BULETINUL ACADEMIEI DE ŞTIINŢE
A REPUBLICII MOLDOVA. MATEMATICA
Number 3(55), 2007, Pages 23–34
ISSN 1024–7696

LiScNLE – a Matlab package for some nonlinear partial differential evolution equations

Titus Petrila, Damian Trif

Abstract. We will present a MATLAB package for nonlinear evolution equations, based on the *Lyapunov-Schmidt* (LS) method. The eigenfunctions basis of the linear part is used to represent the solution at every time level (or for every value of the parameters in the case of bifurcation analysis). These eigenfunctions are calculated in a preprocessing stage or are given by the user.

Mathematics subject classification: 34B15, 49M15, 65M70.
Keywords and phrases: Lyapunov-Schmidt method, evolution equations, MATLAB package.

1 Introduction

Much of the work of an engineer or scientist is that of formulating suitable mathematical models for a particular physical system. For a dynamical system in continuous time, the model is often some system of ordinary differential equations or partial differential equations. When formulating such models, one of the goals is to maximize qualitative correctness in representing the dynamics of the physical system. However, in many cases, correctly representing the dynamics is not the sole objective in formulating mathematical models. In particular, the model needs to be useful for its intended application. For example, if a model is required in some kind of real-time feedback control scheme, then a model that is computationally intensive may be unsuitable for this purpose. We may wish to sacrifice some of the correctness of the model in order to make the equations easier to solve or to allow faster computation of the trajectories. In other words, given a model of a dynamical system that is known to correctly represent the system dynamics, how do we formulate a model of reduced complexity which retains as much of the original predictive capability as possible?

Many of the mathematicians of the twentieth century devoted their efforts to studying boundary value problems for *linear* differential equations. However, in many cases problems arising in biology, mechanics, chemistry, may be seen as nonlinear perturbations of linear ones. All these can be represented in the abstract form $Lu = Nu$ where $L : X \to Y$ (linear) and $N : Y \to Y$ (nonlinear) are suitable operators between Banach spaces X, Y where $X \subset Y$ compactly.

When L is invertible, $Lu = Nu$ can be rewritten as a fixed point equation $u = [L^{-1}N]u$. In case $L^{-1} : Y \to X$ and $N : Y \to Y$ are continuous and carry bounded sets into bounded sets, $L^{-1}N : Y \to Y$ is completely continuous. Thus, the Schauder's Fixed Point Theorem (which extends the well-known Brower's Fixed Point Theorem to completely continuous operators on infinite dimensional Banach spaces) could be used in the treatment of such problems.

Schauder's Fixed Point Theorem had little impact outside the scope of nonlinear perturbations of *invertible* operators. Often we must treat some problems where the equation is a nonlinear perturbation of a linear operator with *nontrivial* kernel (problems at resonance). A useful tool for studying such type of problems is the *Lyapunov-Schmidt reduction method.*

The Lyapunov-Schmidt (LS) method, elaborated in the years 1906-1908 and reformulated in a modern mathematical language by L. Cesari [1] 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.

As a simple example (following [2]), let us consider the problem

$$\begin{gathered} -u'' - \alpha u' - \lambda_1(\alpha)u + g(u) = 0, \quad t \in [0,\pi]\,, \\ u(0) = u(\pi) = 0 \end{gathered} \tag{1}$$

where α is a given real number, $\lambda_1(\alpha) = 1 + \alpha^2/4$ is the first eigenvalue of the linear problem

$$\begin{gathered} -u''(t) - \alpha u'(t) = \lambda u(t), \quad t \in [0,\pi]\,, \\ u(0) = u(\pi) = 0 \end{gathered}$$

and g is a continuous and $T-$ periodic function with zero mean.

In order to apply the Lyapunov-Schmidt reduction method, we consider the linear differential operator

$$L : W_0^{2,1}(0,\pi) \to L^1(0,\pi), \quad Lu = -u'' - \alpha u' - \lambda_1(\alpha)u$$

and the Nemytskii operator

$$N : W_0^{2,1}(0,\pi) \to L^1(0,\pi), \quad Nu(t) = -g(u(t)), \quad \forall t \in [0,\pi]$$

so that (1) is equivalent to the operator equation $Lu = Nu$.

