A Fast Algorithm for the Eigenvalues

Computation of a Toeplitz Matrix

P. Sotiriadis
National Technical Univ. of Athens
Electrical and Computer Eng. Dept.
Polytechnion Zografou,
15773, Athens, Greece.

Prof. Nikos E. Mastorakis,
Hellenic Naval Academy,
Chair of Computer Science,
Terma Hatzikyriakou, 18539,
Piraeus, GREECE.
email: mastor@softlab.ntua.gr
Fax: +301 4181768

Prof. G. Carayiannis
Institute for Language Processing
Margari 22, Psychiko, Athens, Greece.

Abstract: This paper is concerned with the
computation of the eigenvalues of a real,
symmetric Toeplitz matrix. A very efficient
algorithm is described which is based on the
division method and on the Durbin algorithm.

Keywords: Algorithms theory, Eigenvalues
computation, applied matrix theory.

I. INTRODUCTION

The problem of determining the eigenvalues and
eigenvectors of a given matrix is, in general,
one of exceptional difficulty. A large number
of various algorithms, each applicable to a
particular class of matrices, have been
developed for this purpose. All these
algorithms are based on the iteration of
various procedures. Excellent surveys of the
various algorithms can be found in [1] and [2].

A new idea for the computation of the minimum
eigenvalue $\lambda_{\min}$ of a positively definite
Toeplitz matrix $R$ is the following [3], [4].

Consider the matrix $R - \lambda I$. We have $\lambda < \lambda_{\min}(R)$,
if and only if $R - \lambda I > 0$, as well as, $\lambda > \lambda_{\min}(R)$,
if and only if $R - \lambda I$ has at least one negative
eigenvalue. Therefore, if one could find a
criterion for the positivity of $R - \lambda I > 0$, one
could find a criterion for checking if
$\lambda < \lambda_{\min}(R)$. If the given matrix is Toeplitz,
the criterion in question is the Durbin
algorithm.

Suppose that $[A_0, B_0]$ is an a-priori known
interval in which $\lambda_{\min}$ is lying, so $\lambda_{\min} \in
[A_0, B_0]$. The value of $\lambda_{\min}$ can be approximated
by the value of $\lambda$ using the following algorithm.
Step 1: $A := A_0$, $B := B_0$
Step 2: $\lambda := \frac{A \lambda B}{2}$
Step 3: If $R \cdot \lambda I > 0$ Then $A := \lambda$, else $B := \lambda$
Step 4: If $|A - B| > 2\varepsilon$
    (where $\varepsilon$ is the absolute error),
    Then go to Step 2
Step 5: $\lambda := \frac{A \lambda B}{2}$
Step 6: End

Note that after each iteration of steps 1 and 2, the interval which contains $\lambda_{\text{min}}$ is subdivided and a final error estimation after the $k$-th iteration is:

$$|\lambda - \lambda_{\text{min}}| \leq \frac{B_0 - A_0}{2^k}$$

The main advantage of the present algorithm is its sureconvergence and the absolute bound of the final error.

The paper is organised as follows. In Section II, some basic theoretical results from Matrix Theory are presented. In Section III, we combine these results in order to produce the proposed algorithm in its formal form. In Section IV, the computational complexity of the algorithm is given. Finally, there exists a conclusion.

II. PRELIMINARY RESULTS

In this Section, some important Theorems from Matrix Theory are given. First, a definition is stated.

Definition 1: Leading k×k principal submatrix of a n×n matrix $R$ is the k×k $R_{i\bar{j}}$, i,j=1...k.

The following proposition is proved in [5] (pp. 103-104).

Proposition 1 (Interlacing Property)
Let $R_{k+1}$ be the leading k×k principal submatrix of the n×n symmetric matrix $R$, then k+1×k+1 one has:

$$\lambda_{k+1}(R_{k+1}) \geq \lambda_k(R_{k+1}) \geq \lambda_k(R_{k}) \geq \lambda_{k+1}(R_k)$$

where $\lambda_1(R_{k+1}), \lambda_2(R_{k+1}), \ldots, \lambda_k(R_{k+1})$

and

$$\lambda_1(R_{k+1}) \leq \lambda_2(R_{k+1}) \leq \ldots \leq \lambda_k(R_{k+1}) \leq \lambda_{k+1}(R_{k+1})$$

are the eigenvalues of the matrices $R_{k+1}$ and $R_k$ respectively. It is assumed that

$$\lambda_1(R_{k}) \leq \lambda_2(R_{k}) \leq \ldots \leq \lambda_k(R_k) \leq \lambda_{k+1}(R_{k+1})$$

Then

$$\lambda_{k+1}(R_{k+1}) \leq \lambda_k(R_{k+1}) \leq \lambda_k(R_k) \leq \lambda_{k+1}(R_{k+1})$$

Proof: See in [5], pp. 103-104.

