# A Study of Nonlinear Singular BVP In Circular Membrane: Interpolation-Based Fail-Proof Matlab Solver 



Ragini

Research Scholar, University Department of Electronics, BRA Bihar University Muzaffarpur, Bihar, India


#### Abstract

A singular two-point boundary value problem associated with a nonlinear ordinary differential equation arising in circular membrane is solved using successive linear interpolation based iterative technique. The technique coded in Matlab makes use of mathematical lower and upper solution bounds. It overcomes the integration tolerance problem which could be encountered and obviates possible numerical failure


Keywords : Linear Interpolation Based Iterative Technique, Matlab, Ordinary Differential Equation

## I. INTRODUCTION

We consider the following three nonlinear singular boundary value problems (BVPs) associated with an ordinary differential equation (ODE). Problems P1, P2, and P3 The second order ODE that models the radial stress in a circular membrane under normal uniform pressure may be written as [1-5]

$$
\begin{equation*}
D^{2} y+(3 / x) D y+k / y^{2}=0, \quad 0<x<1 \tag{1}
\end{equation*}
$$

where $D \equiv d / d x, D^{2} \equiv d^{2} / d x^{2}, k>0$ is a constant, $x$ is the radial coordinate and $y(x)$ the radial stress. At the center (for symmetry), i.e., at $x=0$,

$$
\begin{equation*}
D y(0)=0 \tag{2}
\end{equation*}
$$

At the edge, i.e., at $x=1$, we have the condition

$$
\begin{equation*}
y(1)=\lambda>0 \tag{3}
\end{equation*}
$$

where $\lambda$ is a constant which represents radial stress at the boundary viz. $\mathrm{x}=1$, or the condition

$$
\begin{equation*}
D y(1)+(1-v) y(1)=0, \quad 1-v>0 \tag{4}
\end{equation*}
$$

where $v$ is the poisson's ratio.
If we now introduce the change of variable, in equation (1), $x=1 / t$, then we get

$$
\begin{equation*}
D^{2} y-(1 / t) D y+\left(k / t^{4}\right)\left(1 / y^{2}\right)=0, \quad 1<t<\infty \tag{5}
\end{equation*}
$$

as an infinite interval boundary value problem (BVP), where $y(1)=\lambda>0$ and the value of $D y(1)$ is such that $y(t)$ remains bounded. Or, more precisely, $D y(\infty)=0$.

We focus on the numerical solution of the following singular nonlinear boundary value problems using known analytically derived lower and upper solutions [2], where $v_{1}=1-v$ and $v_{3}=3-v$ :
(i) Problem PI ODE (1) with BCs (1), (3), and $y(x) \in\left[\lambda,\left(k /\left(8 \lambda^{2}\right)\right)\left(1-x^{2}\right)+\lambda\right]$, where $\lambda$ is the lower solution and $\left(k /\left(8 \lambda^{2}\right)\right)\left(1-x^{2}\right)+\lambda$ is the upper solution for $P 1$.
(ii) Problem P2 ODE (1) with BCs (2), (4), and

$$
y(x) \in\left[\left(k v_{1}^{2} / v_{3}^{2}\right)^{1 / 3}\left(\left(v_{3} / v_{1}\right)-x^{2}\right) / 2,\left(k v_{1}^{2} / 32\right)^{1 / 3}\left(\left(v_{3} / v_{1}\right)-x^{2}\right)\right],
$$

(iii) Problem P3 ODE(5) with BCs (3), $D y(\infty)=0$ and

$$
y(t) \in\left[\lambda, \lambda+\left(\left(k /\left(8 \lambda^{2}\right)\right)\left(1-\left(1 / t^{2}\right)\right)\right],\right.
$$

the first and the second functions within the square brackets correspond to lower and upper solutions of the respective problems P1, P2, and P3. Note that the lower solution of P1 and P3 is a constant function $y(x)=\lambda$.

