)$, $j = 1, \ldots, r$, of the polynomial $p(x)$.*

The overall cost of this algorithm is $O(kn \log n)$ flops.

NUMERICAL TEST FOR CAYLEY MAP AND ROOT-SQUARING ALGORITHM

The test candidates are products of rth Chebyshev polynomials and polynomials of the form $1+2x+3x^2 \ldots + (n-r+1)x^{n-r}$. The number of real roots of such polynomials equals to the degree r of the Chebyshev polynomial. The iteration stops when there are only $r+1$ nonzero coefficients that have absolute value greater than the tolerance bound 10^{-5}. The descending procedure was achieved by a Proximity Test with Newton's Iteration.

Additional topics

Matrix version of Cayley Map Algorithm

$$H(M) = (M + \sqrt{-1}I)(M - \sqrt{-1}I)^{-1}$$

Let P=H(M)

$$P^k - P^{-k} = (P^{2k} - 1)P^{-k} = (\prod_{i=0}^{k-1}(P^2 - \omega_k^i))(P^{-k}) = \prod_{i=0}^{k-1}(P - \omega_k^i P^{-1})$$

Real eigen-solving by means of factorization.

INPUT: *a real $n \times n$ matrix M having r real eigenvalues and $s = (n-r)/2$ pairs of nonreal complex conjugate eigenvalues, neither of them equal to $\sqrt{-1}$.*

OUTPUT: *approximations to the real eigenvalues $x_1, \ldots, x_r$ of the matrix M.*

COMPUTATIONS:

1. *Compute the matrix* $P = (M + \sqrt{-1}\ I)(M - \sqrt{-1}\ I)^{-1}$. *(This is the matrix version of a Cayley map of Theorem 3.1.3. It moves the real and only the real eigenvalues of the matrix* M *into the the eigenvalues of the matrix* P *lying on the unit circle* $C(0,1)$.*)*

COMPUTATIONS:

2. *Fix a sufficiently large integer* k *and compute the matrix* $Y = (P^k - P^{-k})^{-1}$ *in the following factorized form* $\prod_{i=0}^{k-1}(P - \omega_k^i P^{-1})^{-1}$ *where* $\omega_k = \exp(2\pi\sqrt{-1}/k)$. *(For any integer* k *the images of all real eigenvalues of the matrix* M *have absolute values at least 2, whereas the images of all nonreal eigenvalues of that matrix converge to 0 as* $k \to \infty$.*)*

COMPUTATIONS:

Complete the computations following "Basic Steps" starting at step 2.

The arithmetic complexity of this algorithm is $O(kn)$, which makes the algorithm attractive for real polynomial root-finding as long as it converges for a reasonably small integer k.

BASIC POLYNOMIAL p(x)

$$p(x) = \sum_{i=0}^{n} p_i x^i = p_n \prod_{j=1}^{n} (x - \lambda_j).$$

Basic Theorem 1

Stewart 2001, Vol. 2, [Theorem 4.1.2].
For all matrix bases $U \in \mathbb{C}^{n \times r}$ of an eigenspace $\mathcal{U}$ of $M \in \mathbb{C}^{n \times n}$ we have $MU = UL$ for unique matrix $L = U^{(I)} MU$.

THE SECOND BASIC THEOREM. The Eigenproblems for a Matrix and Its Function.

Let a rational function $f(\lambda)$ be defined on the eigenvalues of M.

Let $M\mathbf{v} = \lambda\mathbf{v}$. Let $f(M)\mathcal{U} = \mathcal{U} = \mathcal{U}(\mu)$.

Then $M\mathcal{U} = \mathcal{U} = \mathcal{U}(\Lambda)$, Λ is the set of the eigenvalues λ of M such that $f(\lambda) = \mu$.

Λ is a singleton if μ is a simple eigenvalue of $f(M)$.

$\Longrightarrow$ WE CAN RECOVER AN EIGENSPACE OF M FROM ITS IMAGE IN $f(M)$.