ROMAI J., **2**, 1(2006),
203–208

LISC – A MATLAB PACKAGE FOR LINEAR DIFFERENTIAL PROBLEMS

Damian Trif
Babes-Bolyai University of Cluj-Napoca
dtrif@math.ubbcluj.ro

Abstract A MATLAB package, based on the Lanczos tau method, useful in the *Lyapunov-Schmidt* (LS) method for nonlinear second order differential boundary value problems of the type $Lu = Nu$ is presented. Future applications to evolution problems or bifurcation studies are also taken into account.

1. INTRODUCTION

The Lyapunov-Schmidt (LS) method, elaborated in the years 1906-1908 and reformulated in a modern mathematical language by L. Cesari [3] after 1963 applies to some nonlinear equations of the type $Lu = Nu$, in the presence of boundary conditions, considered on the domain of the linear operator L.

This method could be easily extended to the case of a nonlinear evolution equation on a Hilbert space H (usually a L^2 space) of the form $\frac{du}{dt} = F(u) \equiv Lu+Nu$, where the domain of F is dense in H. We can also use the Lyapunov-Schmidt method to discuss the bifurcation problem $x - \lambda Ax + G(x, \lambda) = 0$, $x \in X$, where X is a Banach space, $A : X \to X$ is a linear compact operator and $G : X \times \mathbb{R} \to \mathrm{X}$ is a continuous mapping such that $G(x, \lambda) = o(\|x\|)$ for all $\lambda \in \mathbb{R}$.

Although it has been used for a long time only for the theoretical demonstration of the existence and branching of the solutions of such problems, the LS method (or, by Cesari, the *alternative method*) is also very useful to the effective approximation of these solutions.

We will present LISC, a MATLAB package based on the Lanczos-tau method, for Sturm-Liouville problems with applications to the Lyapunov-Schmidt method for nonlinear equations. The advantage of the LS method consists of the important reduction of the dimension of the nonlinear system to be solved together with the possibility to oversee the approximating errors. This advantage occurring in some examples, proves that the LS method behaves better than other known methods such as `bvp4c` or `sbvp`.

2. THE LS METHOD

We assume that the linear part L of the equation $Lu = Nu$ is a Sturm-Liouville operator

$$Ly \equiv \frac{1}{r(x)}\left[-(p(x)y')' + q(x)y\right], \quad x \in [a,b]$$
$$y(a)\cos\alpha + (py')(a)\sin\alpha = 0,$$
$$y(b)\cos\beta + (py')(b)\sin\beta = 0,$$

where $1/p$, q, r are real-valued functions on $[a,b]$, $p(x) > 0$, $r(x) > 0$ on $[a,b]$, $p \in C^1[a,b]$, $q, r \in C[a,b]$. It is well known that the eigenvalues of L form an increasing sequence $\lambda_0 < \lambda_1 < ...$ converging to infinity and the corresponding eigenfunctions φ_n form an orthogonal (orthonormal) basis of the Hilbert space $L^2_r(a,b)$. The asymptotic behaviour of the eigenvalues is $\lambda_n \in O(n^2)$.

A theoretical constructive variant of the LS method can be found in [7], [8]. Summarizing, the approximating algorithm is:

a) we are looking for an approximate solution of the equation $Lu = Nu$ of the form

$$u = \sum_{k=1}^{m} c_k\varphi_k + \sum_{k=m+1}^{N} c_k\varphi_k,$$

where $0 \le m \le N$;

b) by fixing $u^* = \sum_{k=1}^{m} c_k\varphi_k$, we generate the associate function $y(u^*)$ performing the iterations

$$y^0 = u^*,\ y^{s+1} = u^* + H_m N y^s = u^* + \sum_{k=m+1}^{N} C_k^s\varphi_k,\ s = 0,1,...,S;$$

c) with $y = y^{S+1}$ as an approximation of the associated function, we can write the system $Lu^* = P_m N y$ of the determining equations, with the unknowns $c_1, ..., c_m$. This system of the form $F(c_1, ..., c_m) = 0$ is then numerically solved, by a suitable method, for instance by the Newton's method. Every evaluation of the function F means the reiteration of the b) step. Finally, thus determined u^* generates, also by the b) iterations, an approximation of the solution of the equation $Lu = Nu$.

We remark that in the case of the *Galerkin's method,* the approximating solutions are being looked for under the form $u^* = \sum_{k=1}^{N} c_k\varphi_k$, where the coefficients $c_k, k = 1, ..., N$ are determined from the equations $(Lu^* - Nu^*, \varphi_k) = 0, k = 1, ..., N$ i.e.

$$(\lambda_k u^* - Nu^*, \varphi_k) = 0, k = 1, ..., N.$$

These equations are got from the determining equations for $m = N$. If $m = 0$ the system of the determining equations disappears. The function associated

with a certain u^* satisfies the equation $y = L^{-1}Ny$. Therefore, in this case, the algorithm is reduced to the transformation of the equation $Lu = Nu$ into a fixed point problem. Obviously, this case arises only when the inverse L^{-1} exists and $L^{-1}N$ is a contraction.

