International Journal of Modern Mathematical Sciences, 2014, 12(2): 64-73

Modern Scientific Press
International Journal of Modern Mathematical Sciences
ISSN:2166-286X
Journal homepage:www.ModernScientificPress.com/Journals/ijmms.aspx
Florida, USA

Article

Solving Multi-Parameter Eigenvalue Problem Using Osculator Interpolation Method

Luma. N. M. Tawfiq* & **Doaa. R. Abod**

Department of Mathematics, College of Education Ibn Al-Haitham, Baghdad University

* Author to whom correspondence should be addressed; Email: drluma_m@yahoo.com

Article history: Received 27 August 2014, Received in revised form 30 October 2014, Accepted 5 November 2014, Published 8 November 2014.

Abstract: This paper is concerned with Osculator interpolation polynomial to solve multi - parameter eigenvalue problems for ordinary differential equations. The method finds the multi - parameter eigenvalues and the corresponding nonzero eigenvector can be decoupled using new technique which represent the solution of the problem in a certain domain. Illustration examples is presented, which confirm the theoretical predictions with a comparison between suggested technique and other methods.

Keywords: Ordinary differential equation, Eigenvalue, Eigenvector, Interpolation polynomial, multi-parameter eigenvalue problems.

Mathematics Subject Classification (2010): 34L05, 34B15, 34L15, 34L16, 34L20, 41A05, 42A15, 65D05

1. Introduction

The present paper is concerned with a repeated interpolation polynomial to solve multi - parameter eigenvalue problems using osculator interpolation polynomial. The problem involves finding an eigenvalue λ and the corresponding nonzero eigenvector that satisfy the solution of the problem.

The eigenvalue problems can be used in a variety of problems in science and engineering. For example, in oscillation analysis with damping [1-3] and stability problems in fluid dynamics [4], and the three-dimensional (3D) Schrödinger equation can result in a cubic eigenvalue problem [5].

Many methods are used to solve multi - parameter eigenvalue problems (MPEP). Polynomial eigenvalue problems are typically solved by linearization [6], [7], which promotes the k-th order n × n matrix polynomial into the larger kn × kn linear eigenvalue problem. Other methods, such as Arnoldi shift and invert strategy [8], can be used when several eigenvalues are desired. A disadvantage of the shift-and-invert Arnoldi methods is that a change of the shift parameter requires a new Krylov subspace to be built. Another approach is a direct solution obtained by means of the Jacobi-Davidson method [9], although this method has been investigated far less extensively.

In the present paper, we propose a series solution of multi - parameter eigenvalue problems by means of the osculator interpolation polynomial. The proposed method enables us to obtain the eigenvalues and the corresponding nonzero eigenvector of the 2nd order BVP.

The remainder of the paper is organized as follows. In the next section, we introduce the osculator interpolation polynomial. In section 3, we present the suggested technique. Illustration example is shown in Section 4. Finally, conclusions are presented in Section 5.

2. Osculator Interpolation Polynomial

In this paper we use two-point osculatory interpolation polynomial, essentially this is a generalization of interpolation using Taylor polynomials. The idea is to approximate a function y by a polynomial P in which values of y and any number of its derivatives at a given points are fitted by the corresponding function values and derivatives of P [10].

We are particularly concerned with fitting function values and derivatives at the two end points of a finite interval, say [0, 1], i.e., $P^{(j)}(x_i) = f^{(j)}(x_i)$, $j = 0, \dots, n$, $x_i = 0, 1$, where a useful and succinct way of writing osculatory interpolant P_{2n+1} of degree 2n + 1 was given for example by Phillips [11] as:

$$P_{2n+1}(x) = \sum_{j=0}^{n} \{ y^{(j)}(0)\, q_j(x) + (-1)^j\, y^{(j)}(1)\, q_j(1-x) \}, \tag{1}$$

$$q_j(x) = (x^j / j!)(1-x)^{n+1} \sum_{s=0}^{n-j} \binom{n+s}{s} x^s = Q_j(x) / j!, \tag{2}$$

so that (1) with (2) satisfies:

$$y^{(j)}(0) = P_{2n+1}^{(j)}(0)\ , \quad y^{(j)}(1) = P_{2n+1}^{(j)}(1)\ , \quad j = 0, 1, 2, \dots, n\ .$$

implying that P_{2n+1} agrees with the appropriately truncated Taylor series for y about x = 0 and x = 1. We observe that (1) can be written directly in terms of the Taylor coefficients a_i and b_i about x = 0 and x = 1 respectively, as:

$$P_{2n+1}(x) = \sum_{j=0}^{n} \{ a_j Q_j(x) + (-1)^j b_j Q_j(1-x) \}, \qquad (3)$$

