31

QUADRATIC SPLINE FUNCTION APPROXIMATION FOR SOLUTION OF NONLINEAR BOUNDARY VALUE PROBLEM AND ITS APPLICATION TO HENCKY PROBLEM

by

H. S. Hung

Technical Report #38

January 1969

QUADRATIC SPLINE FUNCTION APPROXIMATION FOR SOLUTION OF NONLINEAR BOUNDARY VALUE PROBLEM AND ITS APPLICATION TO HENCKY PROBLEM

by

H. S. Hung

1. INTRODUCTION

In this paper we consider the nonlinear system of ordinary differential equations

(1.1) $$x'(t) = g(x), \qquad 0 \le t \le b ,$$

where $x(t)$ is the N-dimensional vector function with components $x_i(t)$, and $g(x)$ is the N-dimensional vector function with components $g_i(x)$, $i = 1, \ldots, N$. Let the boundary conditions have the form

(1.2) $$\begin{aligned} (x(0), a_i) &= b_i, & i &= 1, 2, \ldots, k, \\ (x(b), a_i) &= b_i, & i &= k+1, \ldots, N, \end{aligned}$$

where a_i are given vectors and the b_i are given scalars, and (x, y) as usual denotes the vector inner product

(1.3) $$(x, y) = \sum_{i=1}^{n} x_i y_i .$$

It is our aim to obtain the solution for (1.1) and (1.2), when it exists. The computational solution to be described is an iterative method in which all the boundary conditions are satisfied at every iteration. This is accomplished by what is essentially a Newton-Raphson method with a quadratic spline function approximation solution at each iteration. In the present paper we are going to show that the proposed method is convergent, and that the error estimate can be obtained along with the numerical solution. In Section 4 of this paper a numerical example is given to show how the method really works, and in Section 5 we apply the method to the Hencky problem, which is a difficult problem to solve numerically.

2. DESCRIPTION OF THE METHOD

2.1 Newtonian approximation

The material in this section is contained in [1, 2]. We conclude it here for the sake of completeness.

Application of the Newton-Raphson method to (1.1) and (1.2), we generate a sequence of vectors $\{x^n\}$ by means of linear equations.

$$\frac{dx^{n+1}}{dt} = g(x^n) + J(x^n)(x^{n+1} - x^n), \quad n = 0, 1, 2, \ldots, \tag{2.1}$$

and the boundary conditions

$$\begin{aligned} (x^{n+1}(0), a_i) &= b_i, \qquad i = 1, 2, \ldots, k, \\ (x^{n+1}(b), a_i) &= b_i, \qquad i = k+1, \ldots, N, \end{aligned} \tag{2.2}$$

with x^o as initial guess. Here $J(x^n)$ is the Jacobian matrix defined by

$$J(x^n) = \left(\frac{\partial g_i}{\partial x_j}\right)_{x=x^n}, \qquad i, j = 1, \ldots, N. \tag{2.3}$$

If X^{n+1} is the matrix solution of

$$\frac{dX^{n+1}}{dt} = J(x^n)X^{n+1} \tag{2.4}$$

$$X^{n+1}(0) = I,$$

and

p^{n+1} is the vector solution of

$$\frac{dp^{n+1}}{dt} = J(x^n)p^{n+1} + g(x^n) - J(x^n)x^n \tag{2.5}$$

$$p^{n+1}(0) = 0,$$

then the solution of (2.1) and (2.2) has the form

$$x^{n+1} = X^{n+1}c^{n+1} + p^{n+1}, \tag{2.6}$$

where the vector c^{n+1} is determined by the system of N linear equations

$$\begin{aligned} (c^{n+1}, a_i) &= b_i, & i &= 1, 2, \ldots, k, \\ (X^{n+1}(b)c^{n+1} + p^{n+1}(b), a_i) &= b_i, & i &= k+1, \ldots, N. \end{aligned} \tag{2.7}$$

Carrying out these operations requires the simultaneous solution of $N^2 + N$ first order differential equations, and the solution of a set of N simultaneous algebraic equations. There are so many standard methods to solve for X^{n+1} and p^{n+1} in (2.4) and (2.5) respectively, but it is our aim in this paper to introduce the quadratic spline approximation method, which we shall discuss in detail in section (2.2). However, before we leave section (2.1) we shall state but not prove the following theorem, which will be useful, as we shall see later, in proving the convergence of our method.

Theorem 2.1. If $g_i(x_1, x_2, \ldots, x_N)$ are continuous, and $\frac{\partial^2 g_i}{\partial x_i \partial x_j}$ are bounded on $[0, b]$ for $i, j = 1, \ldots, N$, then the sequence $\{x^n\}$ defined by (2.6) and (2.7) converges quadratically to the solution of equations (1.1) and (1.2) assuming that b is sufficiently small.

2.2 Quadratic spline approximation

Spline function approximations for solutions of initial value problems with a single ordinary differential equation has been discussed recently in [3,4]. In this section we extend the idea to system of equations.

Let the system of differential equations be

$$y'(t) = f(t, y(t)), \qquad 0 \le t \le b, \tag{2.8}$$

and the initial conditions be

$$y(0) = y_0, \tag{2.9}$$

where

$y(t)$ is the N-dimensional vector function with components $y_i(t)$, $i = 1, 2, \ldots, N.$; $f(t, y(t))$ is the N-dimensional vector function with components $f_i(t, y(t))$ $i = 1, 2, \ldots, N$; and y_0 is the N-dimensional vector with components $y_{0,i}$, $i = 1, \ldots, N$.

If $\mathcal{S} = \{(t, y) \mid 0 \le t \le b\}$ then we assume that $f(t, y)$ is continuous on $\mathcal{S}$, and that it satisfied the Lipschitz condition

$$\| f(t, y) - f(t, y^*) \| \le L \| y - y^* \| \quad 0 \le t \le b. \tag{2.10}$$

These assumptions on $f(t, y)$ guarantee the existence of a unique solution to (2.8) and (2.9). The proof of which can be found in [5], and will not be repeated here.