Theorem 1
Let $R$ be a symmetric n×n matrix and $R_k$ be its leading k×k principal submatrix for k=1...n and $d_k = \det(R_k)$. If $d_k = 0$ for all k=1...n, then the matrix $R$ appears to have m-negative eigenvalues and (n-m) positive ones, where m is the number of sign alterations of the sequence $d_1, d_2, \ldots, d_n$.

Proof: Suppose that p of the k eigenvalues $\lambda_i$ are negative, that is

$$\lambda_{k-p+1}(R_k) < \lambda_{k-p+1}(R_k) < \ldots < \lambda_{k-p+1}(R_k)$$

Then $\text{sign}(\det(R_k)) = (-1)^p$.

According to Proposition 1, p eigenvalues of $R_{k+1}$ are negative and (k-p) are positive. For the $\lambda_{k-p+1}(R_{k+1})$, it holds that:

$$\lambda_{k-p+1}(R_{k+1}) \leq \lambda_{k-p+1}(R_{k+1}) \leq \lambda_{k-p+1}(R_{k+1})$$

$\lambda_{k-p+1}(R_{k+1}) = 0$ (according to our assumption).

One can verify that $\text{sign}(\det(R_{k+1})) = (-1)^p \text{sign}(\lambda_{k-p+1}(R_{k+1}))$, or $\text{sign}(\det(R_{k+1})) = 0$.

Therefore if there exists a sign alteration from $d_k$ to $d_{k+1}$, then $R_{k+1}$ has one more negative eigenvalue than $R_k$. If $d_k$ and $d_{k+1}$ are of the same sign, then $R_k$ and $R_{k+1}$ have the same number of negative eigenvalues. The proof is completed by induction.
In the case that the matrix $R$ is a symmetric Toeplitz matrix, Theorem 1 can be applied directly to the results found by the Durbin algorithm. Suppose now that $R=(r_{i-j})_{i,j=0}^{n-1}$.

From Durbin's algorithm ([1]) one obtains $a_{i}^{n-1}$, where $R_{i}$ is the leading principal submatrix of $R$ [1]. Therefore $R_{i}=(r_{j})_{j=0}^{n-1}$ and $\det(R_{i})=a_{i-1}a_{i-2}...a_{0}$ for $i=1,...,n$. Now, we are interested in the number of sign alternation in the sequence $a_{0},a_{1},...,a_{i}$: This number is equal to the number of negative terms in the sequence $a_{0},a_{1},...,a_{i}$ (Note that from the assumption of the first theorem $a_{n}=0$).

Definition: Let $R$ be a symmetric $n\times n$ Toeplitz matrix. We define the function $D(R)$ as follows: $D(R)=\sum(\text{the number of negative terms in the sequence } a_{0}, a_{1},..., a_{i-1})$, where $a_{n}=0,...,n-1$. result by the application of Durbin's Algorithm on $R$.

For the Durbin's algorithm, it is well known that a sign alternation exists from $a_{0}$ to $a_{n}$ if and only if $|a_{i}|>0$. For this reason, we can also define $D(R)$ as:

$D(R) = r_{n}^{n-1} \sum \{ |1-k_{i}| < 0 \}$, where the operator $(x<0)$ is defined as follows:

$(x<0) = 1$ if $x<0$

$(x<0) = 0$ if $x=0$

One can easily seen, based on Theorem 1 and the above remarks that $D(R)=D(R)$. In the following Proposition a new assisting parameter $\lambda$ is introduced.

Proposition 2
Let $R$ be a symmetric $n\times n$ Toeplitz matrix and $\lambda_{1}(R)=A_{1}(R), \lambda_{2}(R)=A_{2}(R),...,$ $\lambda_{n}(R)$. Then $\lambda=\lambda_{n}(R)$ if and only if $\forall \epsilon > 0$, $D(R,\lambda_{n}(R)+\epsilon) > n-m$ and $D(R,\lambda_{n}(R)-\epsilon) < n-m$.