3. Solution of the Multi - Parameter Eigenvalue Problems

In this section, we suggest a repeated interpolation technique which is based on osculatory interpolating polynomials P_{2n+1} and Taylor series expansion to solve 2nd order multi-parameter eigenvalue problems. A general form of 2nd order MPEVP is:

$$y''(x) = f(x, y, y', \lambda_i), \quad i = 1,\dots,n,\ n \in I^+, \quad 0 \le x \le 1, \qquad (4a)$$

Subject to the boundary condition (BC):

In the case Dirichlet BC: $y(0) = A,\ y(1) = B$, where $A, B \in R$ (4b)

In the case Neumann BC: $y'(0) = A,\ y'(1) = B$, where $A, B \in R$ (4c)

In the case Cauchy or mixed BC: $y(0) = A,\ y'(1) = B$, where $A, B \in R$ (4d)

Or $y'(0) = A,\ y(1) = B$, where $A, B \in R$ (4e)

where f, in general, nonlinear function of their arguments.

Now, to solve the problem by the suggested method doing the following steps:

Step one

Evaluate Taylor series of y(x) about x = 0:

$$y = \sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + \sum_{i=2}^{\infty} a_i x^i \qquad (5)$$

where $y(0) = a_0$, $y'(0) = a_1$, $y''(0) / 2! = a_2, \dots, y^{(i)}(0) / i! = a_i$, i= 3, 4,…

And evaluate Taylor series of y(x) about x = 1:

$$y = \sum_{i=0}^{\infty} b_i (x-1)^i = b_0 + b_1 (x-1) + \sum_{i=2}^{\infty} b_i (x-1)^i \qquad (6)$$

where $y(1) = b_0$, $y'(1) = b_1$, $y''(1) / 2! = b_2, \dots, y^{(i)}(1) / i! = b_i$, i = 3, 4,…

Step two

Insert the series form (5) into equation (4a) and put x = 0, then equate the coefficients of powers of x to obtain a_2.

Insert the series form (6) into equation (4a) and put x = 1, then equate the coefficients of powers of (x-1) to obtain b_2.

Step three

Derive equation (4a) with respect to x, to get new form of the equation say (7) as follows:

$$y'''(x) = \frac{df(x, y, y', \lambda_i)}{dx} \qquad (7)$$

then, insert the series form (5) into equation (7) and put x = 0 and equate the coefficients of powers of x to obtain a_3, again insert the series form (6) into equation (7) and put x = 1, then equate the coefficients of powers of (x-1) to obtain b_3.

Step four

Iterate the above process many times to obtain a_4, b_4 then a_5, b_5 and so on, that is, to get a_i and b_i for all $i \geq 2$, the resulting equations can be solved using MATLAB version 7.12, to obtain a_i and b_i for all $i \geq 2$.

Step five

The notation implies that the coefficients depend only on the indicated unknowns a_0, a_1, b_0, b_1, and λ_i, i= 1,…,n, $n \in I^+$, use the BC to get two coefficients from these unknown coefficients.

Now, we can construct two point osculatory interpolating polynomial $P_{2n+1}(x)$ by insert these coefficients (a_i's and b_i's) into equation (3).

Step six

To find the unknowns coefficients integrate equation (4a) on [0, x] to obtain:

$$y'(x) - y'(0) - \int_0^x f(s, y, y', \lambda_i)\, ds = 0 \qquad (8a)$$

and again integrate equation (8a) on [0, x] to obtain:

$$y(x) - y(0) - y'(0)\, x - \int_0^x (1 - s)\, f(s, y, y', \lambda_i)\, ds = 0 \qquad (8b)$$

Step seven

Putting x = 1 in equations (8) to get:

$$b_1 - a_1 - \int_0^1 f(s, y, y', \lambda_i)\, ds = 0 \qquad (9a)$$

and

$$b_0 - a_0 - a_1 - \int_0^1 (1 - s)\, f(s, y, y', \lambda_i)\, ds = 0 \qquad (9b)$$

