(IJRST) 2015, Vol. No. 5, Issue No. II, Apr-Jun **ISSN: 2249-0604**

USING SEMI-ANALYTIC TECHNIQUE TO SOLVE SINGULAR QUADRATIC EIGENVALUE PROBLEMS

Luma. N. M. Tawfiq, Aseel Hisham
Baghdad University, College of Education for Pure Science, Department of Mathematics.

ABSTRACT

This paper is concerned with a semi-analytic technique to find the solution of quadratic eigenvalue problems for singular ordinary differential equations that arise in the areas of solid mechanics, acoustics and coupled structural acoustics. The technique finds the eigenvalue and the corresponding nonzero eigenvector which represent the solution of the problems in a certain domain. Illustration example is presented, which provided to demonstrate the efficiency and accuracy of the proposed technique where the suggested solution compared with other methods. Also, we propose a new formula developed to estimate the error help reduce the accounts process and show the results are improved.The existing code bvpsuite designed for the estimate the error of solution of boundary value problems was extended by a model for the computation the estimate error for the solution of the problem in this paper.

Keywords: *Ordinary differential equation, Eigenvalue, Eigenvector, Interpolation polynomial, Quadratic eigenvalue problems.*

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

1. INTRODUCTION

The study of quadratic eigenvalue problems for singular ordinary differential equations with boundary conditions is a topic of great interest. In many cases singular quadratic eigenvalue problems (SQEVP′s) model important physical processes [1]. The eigenvalue problems can be used in a variety of problems in science and engineering. For example, quadratic eigenvalue problems arise in oscillation analysis with damping [2], [3] and stability problems in fluid dynamics [4], and the three-dimensional (3D) Schrödinger equation can result in a quadratic eigenvalue problem [5]. singular case, has been studied for several decades. For an overview of the recent work on numerical solutions see, e.g., [6], [7] and references therein

This paper offers a brief introduction to semi analytic technique which used to solve second order SQEVP′s for ODE, which illustrate importance of technique for solving some problems which cannot be solve in other methods or solving but suggested technique is more accurate and easy implemented.

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 [8].

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, . . . , 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 [9] as:

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

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

so that (1) with (2) satisfies :

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

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 :

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

3. SOLVING SINGULAR QUADRATIC EIGENVALUE PROBLEMS

In this section, we suggest a semi-analytic technique which is based on osculatory interpolating polynomials P_{2n+1} and Taylor series expansion to solve 2nd order singular quadratic eigenvalue Problems (SQEVP′s).

A general form of 2nd order SQEVP's is (if the singular point is x = 0):

$$x^m y''(x) = f(x, y, y', \lambda^2), \qquad 0 \le x \le 1 \qquad (4a)$$

where f are in general nonlinear functions of their arguments and m is integer, 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)

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

Step one

Evaluate Taylor series of y(x) about x = 0, i.e.,

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

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

And evaluate Taylor series of y(x) about x = 1, i.e.,

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

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

Step two

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

Step three

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

$$m x^{m-1} y''(x) + x^m y'''(x) = \frac{df(x, y, y', \lambda^2)}{dx} \quad , \qquad (7)$$

then, insert the series form (5) into equation (7) and put x = 0 and equate the coefficients of powers of x to obtain a_2, 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_3, b_4 then a_4, 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, use the BC′s to get two coefficients from these, therefore, we have only two unknown coefficients and λ. Now, we construct two point osculatory polynomial $P_{2n+1}(x)$ by insert these coefficients (a_i and b_i) in to equation (3).

Step six

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

$$x^m y'(x) - m\,x^{m-1}y(x) + m(m-1)\int_0^x s^{m-2}\,y(s)\,ds - \int_0^x f(s, y, y', \lambda^2)\,ds = 0, \qquad (8a)$$

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

$$x^m y(x) - 2m\int_0^x s^{m-1}y(s)\,ds + m(m-1)\int_0^x (1-s)s^{m-2}y(s)ds + \int_0^x (1-s)f(s, y, y', \lambda^2) = 0, \quad (8b)$$

Step seven

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

$$b_1 - mb_0 + m(m-1)\int_0^1 s^{m-2}\,y(s)\,ds + \int_0^1 f(s, y, y', \lambda^2)\,ds = 0, \qquad (9a)$$

and

$$b_0 - 2m\int_0^1 s^{m-1}y(s)\,ds + m(m-1)\int_0^1 (1-s)s^{m-2}\,y(s)\,ds + \int_0^1 (1-s)f(s, y, y', \lambda^2)ds = 0, \quad (9b)$$

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. If the BC is **Dirichlet** , that is, we have a_0 and b_0, then equations (9) have the unknown coefficients a_1, b_1 and λ. If the BC is **Neumann**, that is, we have a_1 and b_1, then equations (9) have unknown coefficients a_0, b_0 and λ. 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 unknown coefficients a_1, b_0 or a_0, b_1 and λ.