We are now going to approximate the solution for (2.8) and (2.9) by quadratic spline functions. Our construction of the approximation solution is as follows. Let the interval $[0, b]$ be partitioned into m-subintervals: $0 = t_0 < t_1 < \cdots < t_m = b$, and let $h = t_{k+1} - t_k$ for $k = 0, 1, \ldots; m - 1$. Let $S(t)$ be a quadratic spline function, class C^1, and having its grids at the points $t = t_k$, $k = 1, 2, \ldots, m - 1$.

Define $S(t)$ by

$$S(t) = \alpha^k + \beta^k(t - t_k) + \gamma^k(t - t_k)^2, \qquad t_k \le t < t_{k+1}, \tag{2.11}$$

where α^k, β^k, γ^k are unknown vectors of dimension N to be determined.

Since $S(t) \in C^1$, then by spline continuity we have

$$\alpha^{k+1} = \alpha^k + h\beta^k + h^2\gamma^k, \tag{2.12}$$

$$\beta^{k+1} = \beta^k + 2h\gamma^k. \tag{2.13}$$

Multiply (2.13) by h, and then subtract from two times of (2.12) gives us

$$\alpha^{k+1} = \alpha^k + \frac{h}{2}(\beta^k + \beta^{k+1}). \tag{2.14}$$

If we require $S(t)$ to satisfy (2.8) at the grid point $t = t_{k+1}$, that is

$$S'(t_{k+1}) = f(t_{k+1}, S(t_{k+1}))$$

which implies

$$\beta^{k+1} = f(t_{k+1}, \alpha^{k+1}) \tag{2.15}$$

Substitute (2.15) into (2.14), we obtain

$$\alpha^{k+1} = \alpha^k + \frac{h}{2}(\beta^k + f(t_{k+1}, \alpha^{k+1})) = F_h(\alpha^{k+1}) \tag{2.16}$$

One Lipschitz constant for $F_h(t)$ is $Lh/2$, where L is the Lipschitz constant for $f(t, y)$. Hence for $h < 2/L$ we have that $F_h(t)$ is a strong contraction mapping, and (2.16) has a unique fixed point α^{k+1}, which may be found by iteration.

In order that we can determine all α^k, β^k, γ^k by using the equations (2.13), (2.15) and (2.16) we must first determine α^0 and β^0. To do so we require $S(t)$ to satisfy (2.8) and (2.9) at $t = t_0$, which gives us

$$\alpha^0 = y_0 \tag{2.17}$$

$$\beta^0 = f(t_0, y_0) \tag{2.18}$$

It is obvious that the quadratic spline approximation function $S(t)$ is uniquely defined by the above construction. It should also be noted that if the equation (2.8) is linear, that is, $f(t, y) = A(t)\,y(t) + r(t)$, then we can express α^{k+1} explicitly in terms of α^k.

$$\alpha^{k+1} = \left(I - \frac{h}{2}A(t_{k+1})\right)^{-1}\left[\left(I + \frac{h}{2}A(t_k)\right)\alpha^k + \frac{h}{2}\left(r(t_{k+1}) + r(t_k)\right)\right] \tag{2.19}$$

Now we want to show that the above method is convergent. The proof of such theorem require the following lemma.

Lemma 2.2 If

$$\|z_{n+1}\| \le a\|z_n\| + d \quad \text{for} \quad n = 0, 1, \ldots, N-1, \tag{2.20}$$

where a and d are certain nonnegative constants independent of n, then

$$\|z_n\| \le a^n \|z_0\| + \begin{cases} \frac{a^n - 1}{a - 1} & a \ne 1 \\ nd & a = 1 \end{cases} \qquad n = 1, 2, \ldots, N. \tag{2.21}$$

Proof. For $n = 1$, (2.21) is identical with (2.20) and thus true by hypothesis. Assuming the truth of (2.21) for a value $n < N$, then by applying (2.20) we find (if $a \ne 1$)

$$\|z_{n+1}\| \le a\left\{a^n \|z_0\| + \frac{a^n - 1}{a - 1} d\right\} + d$$

$$= a^{n+1} \|z_0\| + \left(a \frac{a^n - 1}{a - 1} + 1\right) d$$

$$= a^{n+1} \|z_0\| + \left(\frac{a^{n+1} - 1}{a - 1}\right) d$$

which is (2.21) with n increased by 1. A similar argument establishes the result if $a = 1$. The statement of the lemma thus follows by induction.

Let $y(t)$ be the exact solution to (2.8) and (2.9) and assume $f(t, y(t)) \in C^2$ on $\mathcal{S}$, then $y \in C^3[0, b]$ and so for $t \in [t_k, t_{k+1}]$ we can expand $y(t)$, $y'(t)$, $y''(t)$ respectively in Taylor's series about $t = t_k$.

$$y(t) = y(t_k) + y'(t_k)(t - t_k) + \frac{y''(t_k)}{2!}(t - t_k)^2 + \frac{y'''(\varepsilon_t)}{3!}(t - t_k)^3, \tag{2.22}$$

$$y'(t) = y'(t_k) + y''(t_k)(t - t_k) + \frac{y'''(\bar{\varepsilon}_t)}{2!}(t - t_k)^2, \tag{2.23}$$

$$y''(t) = y''(t_k) + y'''(\tilde{\varepsilon}_t)(t - t_k), \tag{2.24}$$

where $y'''(\varepsilon_t)$ is the N-dimensional vector with components $y_1'''(\varepsilon_t^1), \ldots, y_N'''(\varepsilon_t^N)$;

$y'''(\bar{\varepsilon}_t)$ is the N-dimensional vector with components $y_1'''(\bar{\varepsilon}_t^1), \ldots, y_N'''(\bar{\varepsilon}_t^N)$;

$y'''(\tilde{\varepsilon}_t)$ is the N-dimensional vector with components $y_1'''(\tilde{\varepsilon}_t^1), \ldots, y_N'''(\tilde{\varepsilon}_t^N)$;

and $t_k < \bar{\varepsilon}_t^i, \varepsilon_t^i, \tilde{\varepsilon}_t^i < t$ for $i = 1,2,\ldots,N$.