Step eight

Use $P_{2n+1}(x)$ which constructed in step five as a replacement of y(x), we see that equations (9) have only two unknown coefficients from a_0, a_1, b_0, b_1 and λ_i. If the BC is Dirichlet, that is, we have a_0 and b_0, then equations (9) have two unknown coefficients a_1, b_1 and λ_i. If the BC is Neumann, that is, we have a_1 and b_1, then equations (9) have two unknown coefficients a_0, b_0 and λ_i. Finally, if the BC is mixed condition, i.e., we have a_0 and b_1 or a_1 and b_0, then equations (9) have two unknown coefficients a_1, b_0 or a_0, b_1 and λ_i.

Step nine

If n = 3, then

In the case of Dirichlet BC, we have:

$$F(a_1, b_1, \lambda_i) = b_1 - a_1 - \int_0^1 f(s, y, y', \lambda_i)\, ds = 0 \tag{10a}$$

$$G(a_1, b_1, \lambda_i) = b_0 - a_0 - a_1 - \int_0^1 (1-s)\, f(s, y, y', \lambda_i)\, ds = 0 \tag{10b}$$

$$(\partial F/\partial a_1) - (\partial G/\partial a_1) = 0 \tag{10c}$$

$$(\partial F/\partial b_1) - (\partial G/\partial b_1) = 0 \tag{10d}$$

$$(\partial F/\partial a_1)\, (\partial G/\partial b_1) - (\partial F/\partial b_1)\, (\partial G/\partial a_1) = 0 \tag{10e}$$

In the case of Neumann BC, we have:

$$F(a_0, b_0, \lambda_i) = b_1 - a_1 - \int_0^1 f(s, y, y', \lambda_i)\, ds = 0 \tag{11a}$$

$$G(a_0, b_0, \lambda_i) = b_0 - a_0 - a_1 - \int_0^1 (1-s) f(s, y, y', \lambda_i)\, ds = 0 \tag{11b}$$

$$(\partial F/\partial a_0) - (\partial G/\partial a_0) = 0 \tag{11c}$$

$$(\partial F/\partial b_0) - (\partial G/\partial b_0) = 0 \tag{11d}$$

$$(\partial F/\partial a_0)\, (\partial G/\partial b_0) - (\partial F/\partial b_0)\, (\partial G/\partial a_0) = 0 \tag{11e}$$

In the case of mixed BC, we have:

$$F(a_1, b_0, \lambda_i) = b_1 - a_1 - \int_0^1 f(s, y, y', \lambda_i)\, ds = 0 \tag{12a}$$

$$G(a_1, b_0, \lambda_i) = b_0 - a_0 - a_1 - \int_0^1 (1-s) f(s, y, y', \lambda_i)\, ds = 0 \tag{12b}$$

$$(\partial F/\partial a_1) - (\partial G/\partial a_1) = 0 \tag{12c}$$

$$(\partial F/\partial b_0) - (\partial G/\partial b_0) = 0 \tag{12d}$$

$$(\partial F/\partial a_1)\, (\partial G/\partial b_0) - (\partial F/\partial b_0)(\partial G/\partial a_1) = 0 \tag{12e}$$

Or

$$F(a_0, b_1, \lambda_i) = b_1 - a_1 - \int_0^1 f(s, y, y', \lambda_i)\, ds = 0 \tag{13a}$$

$$G(a_0, b_1, \lambda_i) = b_0 - a_0 - a_1 - \int_0^1 (1-s) f(s, y, y', \lambda_i)\, ds = 0 \tag{13b}$$

$$(\partial F/\partial a_0) - (\partial G/\partial a_0) = 0 \tag{13c}$$

$$(\partial F/\partial b_1) - (\partial G/\partial b_1) = 0 \tag{13d}$$

$$(\partial F/\partial a_0)(\partial G/\partial b_1) - (\partial F/\partial b_1)(\partial G/\partial a_0) = 0 \tag{13e}$$

So, we can find the unknown coefficients by solving the system of algebraic equations: (10) or (11) or (12) or (13) using MATLAB, so insert the value of the unknown coefficients into equation (3), thus equation (3) represent the solution of the problem.

