Institute of Computer Science
Academy of Sciences of the Czech Republic

Verified Eigendecomposition

Jiří Rohn

Technical report No. V-1143

02.01.2012

Pod Vodárenskou věží 2, 182 07 Prague 8, phone: +420 266 051 111, fax: +420 286 585 789,
e-mail:rohn@cs.cas.cz

Institute of Computer Science
Academy of Sciences of the Czech Republic

Verified Eigendecomposition

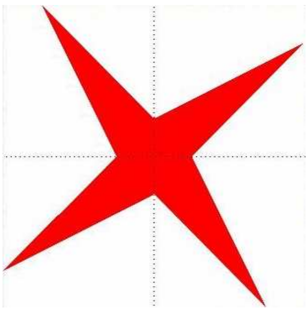
Jiří Rohn[1]

Technical report No. V-1143

02.01.2012

Abstract:

We disclose the file ols.m whose p-coded version is a part of the open source verification software package VERSOFT for computing verified eigenvalues and eigenvectors of a complex (or real) matrix.



Keywords:
Eigenvalue, eigenvector, verified result, interval arithmetic.[2]

[1]This work was supported by the Institutional Research Plan AV0Z10300504.

[2]Above: logo of interval computations and related areas (depiction of the solution set of the system $[2,4]x_1 + [-2,1]x_2 = [-2,2]$, $[-1,2]x_1 + [2,4]x_2 = [-2,2]$ (Barth and Nuding [1])).

1 Introduction

VERSOFT [2], a freely available verification software package written in INTLAB [3], a toolbox of MATLAB, contains as one of its pillars a p-coded (content-obscured) function `ol.p` for computing verified eigendecomposition of a complex (or real) matrix. We make it here publicly available as a function `ols.m` (OL Shortened) in a compact form consisting of only 23 lines of the source code. The original function has been stripped off the output error variable `E` only, the rest has been kept intact.

2 Description and examples

Here is the `help` of the function (not present in the compact source code):

```
function [L,X]=ols(A)
%    OLS         Verified eigenvalues and eigenvectors of a complex (or real) matrix.
%
%    This is an INTLAB file. It requires to have INTLAB installed under
%    MATLAB to function properly.
%
%    For a square complex (or real) matrix A,
%        [L,X]=ols(A)
%    computes (generally complex) interval matrices L and X, L diagonal,
%    that are verified to contain matrices Lo, Xo satisfying
%        A*Xo=Xo*Lo
%    in exact arithmetic, where diag(Lo) is the vector of ALL eigenvalues of A
%    and Xo is a matrix of corresponding eigenvectors; L, X are enclosures
%    of these quantities. Multiple eigenvalues are taken into account.
%
%    The vector
%        lam=diag(L)
%    has the following additional property: for each i, j, the intervals
%    lam(i) and lam(j) are either identical, or disjoint. Thus, if all of
%    them are disjoint, then each of them contains exactly one eigenvalue
%    of A.
%
%    If A is real and symmetric, then L, X are real. If A is Hermitian, then L is real.
%    In these cases both lam.inf and lam.sup are ordered in nondecreasing order.
%
%    EXAMPLE 1 (multiple eigenvalues). The following matrix has a six-tuple
%    eigenvalue 2 corresponding to three Jordan blocks of sizes 1, 2 and 3:
%    A =
%       -60     1    42    -3   -10     4
%       133     0   -92     6    23    -8
%      -186     3   128    -9   -30    12
%       252    -4  -171    14    41   -16
%      -310     5   210   -15   -48    20
```

```
%       372    -6  -252    18    60   -22
%    >>  [L,X]=ols(A); format long, lam=diag(L)
%    intval lam =
%    [   1.99980487061652 -  0.00019256911560i,   2.00019000884616 +  0.00019256911404i]
%    [   1.99980487061652 -  0.00019256911560i,   2.00019000884616 +  0.00019256911404i]
%    [   1.99980487061652 -  0.00019256911560i,   2.00019000884616 +  0.00019256911404i]
%    [   1.99980487061652 -  0.00019256911560i,   2.00019000884616 +  0.00019256911404i]
%    [   1.99980487061652 -  0.00019256911560i,   2.00019000884616 +  0.00019256911404i]
%    [   1.99980487061652 -  0.00019256911560i,   2.00019000884616 +  0.00019256911404i]
%
%    Enclosures of equal eigenvalues are equal, as explained above. Low
%    accuracy is caused by multiplicity; the imaginary parts cannot be
%    filtered out by the program here.
%
%    EXAMPLE 2 (symmetric matrix; Hilbert 4x4). Output for a real symmetric
%    matrix is always real:
%    >> A=hilb(4), [L,X]=ols(A); format long, lam=diag(L), X
%    A =
%      Columns 1 through 3
%       1.000000000000000   0.500000000000000   0.333333333333333
%       0.500000000000000   0.333333333333333   0.250000000000000
%       0.333333333333333   0.250000000000000   0.200000000000000
%       0.250000000000000   0.200000000000000   0.166666666666667
%      Column 4
%       0.250000000000000
%       0.200000000000000
%       0.166666666666667
%       0.142857142857143
%    intval lam =
%    [   0.00009670230402,   0.00009670230403]
%    [   0.00673827360576,   0.00673827360577]
%    [   0.16914122022144,   0.16914122022146]
%    [   1.50021428005924,   1.50021428005925]
%    intval X =
%      Columns 1 through 2
%    [   0.02919332316478,   0.02919332316479] [   0.17918629053545,   0.17918629053546]
%    [  -0.32871205576320,  -0.32871205576317] [  -0.74191779062846,  -0.74191779062845]
%    [   0.79141114583312,   0.79141114583313] [   0.10022813694718,   0.10022813694721]
%    [  -0.51455274999717,  -0.51455274999714] [   0.63828252819360,   0.63828252819363]
%      Columns 3 through 4
%    [  -0.58207569949724,  -0.58207569949723] [   0.79260829116376,   0.79260829116377]
%    [   0.37050218506709,   0.37050218506710] [   0.45192312090159,   0.45192312090160]
%    [   0.50957863450179,   0.50957863450180] [   0.32241639858182,   0.32241639858183]
%    [   0.51404827222216,   0.51404827222217] [   0.25216116968824,   0.25216116968825]
%    Observe the high accuracy of the result.
%
```

```
%    See also EIG, VERIFYEIG.
%
%    Copyright 2008-2011 Jiri Rohn.
%
%    Employs the routine VERIFYEIG by Siegfried M. Rump.
```

3 Download

The source file can be downloaded from
http://uivtx.cs.cas.cz/~rohn/matlab/others/ols.m

Dedication

Dedicated to O. L. after whom the file was named. ◻

Bibliography

[1] W. Barth and E. Nuding, *Optimale Lösung von Intervallgleichungssystemen*, Computing, 12 (1974), pp. 117–125. 1

[2] J. Rohn, *VERSOFT: Verification software in MATLAB/INTLAB*, 2009. http://uivtx.cs.cas.cz/~rohn/matlab. 2

[3] S. Rump, *INTLAB - INTerval LABoratory*, in Developments in Reliable Computing, T. Csendes, ed., Kluwer Academic Publishers, Dordrecht, 1999, pp. 77–104. http://www.ti3.tu-harburg.de/rump/. 2