It is well known that L is a linear Fredholm operator of zero index, $\ker L = sp(\varphi)$, $im(L) = \psi^{\perp}$, where

$$\varphi(t) = \frac{e^{-\frac{\alpha}{2}t}\sin t}{\sqrt{\int_0^{\pi}\left(e^{-\frac{\alpha}{2}s}\sin s\right)^2}}, \quad \psi(t) = \frac{e^{\frac{\alpha}{2}t}\sin t}{\sqrt{\int_0^{\pi}\left(e^{\frac{\alpha}{2}s}\sin s\right)^2}}\,, \quad t \in [0,\pi].$$

The splitting $W_0^{2,1}(0,\pi) = sp(\varphi) \oplus \varphi^{\perp}$ leads us to rewrite any element $u \in W_0^{2,1}(0,\pi)$ as $u = \widetilde{u} + \overline{u}\varphi$, where $\overline{u} \in \mathbb{R}$ and $\widetilde{u} \in \varphi^{\perp}$ and to observe that $L : \varphi^{\perp} \to \psi^{\perp}$

is a topological isomorphism. Let us denote $K : \psi^{\perp} \to \varphi^{\perp}$ the inverse of this isomorphism and define the projection

$$Q : L^1(0,\pi) \to L^1(0,\pi), \quad h \mapsto \left(\int_0^{\pi} h(s)\psi(s)ds \right) \psi.$$

This way, equation $Lu = Nu$ becomes equivalent to the Lyapunov-Schmidt system

$$\widetilde{u} = K(I-Q)N(\widetilde{u} + \overline{u}\varphi), \tag{2}$$

$$\int_0^{\pi} g(\widetilde{u}(s) + \overline{u}\varphi(s))\psi(s)ds = 0. \tag{3}$$

From the *auxiliary equation* (2) we observe that, being N bounded and K compact, the Schauder Fixed Point Theorem implies the existence, for any $\overline{u} \in \mathbb{R}$, of the fixed point $\widetilde{u}(\overline{u}) \in \varphi^{\perp}$. Consequently, the *bifurcation equation* (3) becomes an equation for $\overline{u} \in \mathbb{R}$,

$$\int_0^{\pi} g(\widetilde{u}\left(\overline{u}\right)(s) + \overline{u}\varphi(s))\psi(s)ds = 0.$$

and the solvability of $Lu = Nu$ comes from the solvability of this one-dimensional equation.

This method could be easily extended to the case of a nonlinear evolution equation on a Hilbert space H (usually an L^2 space) of the form $\dfrac{du}{dt} = F(u) \equiv Lu + Nu$ where the domain of F is dense in H. We assume that $\{\varphi_i, i = 0, 1, \dots\}$ forms a complete orthogonal basis for H (for example the eigenfunctions of L).

Fix $m \in \mathbb{N}$ and let $P \equiv P_m : H \to X_m \equiv X$ be the orthogonal projection from H onto the finite dimensional subspace spanned by $\{\varphi_1, \dots, \varphi_m\}$. Let $Q \equiv Q_m = (I - P) : H \to Y \equiv Y_m$ be the complementary orthogonal projection.

Given $u \in H$, let $Pu = p$ and $Qu = q$. The equation can be rewritten as

$$\frac{dp}{dt} = PF(p,q), \tag{4}$$

$$\frac{dq}{dt} = QF(p,q). \tag{5}$$

The strategy is fairly simple: study the dynamics of the low dimensional Galerkin projection (4) (where $q = q(p)$ from (5)) to draw conclusions about the dynamics of the given equation.

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 the *alternative method*, following Cesari) is also very useful for the effective approximation of these solutions.

We will present *LiScNLE*, a MATLAB package for dynamical systems, based on the LS method. The eigenfunctions basis of the linear part L of the system is used to represent the solution at every time level, or for every value of the parameters in the case of bifurcation analysis. These eigenfunctions are calculated in a preprocessing stage [3] or are given by the user. Also, other functions could be used as basis. The package extends a preliminary steady version [4].

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 can be remarked in some examples, which prove that the LS method behaves better than other known methods, such as `bvp4c` or `sbvp`.

The first two sections present the basic theory and the implementation of LiScNLE. The last two sections present examples and conclusions.