4. Example

In this section, we illustrate suggested method using example of multi-parameter eigenvalue problems. The algorithm was implemented in MATLAB version 7.12.

The bvp4c solver of MATLAB has been modified accordingly so that it can solve some class of multi-parameter eigenvalue problems as effectively as it previously solved eigenvalue problems.

The following problem arises in a study of heat and mass transfer in a porous spherical catalyst with a first order reaction. There is a singular coefficient arising from the reduction of a partial differential equation to an ODE by symmetry [12]. The MPEVP is:

$$y'' + \frac{2}{x}y' = \lambda_1^{\,2}\, y e^{\frac{\lambda_2\lambda_3(1-y)}{1+\lambda_3(1-y)}} \quad , \quad x \in [0, 1]$$

The BC (mixed case) are $y(1) = 1$ and the symmetry condition $y'(0) = 0$. Now, we solve this problem by suggested method. Here equations (13) become:

$$F(a_0, b_1, \lambda_1, \lambda_2, \lambda_3) = b_1 - 1 + \lambda_1^2 \int_0^1 s\, y e^{\frac{\lambda_2\lambda_3(1-y)}{1+\lambda_3(1-y)}}\, ds = 0, \tag{13a}$$

$$G(a_0, b_1, \lambda_1, \lambda_2, \lambda_3) = 1 - 2\int_0^1 y(s)ds + \lambda^2 \int_0^1 s\,(1-s)\, y e^{\frac{\lambda_2\lambda_3(1-y)}{1+\lambda_3(1-y)}}\, ds = 0, \tag{13a}$$

$$(\partial F/\partial a_0) - (\partial G/\partial a_0) = 0 \tag{13c}$$

$$(\partial F/\partial b_1) - (\partial G/\partial b_1) = 0 \tag{13d}$$

$$(\partial F/\partial a_0)(\partial G/\partial b_1) - (\partial F/\partial b_1)(\partial G/\partial a_0) = 0 \tag{13e}$$

Now, we have to solve equations (13) for the unknowns a_0, b_1, λ_1, λ_2, λ_3 using MATLAB, then we have $a_0 = 0.64$, $b_1 = 0$, $\lambda_1 = 0.6$, $\lambda_2 = 40$ and $\lambda_3 = 0.2$.

Then from equation (3) we have (where n = 4):

$P_9 = 18.63018391074087\, x^9 - 81.86548669822514\, x^8 + 136.0606573968419\, x^7 - 101.7163034006953\, x^6 + 29.5919367027597\, x^5 - 1.076125341642182\, x^4 + 0.7351374287263839\, x^2 + 0.64.$

Higher accuracy can be obtained by evaluating higher n, now take n = 10, we have:

$P_{21} = 52535.82045238554\ x^{21} - 549779.0547394637\ x^{20} + 2594544.959251828\ x^{19} - 7272910.909707223\ x^{18} + 13413618.49119748\ x^{17} - 17012485.96003969\ x^{16} + 15031675.07293067\ x^{15} - 9139740.80568491\ x^{14} + 3661565.477401016\ x^{13} - 73218.8815826758\ x^{12} + 94190.70616882079\ x^{11} + 7.614357331206804\ x^{10} - 3.760089006497641\ x^{8} + 1.931071345092823\ x^{6} - 1.076125341636627\ x^{4} + 0.7351374287263839\ x^{2} + 0.64$

If we take n = 11, we have:

$P_{23} = 2307273.5847575\ x^{22} - 201189.35259898\ x^{23} - 12048976.340348\ x^{21} + 37826952.9633766\ x^{20} - 79339407.916265\ x^{19} + 116759503.0267302\ x^{18} - 123051968.6595243\ x^{17} + 92896106.64761737\ x^{16} - 49247501.14161112\ x^{15} + 17467140.105114\ x^{14} - 3732006.240472856\ x^{13} + 364068.2388733\ x^{12} + 7.614357331206804\ x^{10} - 3.7600890064976\ x^{8} + 1.9310713450928\ x^{6} - 0.0761253416366\ x^{4} + 0.735137428726\ x^{2} + 0.64.$

Table (1), gave a comparison between P_{21} and P_{23} at the ten equidistance point in the domain. Kubiček et al (see [12], [13]) solved this problem by the collection method and got three solutions assembled by consider a range of parameter values: λ_1, λ_2, λ_3 such that, the values $\lambda_1 = 0.6$, $\lambda_2 = 0.1$, $\lambda_3 = 0.2$, used in ex6bvp.m lead to three solutions that are displayed in Figure (1).