Since $\alpha^k = S(t_k)$, $\beta^k = S'(t_k)$, $\gamma^k = S''(t_k)\,|\,2!$, therefore for $t \in [t_k, t_{k+1})$ we can express $S(t)$, $S'(t)$, $S''(t)$ as

(2.25) $$S(t) = S(t_k) + S'(t_k)(t - t_k) + \frac{S''(t_k)}{2!}(t - t_k)^2 ,$$

(2.26) $$S'(t) = S'(t_k) + S''(t_k)(t - t_k) ,$$

(2.27) $$S''(t) = S''(t_k) .$$

Define error vector function $E(t)$ by

$$E(t) = y(t) - S(t)$$

where $t \in [t_k, t_{k+1}]$ for $k = 0, 1, \ldots, m-1$, then we obtain the equation for $E(t)$ by subtracting (2.25) from (2.22), the equation for $E'(t)$ by subtracting (2.26) from (2.23); and the equation for $E''(t)$ by subtracting (2.27) from (2.24):

(2.28) $$E(t) = E(t_k) + E'(t_k)(t-t_k) + \frac{E''(t_k)}{2!}(t-t_k)^2 + \frac{y'''(\varepsilon_t)}{3!}(t-t_k)^3 ,$$

(2.29) $$E'(t) = E'(t_k) + E''(t_k)(t-t_k) + \frac{y'''(\bar{\varepsilon}_t)}{2!}(t-t_k)^2 ,$$

(2.30) $$E''(t) = E''(t_k) + y'''(\tilde{\varepsilon}_t)(t-t_k) .$$

Letting $t = t_{k+1}$ in (2.28) and (2.29) respectively gives us

$$E(t_{k+1}) = E(t_k) + E'(t_k)\,h + \frac{E''(t_k)}{2!}\,h^2 + \frac{y'''(\varepsilon t_{k+1})}{3!}\,h^3 , \tag{2.31}$$

$$E'(t_{k+1}) = E'(t_k) + E''(t_k)\,h + \frac{y'''(\bar{\varepsilon} t_{k+1})}{2!}\,h^2 . \tag{2.32}$$

Since both $y(t)$ and $S(t)$ satisfy (2.8) at the grid point $t = t_{k+1}$,

$$E'(t_{k+1}) = f(t_{k+1}, y(t_{k+1})) - f(t_{k+1}, S(t_{k+1})) , \tag{2.33}$$

which implies

$$\| E'(t_{k+1}) \| \le L \| E(t_{k+1}) \| . \tag{2.34}$$

Multiply equation (2.32) by h and subtract two times of equation (2.31) we have

$$E(t_{k+1}) = E(t_k) + \frac{h}{2}\left(E'(t_{k+1}) + E'(t_k)\right) + \left(\frac{y'''(\varepsilon t_{k+1})}{6} - \frac{y'''(\bar{\varepsilon} t_{k+1})}{4}\right) h^3 . \tag{2.35}$$

By using (2.34) we obtain from (2.35) that

$$\| E(t_{k+1}) \| \le \frac{1 + \frac{h}{2} L}{1 - \frac{h}{2} L} \| E(t_k) \| + \frac{\left\| \frac{y'''(\varepsilon t_{k+1})}{6} - \frac{y'''(\bar{\varepsilon} t_{k+1})}{4} \right\|}{1 - \frac{h}{2} L}\, h^3 . \tag{2.36}$$

Since $\| E(t_0) \| = 0$, by using Lemma 2.2 we can conclude from (2.36) that

$$\| E(t_k) \| = 0(h^2) ,$$

and so from (2.34) and (2.32)

$$\| E'(t_k) \| = 0(h^2) ,$$

$$\| E''(t_k) \| = 0(h) .$$

Since for $k = 0, 1, 2, \ldots, m$, $\|E(t_k)\| = 0(h^2)$, $\|E'(t_k)\| = 0(h^2)$, and $\|E''(t_k)\| = 0(h)$, it is obvious from equations (2.28), (2.29) and (2.30) that for $t \in [0, b]$

$$\|E(t)\| = 0(h^2) ,$$

$$\|E'(t)\| = 0(h^2) ,$$

$$\|E''(t)\| = 0(h) ,$$

and so we have proved the following theorem.

Theorem 2.2. If $f(t, y(t)) \in C^2$ on $\mathcal{S}$, then there exists a constant K such that for $h < 2/L$

$$\|y(t) - S(t)\| < Kh^2 ,$$

$$\|y'(t) - S'(t)\| < Kh^2 ,$$

$$\|y''(t) - S''(t)\| < Kh ,$$

for $t \in [0, b]$, provided $S(t_k)$, $S'(t_k)$, $S''(t_k)$ are given by equations (2.16), (2.15), and (2.13).

3. CONVERGENCE AND ERROR ESTIMATE

Assume that X_s^{n+1} is the quadratic spline approximation to the exact solution X^{n+1} in (2.4), and P_s^{n+1} is the quadratic spline approximation to the exact solution p^{n+1} in (2.5), then it is easy to show that

$$S^{n+1} = X_s^{n+1} \; c^{n+1} + P_s^{n+1} \tag{3.1}$$

is the quadratic spline approximation to x^{n+1}, which is the exact solution of (2.1) and (2.2).

In order to discuss convergence and to give error estimate we assume that there exist solution to the given problem and the approximating linear problems.