Proof: Since $R$ is symmetric, there exists an orthonormal matrix $Q$ such that $R=QQ^{T}$. It can be assumed, without loss of generality, that $Q=(\lambda_{1}(R),\lambda_{2}(R),...,$ $\lambda_{n}(R))$.

From the relation $v=Qv_Q^{T}$ one takes

$\lambda \neq v(v^{T}) = Q \lambda_{1}(R) Q^{T} + \lambda \lambda_{1}(R) - \lambda^{T} Q^{T} v$;

$\lambda_{m}(R)-v, \lambda_{m}(R)-v, \lambda_{m}(R)-v, \lambda_{m}(R)+v)^{T}$.

For $\forall \lambda=\lambda_{m}$ we have

$\lambda_{m}(R)-v<0, \lambda_{m}(R)-v<0$

or equivalently $D(R,\lambda_{m}) > n-m$

for $\forall \lambda=\lambda_{m}$ we have

$\lambda_{m}(R)-v<0, \lambda_{m}(R)-v<0$

or equivalently $D(R,\lambda_{m}) < n-m$.

In order to exploit Proposition 2, it now remains to estimate an initial interval containing the eigenvalues of a Toeplitz matrix. For this purpose, the following Theorem is proved in [6] (p.371).

Theorem 2: (Gershgorin Circle Theorem)
The eigenvalues $\lambda_{1}(R), \lambda_{2}(R),...,$ $\lambda_{n}(R)$ of the matrix $R$ fulfill the relation

$\lambda_{i}(R) \in \mathbb{C}^{n\times n}$

$\{ \mathbb{C} : |z-r_{i,j}| \leq \sum_{j=1}^{n} |r_{i,j}| \}$

In the case where $R$ is a symmetric Toeplitz matrix, Theorem 2 gives

$A_{0} = \sum_{i=1}^{n} |r_{i,j}| \lambda_{i}(R) \lambda_{i+j}(R) < \sum_{i=1}^{n} |r_{i,j}| B_{i}$

Now, having the initial interval $[A_{0},B_{0})$, and using the Proposition 2 one can formulate the following algorithm for finding the eigenvalues of the symmetric Toeplitz matrix.
III. THE ALGORITHM

In this section, the proposed algorithm is stated:

Algorithm:
Given the real symmetric $n \times n$ Toeplitz Matrix $R = (R_{i,j})_{1 \leq i,j \leq n}$ or equivalent its elements, $r_0, r_1, \ldots, r_{n-1}$ and the small positive constant $\delta$ (for example $\delta = 0.001$), find the $k$-th eigenvalue of $R$, $\lambda_k(R)$, with respect to the order of the eigenvalues $\lambda_1(R), \lambda_2(R), \ldots, \lambda_n(R)$.

Error: If $\lambda$ is an estimation of $\lambda_k(R)$, $\epsilon = |\lambda - \lambda_k(R)|$ the absolute error, we demand $\epsilon \leq \epsilon^*$, where $\epsilon^*$ is the defined bound of $\epsilon$.

Step 1: $A_0 := r_0 - 2|\Sigma r_k|_{k=1}^{n-1}$, $B_0 := r_0 + 2|\Sigma r_k|_{k=1}^{n-1}$

Step 2: $A := A_0$, $B := B_0$

Step 3: $S := \left[ \log_2 \left( \frac{B - A}{\epsilon^*} \right) - 1 \right]$

Step 4: $\lambda := (1-d)A + (1-d)B$

Step 5: For $j = 1$ to $S$

Step 6: If $DN(R - \lambda I) = 0$ Then $A := \lambda$, else $B := \lambda$

Step 7: $A := \frac{A + B}{2}$

End

Remark 1: The symbol $[n]$ is used for the upper integer part of a number $n$. $S$ is the number of recursions in the for-loop and results from the bound $\epsilon^*$. One can easily verifies that $\epsilon \leq \epsilon^* = \frac{B - A}{2^{\epsilon^*}}$

Remark 2: The small constant $\delta$ is essential: If we set $S = 0$, then the first value of $L$ would be $L = r_0$ and the matrix $(R - \lambda I)$ would have its first leading subdeterminant equal to zero. In that case, an error “division by zero” would appeared in the call of the function $DN$. This constant increase the error bound $\epsilon$ by the negligible quantity $\delta/\epsilon^*$.