Table 1: Comparison between P_{21} and P_{23} for Example 4.1

x_i	P_{23}	P_{21}	$\lvert P_{23} - P_{21} \rvert$
0	**0.640000000000000**	**0.640000000000000**	**0**
0.1	**0.647246012889855**	**0.647245780444789**	**0.000000232445066**
0.2	**0.668049729702792**	**0.667954814870555**	**0.000094914832236**
0.3	**0.705306859754149**	**0.704121734347959**	**0.001185125406190**
0.4	**0.778311965383488**	**0.776224985001824**	**0.002086980381664**
0.5	**0.889394245754096**	**0.890921071212835**	**0.001526825458740**
0.6	**0.981440965886602**	**0.985476906381475**	**0.004035940494873**
0.7	**1.011922794591708**	**1.013556553221331**	**0.001633758629623**
0.8	**1.008022846066691**	**1.008140292157062**	**0.000117446090371**
0.9	**1.001928697887349**	**1.001928959494405**	**0.000000261607056**
1	**0.999999994519209**	**0.999999960351312**	**0.000000034167897**

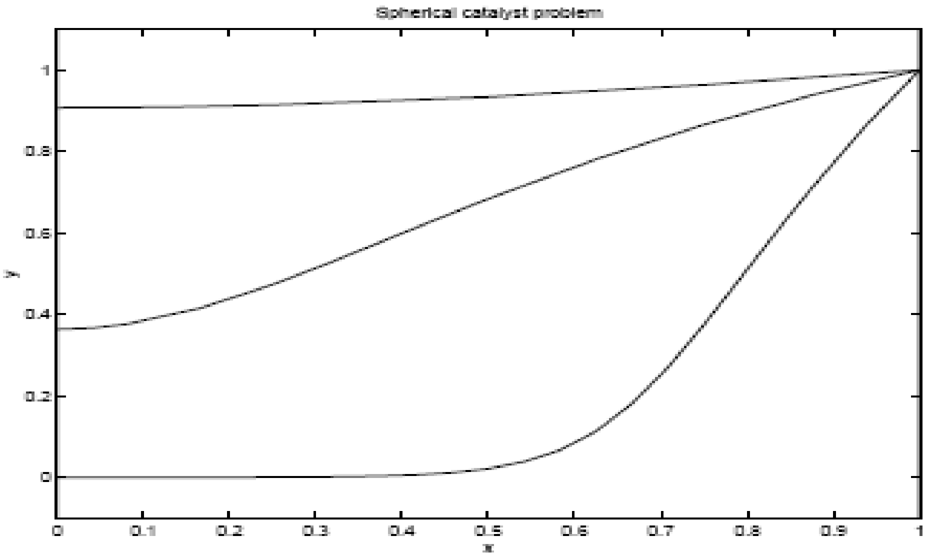

Figure 1: Multiple solutions of example, gave in [12]

Another solution of this problem gave in [14] using collocation method with different code in MATLAB and FORTRAN given in Table **(**2) and Figure (2).

Table 2: Comparisons for different code of solution in [14] with TOL = 10^{-7}

	bvp4c	CW-4	CW-6	sbvp4	sbvp4g	sbvp6	sbvp6g
N	205	41	21	57	57	22	15
f_{count}	6856	1080	840	6051	3699	3741	1431

Where:

– bvp4c (MATLAB 6.0 routine): which is based on collocation at three Lobatto points (see [14]). This is a method of order 4 for regular problems.

– COLNEW (Fortran 90 code): The basic method here is collocation at Gaussian points, we chose the polynomial degrees m = 4 (CW-4) and m = 6 (CW-6), which results in (super convergent) methods of orders 8 and 12 respectively (for regular problems).

– sbvp is used with equidistant (sbvp4 and sbvp6) and Gaussian (sbvp4g and sbvp6g) collocation points and polynomials of degrees 4 and 6 respectively.

Although for reasons mentioned here, the comparison of all three codes is difficult. Table (2) show the number of mesh points (N) and the number of function evaluations (fcount) that the different solvers required to reach tolerance (TOL).

