TRANSACTIONS, ILL. STATE ACAD. SCI.
Vol. 71, No. 2 (1978)

NEW NUMERICAL METHODS FOR SYMMETRIC DIFFERENTIAL EQUATIONS, QUADRATIC EXTREMAL PROBLEMS, AND BANDED MATRICES; THE SECOND ORDER PROBLEM

John Gregory and Ralph Wilkerson
Department of Mathematics
Southern Illinois University
Carbondale, Illinois 62901

ABSTRACT

The main purpose of this paper is to give new methods and ideas for the numerical solution of the differential equation $L(x) = (rx')' + px = 0$ or equivalently the quadratic form $J(x) = \int_a^b [rx'^2 - px^2]dt$. Numerical algorithms are given which are accurate, fast, and very easy to implement. Our methods allow both initial value problems and boundary value problems. These methods follow from a broad mathematical theory, and yield strong convergence properties and ideas which generalized to more difficult problems. The problems are often identified by the terms Raleigh-Ritz and finite element method.

INTRODUCTION

In this paper we will present new numerical algorithms, methods and ideas for second order symmetric differential equations, quadratic extremal problems in the calculus of variations, and banded symmetric matrices. Test runs (including some on hand calculators) indicate that our methods are often faster, more accurate, and require less storage than more usual methods even when these methods are available.

These results will "immediately" generalize to new numerical methods for higher order symmetric differential equations, partial differential equations, and eigenvalue problems. Thus we will quantitatively solve many problems of oscillation. The proofs of these results are very technical. They depend upon an approximation theory of quadratic forms given by the first author [2] and will appear separately in another journal. Thus the presentation in this paper will be solely to focus on numerical algorithms. The reader should consult [2] to consider the mathematical problems and related ideas which can be solved but which we do not discuss here.

The basic problem is to find numerical solutions of $L(x) = 0$ or equivalently numerical extremal solutions to the quadratic form problem $J(x)$ where

(1) $L(x) = (r(t)x'(t))' + p(t)x(t) = 0$, $x(a) = 0$ and

(2) $J(x) = \int_a^b [r(t)x'^2(t) - p(t)x^2(t)]dt.$

The problems are equivalent in that $L(x) = 0$ is the Euler-Lagrange equation for the quadratic form $J(x)$. We require $r(t) > 0$ on $[a, b]$ and assume $r(t)$ and $p(t)$ are continuous although weaker conditions on the coefficients $r(t)$ and $p(t)$ will suffice.

In Section II we construct the numerical approximation $J(x; \sigma)$ of $J(x)$. This is a finite dimensional quadratic form

$$J(x; \sigma) = d^T D(\sigma) d$$

where $D(\sigma)$ is a symmetric tridiagonal matrix and $d = (d_1, d_2, \ldots)^T$ is the coefficient vector of the piecewise linear function $x(t)$ which is a line segment between P_k and P_{k+1} where $P_k = (a + k\sigma, d_k)$ $(k=1,2,3,\ldots)$. In Section III we give the algorithm which defines a vector $c = (c_1, c_2, \ldots)$ and an extremal solution $x_0(t)$ which is both the "Euler-Lagrange solution" to $D(\sigma)$ and the numerical solution of $x_0(t)$ where $x_0(t)$ is a nontrivial solution of $L(x) = 0$, $x(a) = 0$; in the sense that under proper normalization, i.e., $x_\sigma'(a) = x_0'(a)$ we have

$$\lim_{\sigma \to 0} \int_a^b [x_\sigma'(t) - x_0'(t)]^2 dt = 0.$$

This last convergence result is very strong for this type of problem. Furthermore, because integration is a smoothing process (the elements of $D(\sigma)$ are constructed this way) σ may be chosen relatively large. In Section IV we give two new numerical methods for problems of this type. The first is a dynamic method similar to conventional methods of solving initial value problems in differential equations. The second is a relaxation method for finding solutions to boundary value problems. In Section V, we give some numerical examples of our algorithms. Finally in the appendix we give program listings and results of test runs.

THE APPROXIMATING QUADRATIC FORM

In this section we construct the approximating quadratic form $J(x; \sigma) = x^T D(\sigma) x$ and in particular, the elements $e_{\alpha,\beta}$ of $D(\sigma)$.