2 The LS method

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

$$Ly \equiv \frac{1}{w(x)}\left[-(p(x)y')' + q(x)y\right], \quad x \in [a,b],$$

$$y(a)\cos\alpha + (py')(a)\sin\alpha = 0, \quad y(b)\cos\beta + (py')(b)\sin\beta = 0$$

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

A theoretical but constructive variant of the LS method could be found in [5, 6]. We are looking for an approximate solution of the equation $Lu = Nu$ of the form $u = \sum_{i=1}^{N} c_i\varphi_i$ (eigenfunction expansion) which leads to the following equation for the unknowns c_i

$$\sum_{i=1}^{N} c_i L\varphi_i = N\left(\sum_{i=1}^{N} c_i\varphi_i\right).$$

We obtain the equation

$$\sum_{i=1}^{m} c_i\lambda_i\varphi_i + \sum_{i=m+1}^{N} c_i\lambda_i\varphi_i = N\left(\sum_{i=1}^{N} c_i\varphi_i\right) \tag{6}$$

where m is a positive integer, less than N. By applying the partial inverse H_m of L,

$$H_m u = \sum_{i=m+1}^{N} \frac{c_i}{\lambda_i}\varphi_i$$

to (6), we are led to

$$\sum_{i=m+1}^{N} c_i\varphi_i = H_m N\left(\sum_{i=1}^{N} c_i\varphi_i\right) = \sum_{i=m+1}^{N} C_i\varphi_i$$

so that we have

$$c_i = C_i(c_1,\dots,c_N), i = m+1,\dots,N.$$

For a sufficiently great m we may calculate $c_{m+1},\dots,c_N$ as functions of $c_1\dots c_m$, using Banach Fixed Point Theorem.

By applying the projection P_m to (6) we obtain the determining equation

$$\sum_{i=1}^{m} c_i\lambda_i\varphi_i = P_m N\left(\sum_{i=1}^{N} c_i\varphi_i\right)$$

which is a small finite dimensional system for $c_1,\dots,c_m$.

In fact, in LS methods, the true unknowns are $c_1,\dots,c_m$; the other coefficients $c_{m+1},\dots,c_N$ are calculated as coefficients of the associated fixed point.

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

$$Lu = \frac{1}{w(x)}\left[\frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + g(x)u\right],$$

$$au'(0) + bu(0) = 0, \quad cu'(1) + du(1) = 0.$$

There exists a Matlab package `MATSLISE` of V. Ledoux (2004) [7], based on the works of L. Ixaru which uses the so called CP methods to calculate the eigenfunctions of Sturm-Liouville or Schrodinger operators but this package works slowly. A more interesting package is `MWRtools` of R.A. Adomaitis (1998–2001) [8] which uses spectral methods to calculate the eigenfunctions of the Sturm-Liouville operator in order to solve some *linear* boundary value problems.

We remark that in the case of *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 $c_k, k = 1,\dots,N$, are determined from the equations $(Lu^* - Nu^*, \varphi_k) = 0,\ k = 1,\dots,N$, i.e.

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

These equations are got from the determining equations for $m = N$. If $m = 0$ the system of the determining equations disappears. The associate function to a certain u^* verifies the equation $y = L^{-1}Ny$, so the algorithm is reduced, in this case, 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.

3 Implementation

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

Let us consider the problem

$$p_2(x)u'' + p_1(x)u' + p_0(x)u = g(x) \quad x \in (a, b)\,, \tag{7}$$

$$\begin{aligned} \alpha_{11}u(x_{11}) + \alpha_{12}u'(x_{12}) &= \beta_1, \\ \alpha_{21}u(x_{12}) + \alpha_{22}u'(x_{22}) &= \beta_2 \end{aligned} \tag{8}$$

and let us suppose for the moment $a = -1$, $b = 1$. A powerful methods to solve (7) is to express u as a Chebyshev series $u(x) = c_0\dfrac{T_0(x)}{2} + c_1T_1(x) + \dots$ where $T_i(x) = \cos(i\cos^{-1}(x))$ is the standard Chebyshev polynomial of order i. For the practical implementation, we define the vectors c and t by $c^T = (c_0, c_1, c_2, \dots)$, $t^T = \left(\dfrac{T_0}{2}, T_1, T_2, \dots\right)$ so that $u(x) = c^Tt = t^Tc$.

There exists a matrix X for which $x \cdot u(x) = (Xc)^Tt$, see [9, 10],

$$X_{0,1} = 1, X_{ii} = 0, X_{i,i-1} = X_{i,i+1} = \frac{1}{2}.$$

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, i.e.

$$f(x) = \sum_{k=0}^{\infty} f_k \frac{x^k}{k!}.$$

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, generally, 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 giving $\dfrac{d^m u}{dx^m} = (D^m c)^T t$. D is an upper triangular matrix with $D_{ii} = 0$, $D_{ij} = 0$ for $(j - i)$ even and $D_{ij} = 2j$ otherwise.