Step nine

In the case Dirichlet BC, we have:

$$F(a_1,b_1,\lambda^2) = b_1 - mb_0 + m(m-1)\int_0^1 s^{m-2}\,y(s)\,ds + \int_0^1 f(s,y,y',\lambda^2)\,ds = 0, \qquad (10a)$$

$$G(a_1,b_1,\lambda^2) = b_0 - 2m\int_0^1 s^{m-1}y(s)ds + m(m-1)\int_0^1 (1-s)s^{m-2}y(s)ds + \int_0^1 (1-s)f(s,y,y',\lambda^2)ds = 0, \quad (10b)$$

$$(\partial F/\partial a_1)(\partial G/\partial b_1) - (\partial F/\partial b_1)(\partial G/\partial a_1) = 0, \qquad (10c)$$

In the case Neumann BC, we have:

$$F(a_0,b_0,\lambda^2) = b_1 - mb_0 + m(m-1)\int_0^1 s^{m-2} y(s)\,ds + \int_0^1 f(s, y, y', \lambda^2)\,ds = 0, \quad (11a)$$

$$G(a_0,b_0,\lambda^2) = b_0 - 2m\int_0^1 s^{m-1} y(s)ds + m(m-1)\int_0^1 (1-s)s^{m-2} y(s)ds + \int_0^1 (1-s)f(s,y,y',\lambda^2)ds = 0 \quad (11b)$$

$$(\partial F/\partial a_0)(\partial G/\partial b_0) - (\partial F/\partial b_0)(\partial G/\partial a_0) = 0, \quad (11c)$$

In the case mixed BC, we have:

$$F(a_1,b_0,\lambda^2) = b_1 - mb_0 + m(m-1)\int_0^1 s^{m-2} y(s)\,ds + \int_0^1 f(s, y, y', \lambda^2)\,ds = 0, \quad (12a)$$

$$G(a_1,b_0,\lambda^2) = b_0 - 2m\int_0^1 s^{m-1} y(s)ds + m(m-1)\int_0^1 (1-s)s^{m-2} y(s)ds + \int_0^1 (1-s)f(s,y,y',\lambda^2)ds = 0 \quad (12b)$$

$$(\partial F/\partial a_1)(\partial G/\partial b_0) - (\partial F/\partial b_0)(\partial G/\partial a_1) = 0, \quad (12c)$$

Or

$$F(a_0,b_1,\lambda^2) = b_1 - mb_0 + m(m-1)\int_0^1 s^{m-2} y(s)\,ds + \int_0^1 f(s, y, y', \lambda^2)\,ds = 0, \quad (13a)$$

$$G(a_0,b_1,\lambda^2) = b_0 - 2m\int_0^1 s^{m-1} y(s)ds + m(m-1)\int_0^1 (1-s)s^{m-2} y(s)ds + \int_0^1 (1-s)f(s,y,y',\lambda^2)ds = 0 \quad (13b)$$

$$(\partial F/\partial a_0)(\partial G/\partial b_1) - (\partial F/\partial b_1)(\partial G/\partial a_0) = 0, \quad (13c)$$

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. ERROR / DEFECT WEIGHTS

Every known BVP software package reports an estimate of either the relative error or the maximum relative defect. The weights used to scale either the error or the maximum defect differs among BVP software. Therefore, the BVP component of pythODE allows users to select the weights they wish to use. The default weights depend on whether an estimate of the error or maximum defect is being used. If the error is being estimated, then the BVP component of pythODE uses [23]. In this paper we modify this package to consist: SQEVP′s with named "pythSQEVPODE", which defined as:

$$\frac{\|y(x)-p(x)\|_\infty}{1+\|p(x)\|_\infty}; \qquad 0 \le x \le 1 \qquad (14)$$

where y(x) is the exact solution and P(x) is the suggested solution of the problems of this paper.

If the maximum defect is being estimated, then the "pythSQEVPODE" uses for SQEVP′s:

$$\frac{\left\| p''_{2n+1}(x) - \frac{1}{x} f(x, p(x), p'(x), \lambda^2) \right\|_\infty}{1+\left\| \frac{1}{x} f(x, p(x), p'(x), \lambda^2) \right\|_\infty}; \qquad (15)$$

The relative estimate of both the error and the maximum defect are slightly modified from the one used in BVP SOLVER.

Example 1

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 for more details see [10]. Then we get the following singular quadratic eigenvalue problems:

$$xy''+2y' = x\lambda^2 y e^{\frac{\mu\beta(1-y)}{1+\beta(1-y)}}, \quad x \in [0, 1]$$

with mixed BC: $y(1) = 1$ and the symmetry condition $y'(0) = 0$.