The first version of our package applies only to the Sturm-Liouville case for the linear operator L, in the form

$$Lu = \frac{1}{r(x)}\left[\frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + g(x)u\right],$$
$$au'(0) + bu(0) = 0, \quad cu'(1) + du(1) = 0.$$

There exists a Matlab package `MATSLISE` of V. Ledoux (2004), based on the works of L. Ixaru which uses the so called CP methods to calculate the eigenfunctions of Sturm-Liouville or Schrödinger operators but this package works slowly. A more interesting package is `MWRtools` of R. A. Adomaitis (1998-2001) [1] which uses spectral methods to calculate the eigenfunctions of the Sturm-Liouville operator in order to solve some *linear* boundary value problems.

In the next section we propose a Chebyshev-tau method to solve the Sturm-Liouville problem in order to get a good basis φ_i, and the corresponding Matlab package.

3. CHEBYSHEV-TAU METHOD

3.1. THE PROBLEM

The module `Chebyshev` of our package solves problems of the type

$$p_2(x)u'' + p_1(x)u' + p_0(x)u = g(x), \quad x \in (a,b) \tag{1}$$

$$\alpha_{11}u(x_{11}) + \alpha_{12}u'(x_{12}) = \beta_1,$$
$$\alpha_{21}u(x_{12}) + \alpha_{22}u'(x_{22}) = \beta_2 \tag{2}$$

using the Chebyshev-tau method.

For the moment let us suppose that $a = -1$, $b = 1$. A powerful method to solve (1) is to express u as a Chebyshev series $u(x) = c_0\frac{T_0(x)}{2} + c_1T_1(x) + ...$, where $T_i(x) = \cos(i\cos^{-1}(x))$ is the standard Chebyshev polynomial of order i [5], [4].

For the practical implementation, we define the vectors c and t by $c^T = (c_0, c_1, c_2, ...)$, $t^T = (\frac{T_0}{2}, T_1, T_2, ...)$ so that $u(x) = c^Tt = t^Tc$. By using the properties of Chebyshev polynomials we have $x \cdot T_i = \frac{T_{i-1}}{2} + \frac{T_{i+1}}{2}$, hence, $x \cdot u(x) = c^TX^Tt = (Xc)^Tt$, where $X_{0,1} = 1$, $X_{ii} = 0$, $X_{i,i-1} = X_{i,i+1} = 1/2$.

Then, in general, $x^m u(x) = (X^m c)^T t$ and $f(x)u(x) = (f(X)c)^T t$ for analytical functions f. Moreover, $\dfrac{u(x)}{x^m} = \left(X^{-m}c\right)^T t$ if the l.h.s. has no singularity at the origin. Of course, X is a tri-diagonal matrix, X^2 is a penta-diagonal matrix and so on but, in general, the matrix version `funm(X)`of the scalar function $f(x)$ or $X^{-m} = [inv(X)]^m$ are no longer sparse matrices.

Similarly, let D be the differentiation matrix defining $\dfrac{d^m u}{dx^m} = (D^m c)^T t$. The matrix D is a upper triangular and its entries are

$$D_{ii} = 0,\ D_{ij} = 0 \text{ for } (j-i) \text{ even and } D_{ij} = 2j \text{ otherwise.}$$

Applying these formulae to equation (1), we get

$$(p_2(X)D^2 + p_1(X)D + p_0(X))c = g,$$

where G are the coefficients of the r.h.s. function

$$g(x) = g_0 \frac{T_0(x)}{2} + g_1 T_1(x) + ...$$

The condition (2) can be formulated in a similar manner. We define $t_{ij} = t^T(x_{ij})$ so that it can be written in the form $h_i^T c = \beta_i, i = 1, 2$, where

$$h_i^T = \sum_{j=1}^{2} \alpha_{ij} t_{ij}^T D^{j-1}, i = 1, 2.$$

Define the matrices $A = \sum_{i=0}^{2} P_i(X)D^i$ and $H = (h_1, h_2)^T$. Then the vector c satisfies

$$\begin{pmatrix} H \\ A \end{pmatrix} c = \begin{pmatrix} \beta \\ q \end{pmatrix} \tag{3}$$

of the form $\mathcal{A}c = b$, where $\beta = (\beta_1, \beta_2)^T$.

Of course, in reality we cannot work with infinite matrices but only with finite portions $(N \times N)$ of them.