Since by Theorem 2.1, the sequence $\{x^n\}$ converges to the solution x of equation (1.1) and (1.2), therefore given $\varepsilon > 0$ there exists a number $N(\varepsilon, x)$ such that for all $n > N(\varepsilon, x)$

(3.2) $$\|x - x^n\| < \varepsilon/2 .$$

Suppose the interval $[0, b]$ was divided into m subintervals, and if $h = b/m$. It is clear that $h \to 0$ if and only if $m \to \infty$. Now let S^n denoted by S^n_m. Since by Theorem 2.2 for any fixed n, S^n_m converges to x^n, so given $\varepsilon > 0$, there exists a number $M(\varepsilon, t)$ such that for all $m > M(\epsilon, t)$

(3.3) $$\|x^n - S^n_m\| < \varepsilon/2.$$

But by triangular inequalities

(3.4) $$\|x - S^n_m\| < \|x - x^n\| + \|x^n - S^n_m\|,$$

so given $\epsilon > 0$,

$$\|x - S^n_m\| < \varepsilon/2 + \varepsilon/2 = \varepsilon$$

for all $n > N(\varepsilon, t)$, and all $m > M(\varepsilon, t)$.

Hence we have proved the following theorem.

<u>Theorem 3.1.</u> If $g(x)$ given in equation (1.1) is such that $g(x) \in C^3 [0, b]$, then the sequence $\{S^n(t)\}$ generated by (3.1) converges to the solution x of equation (1.1) and (1.2) as $n \to \infty$, and $h \to 0$, and if b is sufficiently small.

Let $S(t)$ be the spline approximation to the solution $x(t)$ of (1.1) and (1.2). Since $S(t)$ does not satisfy (1.1) but satisfies (1.2) exactly we have for $t \in [0, b]$

(3.4) $$S'(t) = g(S(t)) - \delta(t),$$

and

(3.5) $$\begin{aligned} (S(0), a_i) &= b_i, \qquad i = 1, 2, \ldots, k, \\ (S(b), a_i) &= b_i, \qquad i = k+1, \ldots, N, \end{aligned}$$

where $\delta(t)$ is the residue.

Since $g(x) \in C^3$, therefore by mean value theorem, we have

(3.6) $$g(x) \simeq g(S) + J(S)(x - S),$$

where J is the Jacobian matrix as defined in (2.3).

Define the error function $e(t)$ by $e(t) = x(t) - S(t)$. Subtracting equation (3.4) from (1.1), and (3.5) from (1.2) respectively, and using (3.6) we have approximately

(3.7) $$e'(t) = J(S(t))\, e(t) + \delta(t), \qquad t \in [0, b],$$

with

(3.8) $$\begin{aligned} (e(0), a_i) &= 0, \qquad i = 1, 2, \ldots, k, \\ (e(b), a_i) &= 0, \qquad i = k+1, \ldots, N. \end{aligned}$$

Since $S(t)$ is known and $\delta(t)$ can be computed from equation (3.4), therefore $e(t)$ can be approximated by exactly the same technique as we approximate the exact solution of the problem with the exception that we have to reduce the step-size h, and use double precision on $\delta(t)$.

Remark: Our numerical procedure considered above produces smooth, accurate, global approximation to the solution of (2.8) and (2.9). In particular, the step-size h can be changed at any step, if necessary, without added complications.

4. A NUMERICAL EXAMPLE

As a test problem let us consider the following problem whose solution is known. This example is also contained in [6]. The authors of which had solved the problem by linear programming technique.

Equation:

$$y''(t) = 2y^3, \qquad t \in [0, 1], \tag{4.1}$$

with boundary conditions

$$\begin{aligned} y(0) + y'(0) &= 0, \\ y(1) &= 0.5. \end{aligned} \tag{4.2}$$

The exact solution is: $y(t) = (1 + t)^{-1}$.

To obtain the solution of this problem we first convert (4.1) and (4.2) into the form of (1.1) and (1.2), so that our method can be applied. Then for each step-size $h = 1./64.$, $h = 1./128.$, $h = 1./256.$, we take $S_0(t) = 1 - 0.5t$ as initial guess and compute $S_1(t)$, $S_2(t)$ and $S_3(t)$. The numerical results are summerized in the following tables.

TABLE 4.1 (h = 1./64.)

t	$S_0(t)$	$S_1(t)$	$S_2(t)$	$S_3(t)$	y(t)	$y(t) - S_3(t)$	ERROR ESTIMATE
0.	1.0000000	1.0056713	1.0000133	0.9999763	1.0000000	2.3×10^{-5}	2.2×10^{-5}
0.125	0.9375000	0.8940661	0.8889059	0.8888580	0.8888889	3.1×10^{-5}	3.0×10^{-5}
0.250	0.8750000	0.8049158	0.7999967	0.7999684	0.8000000	3.1×10^{-5}	3.1×10^{-5}
0.375	0.8125000	0.7318516	0.7272676	0.7272437	0.7272727	2.9×10^{-5}	2.8×10^{-5}
0.500	0.7500000	0.6706739	0.6666613	0.6666419	0.6666667	2.5×10^{-5}	2.4×10^{-5}
0.625	0.6875000	0.6185707	0.6153797	0.6153652	0.6153846	1.9×10^{-5}	1.9×10^{-5}
0.750	0.6250000	0.5736145	0.5714247	0.5714151	0.5714286	1.3×10^{-5}	1.3×10^{-5}
0.875	0.5625000	0.5344343	0.5333312	0.5333264	0.5333333	6.9×10^{-6}	6.6×10^{-6}
1.000	0.5000000	0.5000000	0.5000000	0.5000000	0.5000000	0	0

TABLE 4.2 ($h = 1./128.$)

t	$S_0(t)$	$S_1(t)$	$S_2(t)$	$S_3(t)$	$y(t)$	$y(t) - S_3(t)$
0.	1.0000000	1.0056848	1.0000311	0.9999941	1.0000000	5.9×10^{-6}
0.125	0.9375000	0.8940858	0.8889137	0.8888117	0.8888889	7.7×10^{-6}
0.250	0.8750000	0.8049369	0.8000204	0.7999921	0.8000000	7.9×10^{-6}
0.375	0.8125000	0.7318713	0.7272894	0.7272655	0.7272727	7.3×10^{-6}
0.500	0.7500000	0.6706908	0.6666798	0.6666605	0.6666667	6.2×10^{-6}
0.625	0.6875000	0.6185839	0.6153943	0.6153798	0.6153846	4.8×10^{-6}
0.750	0.6250000	0.5736236	0.5714348	0.5714252	0.5714286	3.4×10^{-6}
0.875	0.5625000	0.5344389	0.5333363	0.5333316	0.5333333	1.7×10^{-6}
1.000	0.5000000	0.5000000	0.5000000	0.5000000	0.5000000	0