Applying these formulae to equation (7), 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 $g(x)$.

$$g(x) = g_0\frac{T_0(x)}{2} + g_1T_1(x) + \dots\ .$$

The condition (8) can be formulated in a similar manner. We define

$$t_{ij} = \left(\frac{T_0(x_{ij})}{2}, T_1(x_{ij}), T_2(x_{ij}), \dots\right)^T$$

so that it can be written in the form $h_i^T c = \beta_i, i = 1, 2$, where

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

Now we 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

$$\begin{pmatrix} H \\ A \end{pmatrix} c = \begin{pmatrix} \beta \\ q \end{pmatrix} \tag{9}$$

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. For the initial conditions, we restrict t_i to have N components and use the truncation D_N instead of D, so that the computed matrix will be $\begin{pmatrix} H^* \\ A^* \end{pmatrix}$. We then take the first N rows and columns of $\begin{pmatrix} H^* \\ A^* \end{pmatrix}$ as the matrix to use, together with the first N elements of $\begin{pmatrix} \beta \\ q \end{pmatrix}$.

If we have another interval $[a, b]$ instead of $[-1, 1]$ 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.

`LiScNLE` (`Liapunov`-`Schmidt` `Non-Linear` `Evolution`) is a Matlab package for the study of some nonlinear differential evolution equations for the unknown function $u(x, t)$

$$\frac{\partial u}{\partial t} + Lu = Nu, \quad x \in (a, b), t > 0$$

where L is a Sturm-Liouville operator

$$Lu \equiv \frac{1}{w(x)} \left[-\frac{\partial}{\partial x} \left(p(x) \frac{\partial u}{\partial x} \right) + q(x) u \right]$$

and Nu is a nonlinear (differential) operator

$$Nu \equiv N(x, u, \frac{\partial u}{\partial x}).$$

We have also boundary value conditions

$$a_{11} u(a, t) + a_{12} \frac{\partial u}{\partial x}(a, t) = 0,$$

$$a_{12} u(b, t) + a_{22} \frac{\partial u}{\partial x}(b, t) = 0$$

and initial condition $u(x, 0) = u_0(x)$.

We perform a time semi-discretization by Crank-Nicolson method (for example)

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{1}{2} \left(-Lu^{n+1} + Nu^{n+1} - Lu^n + Nu^n \right)$$

i.e.

$$u^{n+1}+\frac{\Delta t}{2}Lu^{n+1}=\frac{\Delta t}{2}Nu^{n+1}+u^n-\frac{\Delta t}{2}Lu^n+\frac{\Delta t}{2}Nu^n$$

where

$$u^n=u(x,n\Delta t),\ u^0=u_0(x)$$

and Δt is the time step. For each n, this problem is of the form

$$\mathcal{L}u^{n+1}=\mathcal{N}u^{n+1}$$

where

$$\mathcal{L}u=\left(I+\frac{\Delta t}{2}L\right)u,$$

$$\mathcal{N}u=\frac{\Delta t}{2}Nu+F,\quad F=u^n-\frac{\Delta t}{2}Lu^n+\frac{\Delta t}{2}Nu^n$$

so that the numerical steady Lyapunov-Schmidt method *LiScNLS* [4] could be applied.

Remark 1. If we have a second order in time equation,

$$\frac{\partial^2 u}{\partial t^2}+Lu=Nu,\quad x\in(a,b),\ t>0,$$

with the same boundary conditions and initial conditions

$$u(x,0)=u_0(x),\quad u_t(x,0)=v_0(x),$$

the Crank-Nicolson discretization looks like

$$\frac{u^{n+1}-2u^n+u^{n-1}}{\Delta t^2}=\frac{1}{2}\left(-Lu^{n+1}+Nu^{n+1}-Lu^{n-1}+Nu^{n-1}\right)$$

i.e.

$$u^{n+1}+\frac{\Delta t^2}{2}Lu^{n+1}=\frac{\Delta t^2}{2}Nu^{n+1}+u^{n-1}-\frac{\Delta t^2}{2}Lu^{n-1}+\frac{\Delta t^2}{2}Nu^{n-1}+2(u^n-u^{n-1}).$$

Remark 2. The backward-Euler method is

$$\frac{u^{n+1}-2u^n+u^{n-1}}{\Delta t^2}=-Lu^{n+1}+Nu^{n+1}$$

i.e.