Problems P1 and P3 are essentially the same except that P3 is derived by the nonlinear transformation $x=1 / t$ from P1. This transformation has both advantages and disadvantages in numerical implementation. However, we discuss to which extent the transformation helps to enable us to select one of P1 and P3 for computer implementation. The knowledge of lower and upper solutions, though not essential, is helpful specifically when the problem is computationally sensitive [7].

The computational procedures to solve the problems P1, P2, and P3 that include a simple linear interpolation method for quick convergence of these singular 2-point nonlinear BVPs in just a few iterations are described in Sec. 2. Sec. 3 consists of the concerned Matlab programs along with the numerical solutions and the corresponding graphical representations while conclusions are included in Sec. 4.

## Matlab Programs for SLIP with Outputs

We present Matlab programs and the numerical results along with graphs for a reader to readily visualize and appreciate the computational importance of the membrane problem under consideration. These programs can
be readily copied, pasted, and modified, if necessary. This will enable a reader to carry out numerical experiments, see for herself/himself the utility of the programs in many similar possible problems and get herself/himself enlightened.

It can be seen that these Matlab programs have been automated as well as made fail-proof unlike the ones presented in [1].

Problem P1 Consider the problem P1: $\quad y^{\prime \prime}+k / y^{2}+(3 / x) y^{\prime}=0,0<x<1$ with boundary conditions $y^{\prime}(0)=0, \quad y(1)=\lambda>0$, where $\lambda \leq y(x) \leq\left(\mathrm{k} / 8 \lambda^{2}\right)\left(1-x^{2}\right)+\lambda$. Observe that Problem P1 is free from parameter $v$ but depends on $\lambda$ The Matlab program for this problem, is as follows.
The procedure
The numerical solution procedures for a general 2-point BVP associated with an ODE - linear or nonlinear, coupled or not - have already been discussed [8-10]. In the procedure to solve anyone of the nonlinear BVPs P1, P2, and P3 we have chosen arbitrarily a numerical value for the Dependent variable y at $\mathrm{x}=0$ ( $0+\mathrm{in}$ an engineering application since $\mathrm{x}=0$ is a singular point) and computed the true initial conditions which corresponds to the concerned boundary conditions. As a matter of fact we convert the BVP into an initial value problem (IVP) and the nonlinear BVP is then solved iteratively using successive linear interpolation. The iteration uses the mathematical lower and upper solution bounds within which the true numerical solution is present. If the bounds bracket a reasonably narrow interval, then we do not normally encounter any integration tolerance (dominant/unacceptable error) problem and the SLIP can be implemented without modification of the lower or upper bound to obtain the solution of the BVP. If, on the other hand, the interval is wider, the integration tolerance problem could crop up. This indicates one of the lower and upper bounds that were used in converting the 2-point BVP into an IVP is far away from the original initial condition and needs to be modified. If lower bound is away from the original initial condition then we modify the lower bound else the upper bound and implement the SLIP in the shrunk (new) interval. We continue this process successively till we get a sufficiently accurate solution of the problem. In order to achieve shrinking we divide the most recent interval into a number of subintervals and discard that subinterval which does not contain the original initial condition.

## CONCLUSION

Need for lower and upper solutions In the successive linear interpolation procedure to solve the nonlinear singular BVPs we have made use of mathematically derived upper and lower solution bounds. These bounds are not essential to solve the singular BVPs. However, the knowledge of the reasonably narrow bounds imposed by the lower and upper solutions is useful particularly, in sensitive problems, i.e., the problems where the solution is violently fluctuating [7]. Interestingly we, through our numerical experiments, discovered that such bounds are indeed useful for solving ODEs similar to problems P1, P2, and P3. In fact, if we do not make use of the bounds, we might end up searching vast real region without hitting at the true initial value(s) of the given BVP. This is particularly so when the singular nonlinear ODE is ill-posed. Searching for a solution in a reasonably narrow interval bounded by lower and upper solutions not only minimizes the computing time but also obviates the failure in obtaining the solution.