TABLE 4.3 (h = 1./256.)

t	$S_0(t)$	$S_1(t)$	$S_2(t)$	$S_3(t)$	$y(t)$	$y(t) - S_3(t)$
0.	1.0000000	1.0056882	1.0000355	0.9999985	1.0000000	1.5×10^{-6}
0.125	0.9375000	0.8940908	0.8889195	0.8888869	0.8888889	1.9×10^{-6}
0.250	0.8750000	0.8049421	0.8000263	0.7999980	0.8000000	2.0×10^{-6}
0.375	0.8125000	0.7318762	0.7272948	0.7272709	0.7272727	1.8×10^{-6}
0.500	0.7500000	0.6706950	0.6666844	0.6666651	0.6666667	1.5×10^{-6}
0.625	0.6875000	0.6185873	0.6153979	0.6153834	0.6153846	1.2×10^{-6}
0.750	0.6250000	0.5736258	0.5714373	0.5714277	0.5714286	8.4×10^{-7}
0.875	0.5625000	0.5344401	0.5333377	0.5333329	0.5333333	4.3×10^{-7}
1.000	0.5000000	0.5000000	0.5000000	0.5000000	0.5000000	0

5. APPLICATION TO THE HENCKY PROBLEM

5.1 Statement of the problem

In this section we are going to consider the following problem

(5.1) $$x''(t) + \frac{3}{t} x'(t) + \frac{2}{x(t)^2} = 0 , \qquad t \in [0, 1],$$

subject to

(5.2) $$x'(0) = 0 , \qquad x(1) = \lambda,$$

where λ is any constant greater than 0 .

Existence and uniqueness of the solution to this problem is discussed recently in [7] and [8]. Numerical solution is also presented in [7] for $\lambda \geq 0.4$. Using our numerical procedure offered no difficulty in solving the Hencky problem for a wide range of values of λ . Before we present the numerical results, it is important to make a brief analysis on how we approach the solution of this problem.

5.2. Analysis

(A). Conversion of (5.1) and (5.2) to the form of (1.1) and (1.2), so that our method can be applied.

Let $x_1 = x$, $x_2 = x'$, then (5.1) becomes

(5.3) $$x_1'(t) = g_1(x_1, x_2) , \qquad x_2'(t) = g_2(x_1, x_2) ,$$

where

$$g_1(x_1, x_2) = x_2$$

$$g_2(x_1, x_2) = -\frac{3x_2}{t} - \frac{2}{x_1^2} ,$$

with boundary conditions

(5.4) $$x_2(0) = 0 ,$$
$$x_1(1) = \lambda .$$

(B). Since the equation (5.1) has a singularity at $t = 0$, approximation of $g_2(x_1(t), x_2(t))$ at $t = 0$ is necessary in the procedure.

Since the equivalent non-linear integral equation to (5.1) is

(5.5) $$x(t) = \lambda - \int_0^1 \frac{\tau^3 \, d\tau}{x^2(\tau)} + \frac{1}{t^2} \int_0^t \frac{\tau^3}{x^2(\tau)} \, d\tau + \int_t^1 \frac{\tau \, d\tau}{x^2(\tau)} ,$$

taking derivative on both sides of (5.5) gives us

(5.6) $$x'(t) = -\frac{2}{t^3} \int_0^t \frac{\tau^3}{x^2(\tau)} \, d\tau$$

and so

(5.7) $$\frac{3}{t} x'(t) = -\frac{6}{t^4} \int_0^t \frac{\tau^3}{x^2(\tau)} \, d\tau$$

As $t \to 0$ equation (5.7) implies

$$\frac{3}{t} x'(t) \simeq -\frac{3}{2} \frac{1}{x^2(0)}$$

Hence at $t = 0$, we can approximate $g_2(x_1(t), x_2(t))$ by

$$g_2(x_1(0),\ x_2(0)) = \frac{3}{2}\ \frac{1}{x_1^2(0)} - \frac{2}{x_1^2(0)}$$

$$= -0.5/x_1^2(0)$$

(C). Choice of the initial approximation.

Based on the informations supplied in [7], we guess that $x_1(0)$ will be in the neighborhood of 1. On the other hand $x_1(1) = \lambda$, therefore for a given λ we let

$$x_1(t) = 1 + (\lambda - 1)t$$

$$x_2(t) = \lambda$$

be the initial approximation.

(D). Choice of the step-size h.

The step-size h should be chosen very small at both ends of the interval $[0, 1]$, because

(i) We approximate $\frac{3}{t} x_2(t)$ by $-\frac{3}{2}\ \frac{1}{x_1^2(0)}$ at $t = 0$.

(ii) The slope of $x_1(t)$ becomes very large as $t \to 1$, especially for λ small.

In the method proposed there will be no added complication to such variation of h.

5.3 <u>Numerical results</u>.

In this section we are going to include results for $\lambda = 1,\ 0.5,\ 0.1,\ 0.05$, and 0.01. For each λ, numerical solution $S(t)$, along with its first two derivatives, residue, and error estimate are tabulated in Table 5.1

to Table 5.5. In Figure 5.1 the solution $S(t)$ is shown for various values of λ .