$$u^{n+1}+\Delta t^2Lu^{n+1}=\Delta t^2Nu^{n+1}+2u^n-u^{n-1}.$$

If $\lambda_k,\varphi_k,\ \ k=1,2,\dots,$ are the eigenvalues and the eigenfunctions of the Sturm-Liouville operator L, then $1+\dfrac{\Delta t}{2}\lambda_k,\varphi_k,\ \ k=1,2,\dots,$ are the eigenvalues and the eigenfunctions of the operator $\mathcal{L}$. Let us suppose that we know the first n eigenfunctions and eigenvalues of $\mathcal{L}$,

$$\mathcal{L}\Phi_k = \lambda_k \Phi_k, k = 1, \dots, n$$

where

$$\int_a^b \Phi_k \Phi_j w dx = \delta_{kj}, \quad k, j = 1, \dots, n.$$

Then, we search for the solution of the nonlinear steady problem

$$\mathcal{L}u = \mathcal{N}u \tag{10}$$

of the form (see [5],[6] for the hypotheses on $\mathcal{L}$ and $\mathcal{N}$)

$$u = \sum_{i=1}^{n} c_i \Phi_i = \Phi \cdot c$$

The nonlinear part is

$$\mathcal{N}(u) = \mathcal{N}\left(\sum_{i=1}^{n} c_i \Phi_i\right) = \sum_{i=1}^{n} C_i \Phi_i$$

where

$$C_i = \int_a^b \mathcal{N}(u) \cdot \Phi_i \cdot w dx, \quad i = 1, \dots, n.$$

Let us choose an index m and project the equation $\mathcal{L}u = \mathcal{N}u$ on $span\,\{\Phi_{m+1}, \dots, \Phi_n\}$, i.e.

$$c_i = \frac{1}{\lambda_i} C_i(c_1, \dots, c_n), \quad i = m+1, \dots, n. \tag{11}$$

For a sufficiently great m, for fixed $c_1, \dots, c_m$, the above operator becomes a contraction so we can iterate until a fixed point

$$c^* = \left(c_1, \dots, c_m, c^*_{m+1}, \dots, c^*_n\right)$$

which is a solution of the equations (11). Of course, c^*_i, $i = m+1, \dots, N$, depend on $c_i, i = 1, \dots, m$.

Now we project the equation $\mathcal{L}u = \mathcal{N}u$ on $span\,\{\Phi_1, \dots, \Phi_m\}$, i.e.

$$\lambda_i c_i = C_i(c_1, \dots, c_m, c^*_{m+1}, \dots, c^*_n) \tag{12}$$

which represents a nonlinear algebraic system for $c_1, \dots, c_m$. Given $c_1, \dots, c_m$, each evaluation of $C_i\left(c_1, \dots, c_m, c^*_{m+1}, \dots, c^*_n\right)$ means the fixed point iterations (11). We solve this system by a Newton method and finally we obtain the solution

$$c^* \equiv \left(c^*_1, \dots, c^*_m, c^*_{m+1}, \dots, c^*_n\right)$$

(i.e. $u = \Phi \cdot c^*$) of the problem (10).

This problem has a natural extension for a nonlinear part of the form $N(x, u(x), u'(x))$, that is

$$N(x, \sum_{i=1}^{n} c_i \Phi_i, \sum_{i=1}^{n} c_i \Phi_i')$$

The main function `of` *LiScNLE* is the function `evol` for the first order (in time) problems:

```
function [lam,phi,phip,x,C,kod]=...
   evol(n,errtol,Lfun,m,Nfun,ICfun,dt,K,scene)
```

Here `n` is the dimension of the discretized problem, `errtol` is the tolerance used in the stopping criteria, `Lfun` describes the linear part of the equation (see *LiScEig Tutorial* [3]), `m` is the truncation parameter.

The nonlinear r.h.s. of the problem (10) is coded in `Nfun (see` ***LiScNLS Tutorial*** `[4]),ICfun` describes the initial condition $u_0(x)$, `dt` is the time step, `K` is the number of time steps to be performed and `scene` is used for the plot of the solution.

For the second order (in time) problems the corresponding file is `evol2`.

The output parameters of `evol` are:

`lam` – the eigenvalues of the linear part

`phi, phip` – the eigenfunctions and their derivatives

`x` – the grid

`C` – the coefficients of the numerical solution with respect to the eigenfunctions `phi`, column $n + 1$ for the $n-$ time level, $n = 0, 1, \ldots, K$.