IV. COMPUTATIONAL MULTIPLEXITY OF THE ALGORITHM

The Durbin recursion appears approximately $2n^2$ MADs (Multiplications and Divisions) [1]. The number of required calls of function $DN$ is $S$ where

$S := \left[ \log_2 \left( \frac{B - A}{\epsilon^*} \right) - 1 \right] + \log_2 \left( \frac{B - A}{\epsilon^*} \right) - \log_2 \left( \frac{4r}{\epsilon^*} \sum |r_k| \right)$

So, the total cost in MADs (Multiplications and Divisions) is $P$, where

$P = 2n^2 \log_2 \left( \frac{4}{\epsilon^*} \sum |r_k| \right)$

If the matrix $R$ is an autocorrelation matrix and the relation $r_0 = |r_k|$, $\forall k \in \{1, \ldots, n-1\}$ holds then $P = 2n^2 \left[ \log_2 (n-1) + \log_2 \left( \frac{4}{\epsilon^*} \right) \right]$.

The total multiplexity can be reduced more than 25% if one wants to compute all the eigenvalues of $R$.

V. CONCLUSION

A new fast and efficient algorithm for the computation of the eigenvalues of a symmetric, real, Toeplitz matrix is proposed. The algorithm has the following advantages:
- It converges always, since it is based on the bisection method.
- Multiple eigenvalues do not effect it.
- A bound for the absolute error is a-priori known.
- Each eigenvalue can be computed directly, without computing all the absolute greater eigenvalues (as in the method of the Powers), and so no additive computational errors accumulate.
- The proposed algorithm can be developed in a parallel version using parallel Durbin or $\Sigma$ recursions.

In Appendix, a Pascal Program is given.
APPENDIX

(const N=8: (dimension of R)
( N-1:7; (N-1:1)
(t) Type Vector=Array[0..N1] of Real;
( EigenIndex=1..N;
( VAR U:VECTOR;

Function
Topeigen(r:Vector;K:EigenIndex:e:Real):Real;
(r=r_1,r_2,..,.r_{N-1},x)
( K is the index of the eigenvalue
( e is the bound of the absolute error
)

Const D=0.001; ( D is the small constant δ
)
Var A,B,L,H:Real;
S,J:Integer;

Function DN(L:Real):integer; (the function DN).
Var Y:Vector;
K,A,r0:Real;
M,1,SUM:Integer;

Begin
r0:=r[0];
r0:=r0-L; (Durbin's algorithm)
K:=r[I]/r[0];
A:=r[0];
If A<0 then SUM:=SUM+1;
Y[I]:=K;
For M:=1 to N-2 do
Begin
a:=(1-K*K)*a;
If A<0 then SUM:=SUM+1;
K:=r[M+1];
For J:=1 to M do
K:=K+r[M-J]*Y[J];
K:=K/A;
For J:=1 to M do
Z[J]:=Y[J]+K*Y[M+1-J];
For J:=1 to M do
Y[J]:=Z[J];
Y[M+1]:=K;
End;

a:=(1-K*K)*a;
If A<0 then SUM:=SUM+1;
r0:=r0;
DN:=SUM;
End:

Begin (main part of the algorithm)
H:=0;
For J:=1 to N-1 do
H:=H+Abs(r[J]);
A:=(r[0]-2)*H;
B:=(r[0]+2)*H;
S:=Round(Exp(0.5));
L:=(1+D)*A+(1-D)*B/2;
For J:=1 to S do
Begin
If DN(L)<N-K then A:=L Else B:=L;
End:
Topeigen:=L;
End: (Topeigen)

BEGIN
U[0]:=1;
U[1]:=-50;
U[2]:=0;
U[3]:=1;
U[4]:=7;
U[5]:=43;
U[6]:=9;
U[7]:=0;
WRITELN:
WRITELN(TOPEIGEN(U,8.1e-10):4:10):

(R=(u_{i-j})_{j=1..8}) (5th
eigenvalue)
(e=1E-10)
END.

VI. REFERENCES