TABLE 5.1 ($\lambda = 1$)

t	S(t)	S'(t)	S''(t)	RESIDUE	ERROR ESTIMATE
0.	1.19446366	0	$-3.5044843 \times 10^{-1}$	0	-1.7×10^{-6}
0.125	1.19170709	$-4.3941594 \times 10^{-2}$	$-3.5368802 \times 10^{-1}$	-1.1×10^{-10}	2.7×10^{-7}
0.250	1.18342981	$-8.8701592 \times 10^{-2}$	$-3.6363606 \times 10^{-1}$	1.2×10^{-10}	2.9×10^{-7}
0.375	1.16946091	$-1.3516209 \times 10^{-1}$	$-3.8107748 \times 10^{-1}$	-1.4×10^{-10}	3.2×10^{-7}
0.500	1.14988402	$-1.8365535 \times 10^{-1}$	$-4.1066026 \times 10^{-1}$	-1.2×10^{-10}	3.1×10^{-7}
0.625	1.12363613	$-2.3805143 \times 10^{-1}$	$-4.4623857 \times 10^{-1}$	2.3×10^{-10}	1.6×10^{-7}
0.750	1.09039433	$-2.9590454 \times 10^{-1}$	$-4.9852451 \times 10^{-1}$	1.4×10^{-10}	1.1×10^{-7}
0.875	1.04968293	$-3.6215691 \times 10^{-1}$	$-5.7347420 \times 10^{-1}$	-8.7×10^{-11}	5.1×10^{-8}
1.000	1.00000000	$-4.3968193 \times 10^{-1}$	$-6.8095420 \times 10^{-1}$	1.7×10^{-10}	0

TABLE 5.2 ($\lambda = 0.5$)

t	S(t)	S'(t)	S''(t)	RESIDUE	ERROR ESTIMATE
0.	0.91009787	0	$-6.0366195 \times 10^{-1}$	0	-3.4×10^{-6}
0.125	0.90534025	$-7.5987929 \times 10^{-2}$	$-6.1638246 \times 10^{-1}$	0	4.7×10^{-9}
0.250	0.89094072	$-1.5524117 \times 10^{-1}$	$-6.5671033 \times 10^{-1}$	-5.8×10^{-11}	4.5×10^{-8}
0.375	0.86623457	$-2.4162653 \times 10^{-1}$	$-7.3236677 \times 10^{-1}$	-1.1×10^{-10}	1.4×10^{-7}
0.500	0.8306789	$-3.3900278 \times 10^{-1}$	$-8.6441823 \times 10^{-1}$	-2.3×10^{-10}	1.9×10^{-7}
0.625	0.78107808	$-4.5897684 \times 10^{-1}$	-1.0751537×10^{0}	-1.7×10^{-10}	-6.0×10^{-9}
0.750	0.71446082	$-6.1482155 \times 10^{-1}$	-1.4587926×10^{0}	-2.9×10^{-10}	-1.7×10^{-8}
0.875	0.62536066	-8.3795389×10^{0}	-2.2411112×10^{0}	2.3×10^{-10}	-1.0×10^{-8}
1.000	0.50000000	-1.22168492	-4.3349452×10^{0}	2.7×10^{-9}	0

TABLE 5.3 ($\lambda = 0.1$)

t	S(t)	S'(t)	S''(t)	RESIDUE	ERROR ESTIMATE
0.	0.81265103	0	$-7.5711495 \times 10^{-1}$	0	-5.3×10^{-6}
0.125	0.80667547	$-9.5576559 \times 10^{-2}$	$-7.7965600 \times 10^{-1}$	-2.9×10^{-10}	-1.0×10^{-6}
0.250	0.78848365	$-1.9700906 \times 10^{-1}$	$-8.5284337 \times 10^{-1}$	5.8×10^{-11}	-9.9×10^{-7}
0.375	0.75687475	$-3.1172689 \times 10^{-1}$	$-9.9744300 \times 10^{-1}$	1.7×10^{-10}	-8.9×10^{-7}
0.500	0.71040122	$-4.4917560 \times 10^{-1}$	-1.2679323×10^{0}	3.4×10^{-10}	-8.4×10^{-7}
0.625	0.64324990	$-6.3582519 \times 10^{-1}$	-1.7816359×10^{0}	2.3×10^{-10}	-1.1×10^{-6}
0.750	0.54743183	$-9.2183666 \times 10^{-1}$	-2.9864027×10^{0}	-3.5×10^{-10}	-1.2×10^{-6}
0.875	0.40309275	-1.4883380×10^{0}	-7.2060488×10^{0}	6.0×10^{-9}	-1.2×10^{-6}
1.000	0.10000000	-5.2039840×10^{0}	-1.8438805×10^{2}	-1.3×10^{-7}	0

TABLE 5.4 ($\lambda = 0.05$)

t	S(t)	S'(t)	S''(t)	RESIDUE	ERROR ESTIMATE
0.	0.80838081	0	$-7.6513485 \times 10^{-1}$	0	-5.7×10^{-6}
0.125	0.80234152	$-9.6604246 \times 10^{-2}$	$-7.8828509 \times 10^{-1}$	0	-1.4×10^{-6}
0.250	0.78394953	$-1.9922673 \times 10^{-1}$	$-8.6355071 \times 10^{-1}$	-1.1×10^{-10}	-1.4×10^{-6}
0.375	0.75197005	$-3.1553202 \times 10^{-1}$	-1.0126936×10^{0}	-3.4×10^{-10}	-1.3×10^{-6}
0.500	0.70489319	$-4.5537441 \times 10^{-1}$	-1.2929155×10^{0}	-3.4×10^{-10}	-1.3×10^{-6}
0.625	0.63672441	$-6.4639431 \times 10^{-1}$	-1.8304876×10^{0}	5.8×10^{-10}	-1.7×10^{-6}
0.750	0.53906376	$-9.4231172 \times 10^{-1}$	-3.1133085×10^{0}	-5.8×10^{-10}	-1.9×10^{-6}
0.850	0.39056222	-1.5440187×10^{0}	-7.8176349×10^{0}	7.4×10^{-9}	-2.2×10^{-6}
1.000	0.05000000	-8.0695549×10^{0}	-7.7579132×10^{2}	-2.0×10^{-6}	0