Now, we solve this problem by suggested method. Here equations (13) become:

$$F(a_0, b_1, \lambda, \mu, \beta) = b_1 - 1 + \lambda^2 \int_0^1 s\, y e^{\frac{\mu\beta(1-y)}{1+\beta(1-y)}}\, ds = 0,$$

$$G(a_0, b_1, \lambda, \mu, \beta) = 1 - 2\int_0^1 y(s)ds + \lambda^2 \int_0^1 s(1-s) f(s, y, y')\, ds = 0,$$

$$(\partial F/\partial a_0) - (\partial G/\partial a_0) = 0$$
$$(\partial F/\partial b_1) - (\partial G/\partial b_1) = 0$$
$$(\partial F/\partial a_0)(\partial G/\partial b_1) - (\partial F/\partial b_1)(\partial G/\partial a_0) = 0$$

Now, we have to solve these equations for the unknowns a_0, b_1, λ, μ, β using MATLAB, then we have $a_0 = 0.64$, $b_1 = 0$, $\lambda = 0.6$, $\mu = 40$ and $\beta = 0.2$.

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

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

Higher accuracy can be obtained by evaluating higher n, now take n = 10, i.e.,

P_{21} = 52535.82045238554 x^{21} - 549779.0547394637x^{20} + 2594544.959251828x^{19} - 7272910.909707223x^{18} + 13413618.49119748x^{17}- 17012485.96003969x^{16} + 15031675.07293067x^{15} - 9139740.80568491x^{14} +3661565.477401016x^{13}- 73218.8815826758x^{12}+ 94190.70616882079x^{11}+7.614357331206804x^{10} - 3.760089006497641x^{8} +1.931071345092823x^{6}- 1.076125341636627x^{4}+ 0.7 3513742 872 63839x^{2}+ 0.64

A Also, for more accuracy we take n = 11, i.e.,

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

Kubiček et al (see [10]) solved this problem by the collection method and got three solutions with the values $\lambda = 0.6$, $\mu = 0.1$, $\beta = 0.2$ used in ex6bvp.m lead to three solutions that are displayed in Figure (1), Table (1) compares the solutions in [10] at the origin to values reported in [7].

Table (1): Computed y(0) for three solutions of the example 1

Method in [10]	Kubiček et alia [7]
0.9071	0.9070
0.3637	0.3639
0.0001	0.0001

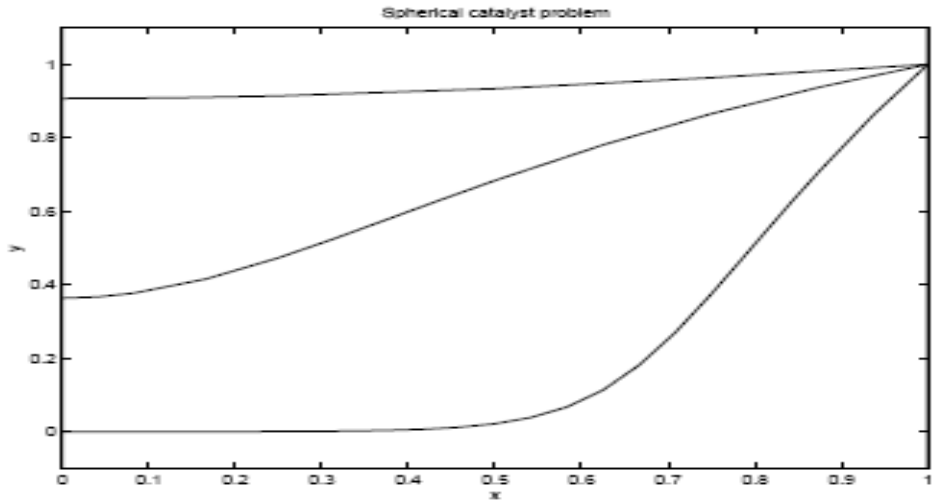

Figure 1: Multiple solutions of example 1, gave in [7]

Another solution of this problem gave in [11] using collocation method with different code in MATLAB and Fortran gave in Table (2) and Figure (2).

Table 2: Comparisons for different code of solution in [11] 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. This is a method of order 4 for regular problems.

– COLNEW (Fortran 90 code): The basic method here is collocation at Gaussian points, 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).