Presented Matlab programs versus the ones in [1]The Matlab programs here are more general than given in [1] on two counts. First, we have used only symbolic parameters as input arguments of the program instead of their actual numerical values as used in [1]. For instance, in [1], an explicit numerical value is used instead of symbolic parameter $k$ in the subprogram "ydash". This is avoided by declaring k as global variable in both the main program and the concerned subprogram.
Evidently, this makes the programs more general. Also the programs avoid undesired modifications in their command structure whenever input numerical data are changed.
Second, and most importantly, the integration tolerance problem is obviated for almost all real-world problems. The programs along with the appropriate comments inserted are simple and easily readable without practically requiring formal programming knowledge. Morever, they are also flexible to take care of wider variety of problems. If a problem which does not come under this variety can still be tackled by a simple modification of a command or two in the program.
Scope of the Matlab programs among other programming languages The Matlab programs are very high-level and easily readable without needing practically any knowledge of a formal programming language. This is quite unlike other lower level programming languages such as C, C++, and FORTRAN. Appropriate comments embedded in the Matlab programs are helpful to the reader and permit easy modification for solving other BVPs associated with a system of ODEs, linear or nonlinear, coupled or not [7, 8,9].
Why linear interpolation and not nonlinear one or bisection For a sensitive function $y(x)$ a nonlinear interpolation also becomes sensitive and may produce a value which may cross the region of convergence and hence may fail. Bisection, on the other hand, may not fail but would take more iterations as it does not use the available knowledge of function values except, of course their signs. Linear interpolation, particularly a nonextrapolatory one, is much less sensitive and makes use of the already known function values to derive the benefit of nonlinear interpolation and that of bisection.
Why four significant digits in the independent variable $x$ No measuring device can produce accuracy more than $0.005 \%$. For any engineering application, four significant digits in the final result (to be used) are enough [11].

## REFERENCES

[1]. S K Sen, H Agarwal, and T samanta, "Interpolation for Nonlinear BVP in Circular Membrane with Known Upper and Lower Solutions", Computers and Mathematics with Applications, 51 (2006), 10211046.
[2]. R.P. Agarwal and D. O'Regan, Singular problems arising in circular membrane theory, Dynamics of Continuous, Discrete and Impulsive Systems10(2003), 965-972.
[3]. A. Constantin, On an infinite interval boundary value problem, Annali di Mathematica pura ed applicata 176(1999), 379-394.
[4]. R.W. Dickey, The plane circular elastic surface under normal pressure, Arch. Rat. Mech. Anl. 26(1967), 219-236.
[5]. J.Y. Shin, A singular nonlinear boundary value problem in the nonlinear circular membrane under normal pressure, J. Korean Math. Soc. 32( 1995), 761-773.
[6]. Irena Rachunková, Gernot Pulverer, and Ewa B. Weinmüller, "A unified approach to singular problems arising in the membrane theory", Applications Of Mathematics, 55 (2010), No. 1, 47-75.
[7]. S.K. Sen and H. Agarwal, Mean value theorem for boundary value problems with given upper and lower solutions, Computers and Mathematics with Applications, 49 (2005), 1499-1514.
[8]. E.V. Krishnamurthy and S.K. Sen, Programming in MATLAB, Affiliated East-West Press, New Delhi (2003).
[9]. S.K. Sen and M. Chanda (1972): A general scheme for solving ordinary differential equations under twopoint boundary conditions, J. Ind. Inst. Sci., 54 (1972), 139 -145.
[10]. E.V. Krishnamurthy and S.K. Sen, Numerical Algorithms: Computations in Science and Engineering, Affiliated East-West Press, New Delhi (2001).
[11]. V. Lakshmikantham and S.K. Sen, Computational Error and Complexity in Science and Engineering, Elsevier, Amsterdam (2005).