TABLE 5.5 ($\lambda = 0.01$)

t	S(t)	S'(t)	S''(t)	RESIDUE	ERROR ESTIMATE
0.	0.80642657	0	$-7.6884774 \times 10^{-1}$	0	-1.1×10^{-5}
0.125	0.80035771	$-9.7081616 \times 10^{-2}$	$-7.92283388 \times 10^{-1}$	0	-6.1×10^{-6}
0.250	0.78187301	$-2.0025465 \times 10^{-1}$	$-8.6852418 \times 10^{-1}$	0	-6.3×10^{-6}
0.375	0.74972162	$-3.17298788 \times 10^{-1}$	-1.0198061×10^{0}	4.1×10^{-10}	-6.6×10^{-6}
0.500	0.70236422	$-4.5826082 \times 10^{-1}$	-1.3046352×10^{0}	-2.3×10^{-10}	-7.3×10^{-6}
0.625	0.63372043	$-6.5134098 \times 10^{-1}$	-1.8536233×10^{0}	6.9×10^{-10}	-8.8×10^{-6}
0.750	0.53519309	$-9.5199336 \times 10^{-1}$	-3.1744950×10^{0}	-1.0×10^{-9}	-1.1×10^{-5}
0.875	0.38469596	-1.5711241×10^{0}	-8.1276254×10^{0}	-3.0×10^{-9}	-1.6×10^{-5}
1.000	0.01000000	$-1.98236408 \times 10^{1}$	-1.9940530×10^{4}	-3.2×10^{-3}	0

Computational results give us a set of 5 points $(\lambda, x(0))$. We use polynomial curve fitting by least square method and obtain

$$P(\lambda) = 0.3607048\lambda^2 + 0.0275942\lambda + 0.8061575$$

The results are tabulated in Table 5.6, and the curve is plotted in Figure 5.2

TABLE 5.6

λ	S(0)	$P(\lambda) - S(0)$
1.00	1.19446366	-7.1×10^{-6}
0.50	0.91009787	3.2×10^{-5}
0.10	0.81265103	-1.2×10^{-4}
0.05	0.80838084	5.8×10^{-5}
0.01	0.80642657	4.3×10^{-5}

It is interesting to note that as $\lambda \to 0$, $x(0)$ can be approximated by 0.8061575.

S(t) is the numerical solution of

$$x''(t) = -\frac{3}{t}x'(t) - \frac{2}{x(t)^2}$$

$$x'(0) = 0$$

$$x(1) = \lambda \ (\lambda = 1.0, 0.5, 0.1, 0.05, 0.01)$$

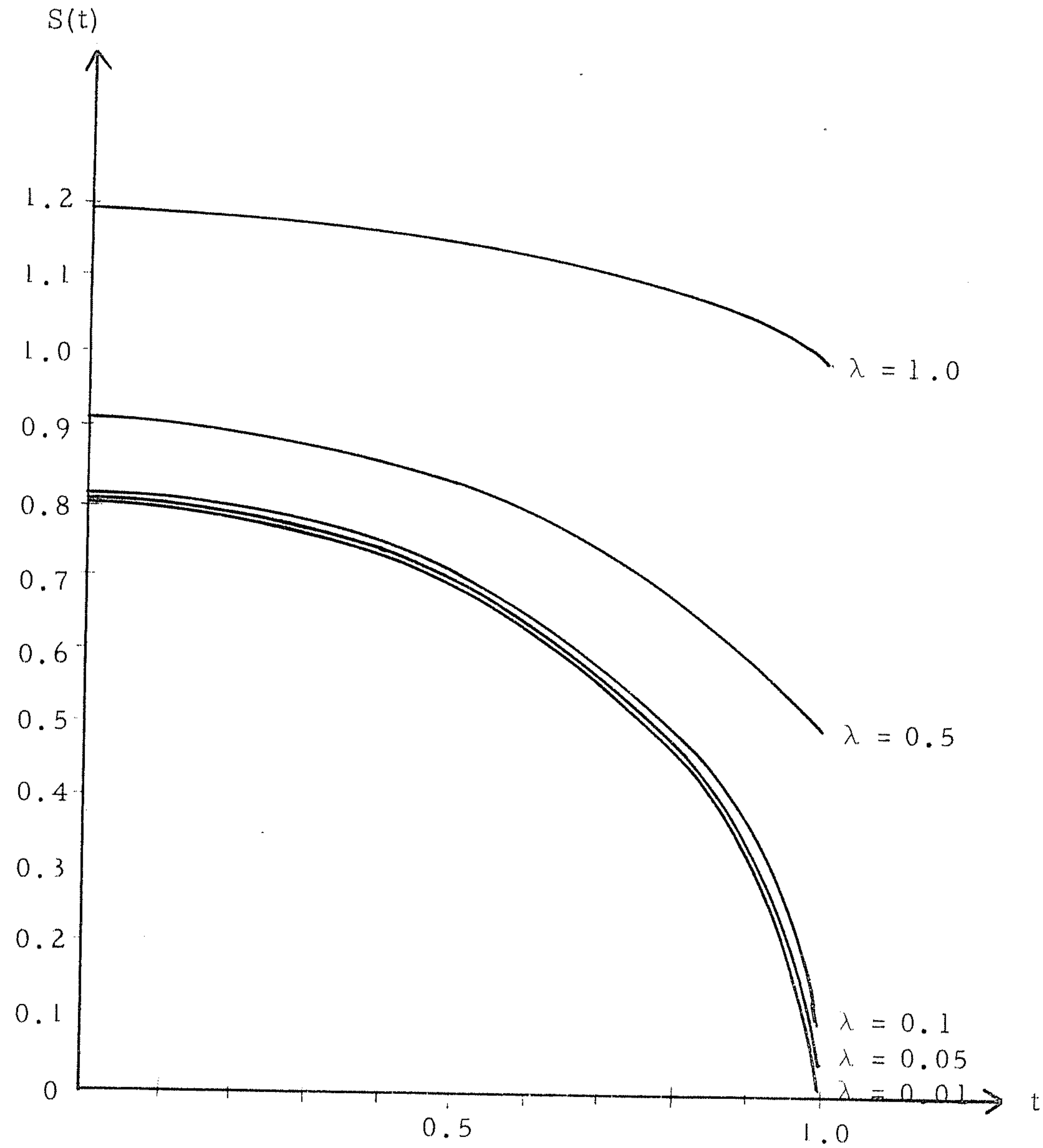

FIGURE 5.1

Polynomial Curve Fiting (By Least Square Method)