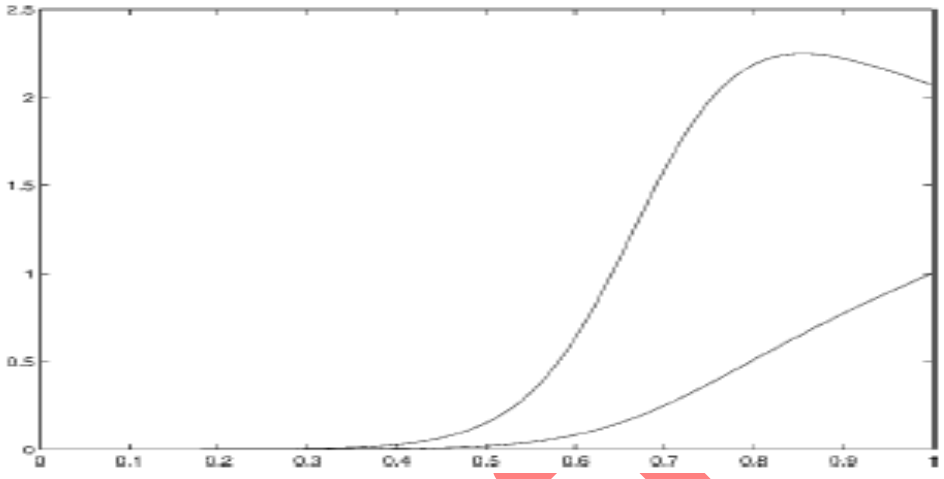

Figure 2: Solution of example 1 gave in [11]

Now, applying equation (15) as follows: n = 10, λ = 0.6, μ = 40 and β = 0.2

$$\frac{\left\| p''_{2n+1}(x) - \frac{1}{x} f(x, p(x), p'(x), \lambda, \mu, \beta) \right\|_\infty}{1 + \left\| \frac{1}{x} f(x, p(x), p'(x), \lambda, \mu, \beta) \right\|_\infty} = \frac{5.046991766682163}{1+3.190094574074464} = 1.470274857452768$$

for more details see Table (3).

5. CONCLUSION

In this paper, we discuss the implementation of our proposed algorithm and investigate the accuracy achieved by applying the semi-analytic technique. The simplicity, efficiency and accuracy of the proposed method is illustrated through example 1.. In this example we see that, if we increase n, we increase the accuracy. Finally, we modify the MATLAB command bvp6c, bvpinit and bvp5c which can solve BVP′s to satisfy our proposed method, the modification involves enlarging to implement for SQEVP instead of SBVP.

REFERENCES

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

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

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

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

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

[6] Muhič, A., and Plestenjak, B., (2008), "On quadratic and singular two-parameter eigenvalue problems", J. of Linear Algebra and Applications, Vol. 46, pp:1-28.

[7] Tawfiq, L. N. M., and Mjthap, H. Z., (2013), "Solving Singular Eigenvalue Problem Using Semi-Analytic Technique", International Journal of Modern Mathematical Sciences, 7, No.1, pp: 121-131.

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

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

[10] Tawfiq, L. N. M., and Rasheed, H. W., (2013),"On Solving Singular Boundary Value Problems and Its Applications", Lap Lambert Academic Publishing.

[11] Ramos, J. I., (2004), "Piecewise quasi-linearization techniques for singular boundary value problems", computer physics communications Journal, Vol. 158, pp:12–25.

Table 3: The Maximum Relative Defect of Example 1

x_i	f(x, p, p′, λ, μ, β)	1+ |f(x, p, p′, λ, μ, β)|	P''_{2n+1}	$\|P''_{2n+1}$-f(x, p, p′, λ, μ, β)|	$\|P''_{2n+1}$ - f(x, p, p′, λ, μ, β)| / (1+ |f(x, p, p′, λ, μ, β)|)
0	0	1.000000000000000	1.470274857452768	1.470274857452768	1.470274857452768
0.1	000388423679298	1.000233054207579	1.349890686018693	1.349657631811114	0.147027485745277
0.2	0.105197905191957	1.063118743115174	1.444290829667609	1.381172086552435	0.134962617820517
0.3	1.260347556603914	1.756208533962349	3.654228939949808	2.898020405987459	0.137250898236432
0.4	3.057609778479788	2.834565867087873	4.526936321820129	2.692370454732256	0.269427688840084
0.5	0.854880679059220	1.512928407435532	-2.057101628638377	2.570030036073909	0.227500567825625
0.6	-3.190094574074464	2.914056744444678	-6.961048511126842	5.046991766682163	0.244463762756740
0.7	-2.481399511031546	2.488839706618927	-3.375301835184350	1.886462128565423	0.423616562766120
0.8	-0.379446599682801	1.227667959809680	0.056823289127933	0.284491248937614	0.164199534220901
0.9	-0.003255753256600	1.001953451953960	0.440160003863823	0.442113455817783	0.027815847176066
1	-000002179574504	1.000001307744702	0.359998008306366	0.359999316051068	0.044202710794601
Max. error				5.046991766682163	1.470274857452768