If instead of $[-1, 1]$ we have another interval $[a, b]$ for x, we use the change of coordinates $x = \alpha\xi + \beta$ where $\alpha = \dfrac{b-a}{2}$ and $\beta = \dfrac{b+a}{2}$, so that $\xi \in [-1, 1]$. We must change X to $\alpha X + \beta I$ and D to D/α.

4. EXAMPLES FOR LS METHOD

The package LISC is under construction. Here we present some test problems for nonlinear equations, using 50 grid points and 21 eigenfunctions. The tutorials and the m-files of the package can be obtained by e-mail from the author (`dtrif@math.ubbcluj.ro`) or from Matlab Central - File Exchange.

For the moment, it is based on a Sturm-Liouville solver LiScEig for a bounded interval (for the φ_i basis) and the files `as.m` and `lisc.m` which implement the LS method.

Example 1. Solve the boundary value problem

$$x'' + x = 0.5t - 0.5x^3,$$
$$x(0) = 0,$$
$$x(1) + x'(1) = 0.$$

The eigenvalues of the linear part are $\lambda_i = 1 - l_i^2$, where $\sin l_i = -l_i \cos l_i$, $\varphi_i = c_i \sin(l_i t)$. The errors between our solution and the solution calculated by `bvp4c` of Matlab is less than 14×10^{-6}.

Example 2. Consider the two-point problem for the Emden equation (with a singularity)

$$y'' + \frac{2}{x}y' + y^5 = 0,$$
$$y'(0) = 0, \quad y(1) = \frac{\sqrt{3}}{2}.$$

The general Sturm-Liouville operator now has the form $\frac{1}{x^2}\left(x^2 u'\right)' = \lambda u$. The closed-form solution reads $y(x) = \left(1 + \frac{x^2}{3}\right)^{-\frac{1}{2}}$ and the error is of order $1.e{-}5$.

Example 3. If comes from the combustion theory [6],

$$-u'' + Mu' = DY_o Y_f e^{-\frac{\theta}{T_0 + u}},$$
$$Mu(0) - u'(0) = 0, u(1) = 0,$$

where $Y_o = -u + Y_{o1} e^{M(x-1)}$, $Y_f = -u + 1 - e^{M\,(x-1)}$, $T_0 = 0.118$, $Y_{o1} = 0.21$, $\theta = 2.6$. The computed solution for $M = 6$ and $D = 5.5 \times 10^8$ (with a high gradient at the burning front) agrees well with those of [6].

5. CONCLUSIONS

The comparison between `LISC` and `SBVP 1.0` of Auzinger [2] and `bvp4c` of Matlab (see Matlab help) shows a computing time for LISC about 1 -2 times larger. The Matlab profile reports show that about 75% of the computing time was spent on computation of the eigenfunctions and only about 6% on the effective computation of the numerical solution. But we have good reasons to use LS method.

We can build a database with known eigenfunctions. In the problems with parameters, where we have (for example) bifurcations, or in evolution problems, we can use repeatedly the same eigenfunctions. The eigenfunctions carry physical information, so that our LS solution has a better structure for studies. LS method could be easily extended to 2D or 3D (evolution) problems, with non-invertible linear part. In all the cases, we finally have to solve a very small nonlinear system, usually with $m = 0, 1, 2$ values, which also carry information about bifurcation behaviour.

References

[1] Adomaitis, R. A., Lin Yi-hung, *A collocation/quadrature based Sturm-Liouville problem solver*, ISR Technical Research Report TR 99-01.

[2] Auzinger, W., Kneisl, G., Koch, O., Weinmuller E. B., *SBVP 1.0 A Matlab solver for singular boundary value problems*, Technische Universität Wien, ANUM preprint no. 02/02.

[3] Cesari L., *Functional analysis and Galerkin's method*, Mich. Math. J., **11**, 3(1964), 383-414.

[4] Liefvendahl, M., *A Chebyshev tau spectral method for the calculation of eigenvalues and pseudospectra* (`http://www.nada.kth.se/~mli/research.html`)

[5] MacLeod, A. J., *An automatic matrix approach to the Chebyshev series solution of differential equations* (`http://maths.paisley.ac.uk/allanm/amcltech.htm`)

[6] Rameau, J. F., Schmidt-Laine, C., *Perturbed bifurcation phenomenon in a diffusion flame problem*, Appl. Numer. Math., **5**,3(1989), 237-255.

[7] Petrila T., Trif D., *Basics of fluid mechanics and introduction to computational fluid dynamics*, Springer, Berlin, 2005.

[8] Trif, D., *The Lyapunov-Schmidt method for two-point boundary value problems*, Fixed Point Theory, **6**,1(2005), 119-132.