$$p(\lambda) = 0.3607048\,\lambda^2 + 0.0275942\lambda + 0.8061575$$

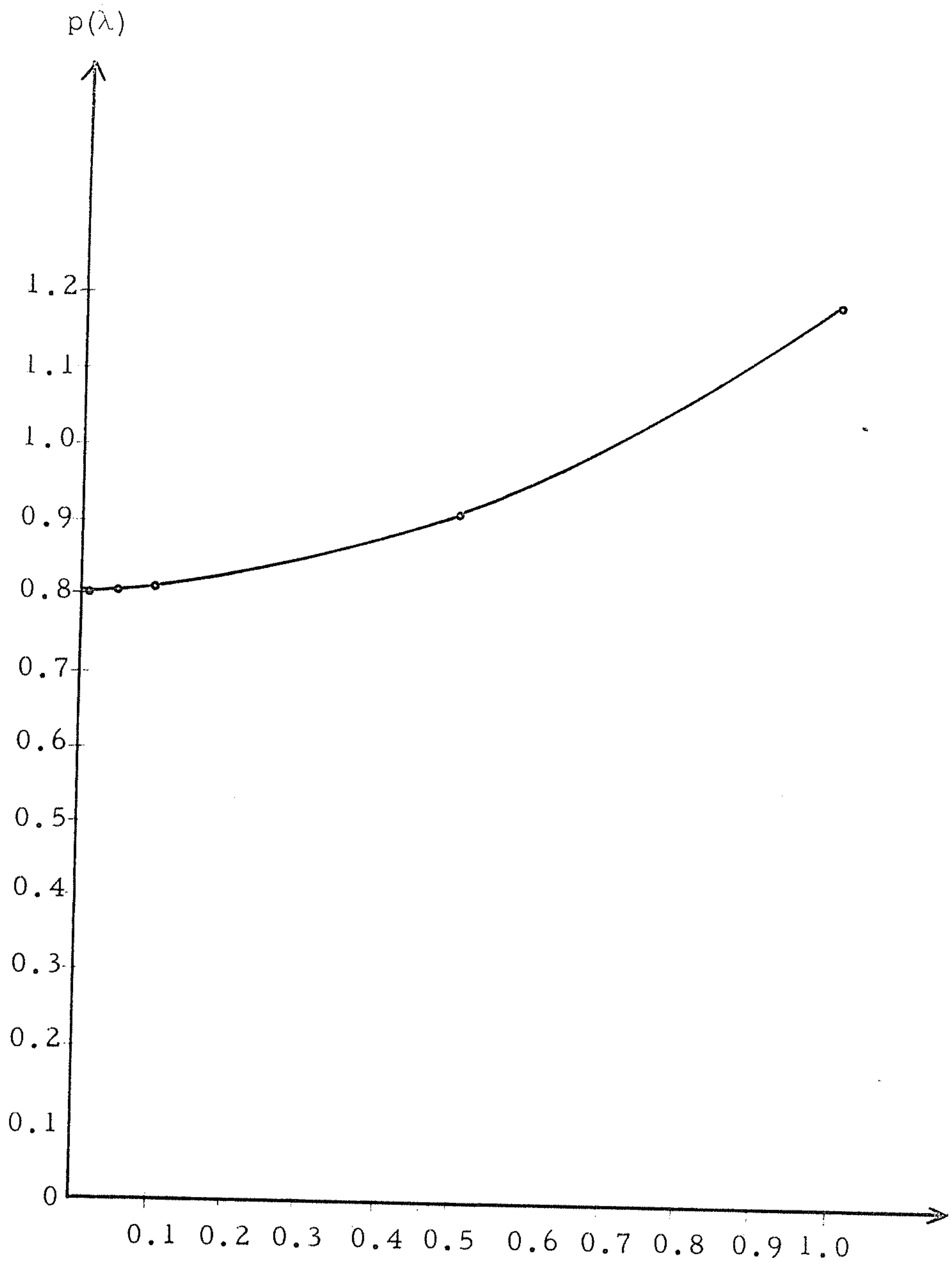

FIGURE 5.2

ACKNOWLEDGMENT

The author is indebted to Professor J. B. Rosen for his instruction and guidance. Gratitude is also expressed to T. Y. Cheung for many valuable discussions during the preparation of this report.

REFERENCES

[1] Bellman, R. E. and Kalaba, R. E., Quasilinearization and Nonlinear Boundary-value Problem, American Elsevier Publishing Co., Inc., New York, 1965.

[2] Kalaba, R. E., "On Nonlinear Differential Equations, The Maximum Operator, and Monotone Convergence". J. Math. Mech., 8 (1959) pp. 519-574.

[3] Loscalzo, F. and Talbot, T., "Spline Function Approximations for Solutions of Ordinary Differential Equations." J. SIAM Numer. Anal. Vol. 4. No. 3, 1967.

[4] Loscalzo, F. and Schoenberg, I. J., "The Use of Spline Functions for the Approximation of Solutions of Ordinary Differential Equations." MRC Technical Summary Report #723, 1967.

[5] Henrici, P., Discrete Variable Methods in Ordinary Differential Equations, John Wiley, New York, 1962.

[6] Rosen, J. B. and Meyer, R., "Solution of Nonlinear Two-points Boundary-value Problems by Linear Programming." Comp. Sci. Technical Report #1, University of Wisconsin, Jan. 1967.

[7] Dickey, R. W., "The Plane Circular Elastic Surface under Normal Pressure in Achive for Rational Mechanics and Analysis." Vol. 26, No. 3, 1967, pp. 219-236.

[8] Callegari, A. and Reiss, E., "Nonlinear Boundary Value Problem for the Circular Membrane." Technical Report. Feb. 1968, Courant Institute of Mathematical Sciences, New York University, New York.