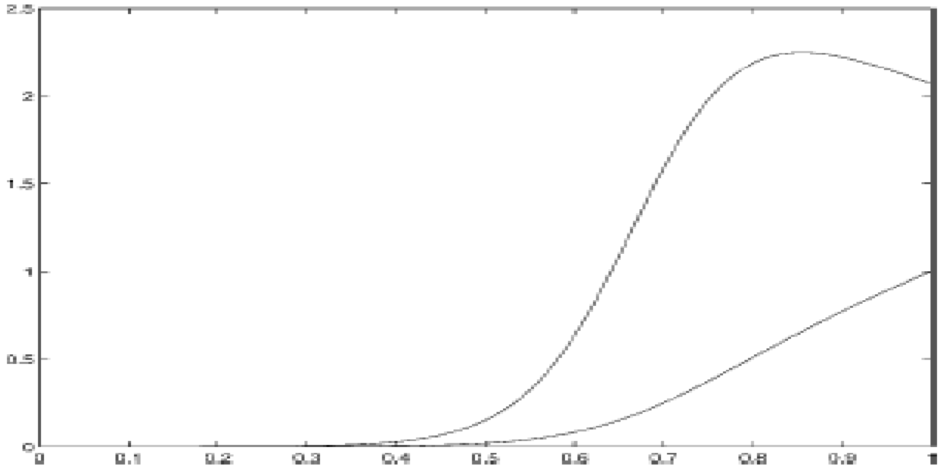

Figure 2: Solution of example gave in [14]

5. Conclusions

In the present paper, we have proposed osculator interpolation method to solve multi-point eigenvalue problems. It may be concluded that the suggested technique is a very powerful, efficient and can be used successfully for finding the solution of nonlinear multi-point eigenvalue problems with boundary condition.

The bvp4c solver of MATLAB has been modified accordingly so that it can solve some class of multi-point eigenvalue problems as effectively as it previously solved BVP.

References

[1] Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., and Van Der Vorst, H., *Templates for the solution of algebraic eigenvalue problems: A practical guide*, SIAM, Philadelphia, 2000.

[2] Higham, N. J., and Tisseur, F., Bounds for eigenvalues of matrix polynomials, *Lin. Alg. Appl.*, 358 (2003): 5-22.

[3] Tisseur, F., Meerbergen, K., The quadratic eigenvalue problem, *SIAM Rev.*, 43 (2001): 235-286.

[4] Heeg, R. S., *Stability and transition of attachment-line flow*, Ph.D, thesis, Universiteit Twente, Enschede, the Netherlands, 1998.

[5] Hwang, T., Lin, W., Liu, J., and Wang, W., Jacobi-Davidson methods for cubic eigenvalue problems, *Numer. Lin. Alg. Appl.*, 12(2005): 605-624.

[6] Mackey, D. S., Mackey, N., Mehl, C., and Mehrmann, V., Vector spaces of linearizations for matrix polynomials, *SIAM J. Matrix Anal. Appl.*, 28(2006): 971-1004.

[7] Pereira, E., On solvents of matrix polynomials, *Appl. Numer. Math.*, 47(2003): 197-208.

[8] Bai, Z., and Su, Y., SOAR: A second order Arnoldi method for the solution of the quadratic eigenvalue problem, *SIAM J. Matrix Anal. Appl*., 26(2005): 640-659.

[9] Asakura, J. Sakurai, T., Tadano, H., Ikegami, T., and Kimura, K., A numerical method for polynomial eigenvalue problems using contour integral, *Japan Journal of Industrial and Applied Mathematics*, 27(1) (2010):73-90 .

[10] Burden, L. R., and Faires, J. D., *Numerical Analysis*, Seventh Edition, 2001.

[11] Phillips, G .M., Explicit forms for certain Hermite approximations, *BIT*, 13(1973): 177-180.

[12] Shampine, L. F., Singular Boundary Value Problems for ODEs, *Southern Methodist University J.*,1(2002): 1-14.

[13] Tawfiq, L. N. M. and Rasheed, H. W., *On Solving Singular Boundary Value Problems and Its Applications*, LAP LAMBERT Academic Publishing, 2013.

[14] Morgado, L., and Lima, P., Numerical Methods for a Singular Boundary Value Problem with Application to a Heat Conduction Model In The Human Head, *Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering* (CMMSE), (2009)**.**