`kod` – indicates the status of the solution.

We remark that, given the coefficients C of the solution with respect to the eigenfunctions `phi`, the values of the solutions at the Legendre grid points x are $phi * C$ and a plot of the solution could be obtained using the command `plot(x,phi*C);`

More details about the implementation could be found in the tutorial of LiScNLE [11].

4 Examples

The tutorial of *LiScNLE* [11] contains many difficult examples:

– the Burgers equation, which exhibits a near shock,

– a steady solution of a nonlinear reaction-difusion problem,

– the blowing-up behavior for a forced heat equation,

– the Allen-Cohn equation,

– periodical stable and unstable solutions for Kuramoto-Sivashinski equation,

– a moving step solution for Fisher equation,

– an example from electrodynamics (system, also in MATLAB demo),

– the sine-Gordon equation (second order in time).

Let us present here the Fisher equation example,

$$\begin{aligned} u_t - u_{xx} &= u(1-u), x \in \mathrm{I\!R}, \mathrm{t} > 0, \\ u(t, -\infty) &= 1, u(t, \infty) = 0, \\ u(0, x) &= \frac{1}{\left(1 + e^{\frac{x}{\sqrt{6}}}\right)^2} \end{aligned}$$

with the exact solution

$$u(t, x) = \frac{1}{\left(1 + e^{-\frac{5t}{6} + \frac{x}{\sqrt{6}}}\right)^2}.$$

First, the spatial domain is truncated to $[-50, 50]$. Next, the boundary values are homogenized by using the function $u_0(x) = (50 - x)/100$. The basic command is

```
[lam,phi,phip,x,C,kod]=evol(256,1.e-5,'LFisher',...
  0,'NFisher','ICFisher',0.01,1000,[-50 50 0 1]);
```

The absolute error is $2.5x10^{-3}$ in the region of the step and about 10^{-5} in general, due to truncation of the spatial domain.

5 Conclusions

The comparison between `LiScNLE` and `SBVP 1.0` of Auzinger [12] or `bvp4c` of Matlab (see Matlab help) shows an important reduction of the computing time for LiScNLE.. 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 calculations of the numerical solution. We have good reasons to use LS method.

1. We can build a database with known eigenfunctions.

2. In the problems with parameters, where we have (for example) bifurcations, or in evolution problems, we can use repeatedly the same eigenfunctions.

3. The eigenfunctions carry physical information, so that our LS solution has a better structure for studies.

4. LS method could be easily extended to 2D or 3D (evolution) problems, with non-invertible linear part.

5. 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] Cesari L. *Functional Analysis and Galerkin's Method*. Mich. Math. J., 1964, **11**, N 3, p. 383–414.

[2] Cañada A., Ureña A.J. *Asymptotic Behaviour of the Solvability Set for Pendulum Type Equations with Linear Dampuing and Homogeneous Dirichlet Conditions*. Electron. J. Diff. Eqns., Conf. 06, 2001, p. 55–64, `http://ejde.math.unt.edu`.

[3] Trif D. *LiScEig Tutorial 2005*. MATLAB Central > File Exchange > Mathematics > Differential Equations > LiScEig 1.0, `http://www.mathworks.nl/matlabcentral/fileexchange`.

[4] Trif D. *LiScNLS Tutorial 2005*. MATLAB Central > File Exchange > Mathematics > Differential Equations > LiScNLS 1.0, `http://www.mathworks.nl/matlabcentral/fileexchange`.

[5] Petrila T., Trif D. *Basics of Fluid Mechanics and Introduction to Computational Fluid Dynamics*. Springer, 2005.

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

[7] Ledoux V. *MATSLISE package*. `http://users.ugent.be/~vledoux/MATSLISE/`.

[8] Adomaitis R.A., Lin Yi-hung. *A Collocation/Quadrature Based Sturm-Liouville Problem Solver*. ISR Technical Research Report TR 99-01.

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

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

[11] Trif D. *LiScNLE Tutorial 2006*. MATLAB Central > File Exchange > Mathematics > Differential Equations > LiScNLE 1.0, `http://www.mathworks.nl/matlabcentral/fileexchange`.

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

Titus Petrila
Department of Applied Mathematics
Babes-Bolyai University of Cluj-Napoca
Romania

E-mail: *tituspetrila@yahoo.com*

Received September 21, 2006