Electronic Journal of Differential Equations, Vol. **1994**(1994), No. 06, pp. 1–10.
ISSN: 1072-6691. URL: http://ejde.math.swt.edu or http://ejde.math.swt.edu
ftp (login: ftp) 147.26.103.110 or 129.120.3.113

Appendix to: Computing Eigenvalues of Regular Sturm-Liouville Problems *

H. I. Dwyer & A. Zettl

Abstract

This appendix contains a remark about the scalar problem. It also shows a FORTRAN implementation of the algorithm discussed in the main article.

1 Scalar Problems

If the original problem to be solved is scalar, that is, the coefficients are 'matrices' of dimension 1, then the transformed, separated problem will be of dimension 2. In this case, it is possible to simplify the algorithm for solving separated matrix problems considerably, by taking advantage of the fact that there are simple formulas for the inverse and eigenvalues of 2x2 matrices. This allows the references to routines in a linear algebra library (such as NAG or IMSL) to be replaced by simple computational routines. Several other simplifications are also possible:

1. It becomes relatively simple to verify that the boundary conditions specified by the user are correct; the requirements for a self-adjoint problem as specified in Theorem 1 reduce to four simple tests.
2. The initial value for the Theta matrix is easily computed 'by-hand'.
3. One of the difficulties encountered in the early attempts to implement the algorithm for separated matrix problems was caused by the use of a library routine to compute eigenvalues. The algorithm tracks changes in the eigenvalues, and this was made somewhat more difficult because the library routine did not return the list of eigenvalues in any predictable order. Thus, it was necessary to inspect the list, and 'unshuffle' them if the order had been changed. Since the eigenvalues are now computed by formulas, rather by a library routine, they will always be listed in the same order, and the 'unshuffle' step becomes unnecessary.

* Attached on January 9, 1996 to the article published on August 23, 1994.

2 Sample FORTRAN Implementation

A simple FORTRAN implementation of the algorithm, as applied to scalar problems, is provided below. The user is required to edit the program and define the coefficient functions for the original, coupled problem. The code must then be compiled before it can be run. The coefficient functions in this sample listing correspond to Problem b. The results using this implementation agree very well with the results obtained using the more general version of the code.

Note: This particular version uses a basic Runge-Kutta method for the numerical integration. More advanced methods could certainly be used, but this simple approach has worked well enough during the initial testing.

The FORTRAN program can be found under the name “Dwyer.fort” in the directory EJDE/Volumes/1994/06-Dwyer-Zettl/