Let σ be small and positive and choose a partition of the interval $[a, b]$ namely $\pi(\sigma) = (a_0 = a < a_1 < a_2 < \ldots < a_{N+1} \leq b)$ where $a_k = a + k\sigma$ $(k = 0,1,2,3,\ldots,N)$. For convenience we assume $a_{N+1} = b$. For each k let $z_k(t)$ be the spline hat function of degree 1, namely

$$(3) \qquad z_k(t) = \begin{cases} 1 - |t - a_k|/\sigma & \text{if } t \text{ in } [a_{k-1}, a_{k+1}] \\ 0 & \text{otherwise} \end{cases}$$

We note that $\{z_1, \ldots, z_N\}$ is a basis for the vector space of piecewise

linear functions $x_\sigma(t)$ which vanish at $t = a$ and $t = b$; we denote this space by $\mathcal{O}(\sigma)$ to conform to Reference [2].

To define the approximating quadratic form $J(x; \sigma)$ we choose $r_\sigma(t) = r(a_k^*)$ and $p_\sigma(t) = p(a_k^*)$ if $a_k \le t < a_{k+1}$, where $a_k^* = a_k + \sigma/2$. Then

$$J(x; \sigma) = \int_a^b [r_\sigma(t)x'^2(t) - p_\sigma(t)x^2(t)]dt \tag{4}$$

defines a quadratic form on the space $\mathcal{O}(\sigma)$. Choosing $x(t) = b_\alpha z_\alpha(t)$ (repeated indices are summed) we have $J(x; \sigma) = J(b_\alpha z_\alpha, b_\beta z_\beta; \sigma) = b_\alpha b_\beta J(z_\alpha, z_\beta; \sigma) = b_\alpha b_\beta e_{\alpha\beta}$.

THEOREM 1. $J(x; \sigma)$ defined by (4) on the space $\mathcal{O}(\sigma)$ is a quadratic form whose associated matrix $D(\sigma) = (e_{\alpha\beta})$ is symmetric and tridiagonal where $e_{\alpha,\beta}$ is defined below.

To construct $e_{\alpha\beta}$ we note that if $|\alpha - \beta| \ge 2$ we have $e_{\alpha\beta} = 0$ since $z_k(t)$ vanishes off the interval (a_{k-1}, a_{k+1}). Thus

$$\begin{aligned}
e_{k,k} &= \int_{a_{k-1}}^{a_{k+1}} [r_\sigma(t)z_k'^2(t) - p_\sigma(t)z_k^2(t)]dt \\
&= \int_{a_{k-1}}^{a_k} [r_\sigma(t)z_k'^2(t) - p_\sigma(t)z_k^2(t)]dt + \\
&\qquad + \int_{a_k}^{a_{k+1}} [r_\sigma(t)z_k'^2(t) - p_\sigma(t)z_k^2(t)]dt \\
&= r(a_{k-1}^*) \int_{a_{k-1}}^{a_k} \left(\frac{1}{\sigma}\right)^2 dt - \frac{p(a_{k-1}^*)}{\sigma^2} \int_{a_{k-1}}^{a_k} (t - a_{k-1})^2 dt \\
&\quad + r(a_k^*) \int_{a_k}^{a_{k+1}} \left(-\frac{1}{\sigma}\right)^2 dt - \frac{p(a_k^*)}{\sigma^2} \int_{a_k}^{a_{k+1}} (a_{k+1} - t)^2 dt \\
&= \sigma r(a_{k-1}^*)/\sigma^2 - \frac{p(a_{k-1}^*)}{\sigma^2} \left(\frac{1}{3}\right)(t - a_{k-1})^3 \Big|_{a_{k-1}}^{a_k} \\
&\quad + \sigma r(a_k^*)/\sigma^2 - \frac{p(a_k^*)}{\sigma^2} \left(\frac{1}{3}\right)(a_{k+1} - t)^3 \Big|_{a_k}^{a_{k+1}} \\
&= [r(a_{k-1}^*) + r(a_k^*)]/\sigma - \frac{\sigma}{3}[p(a_{k-1}^*) + p(a_k^*)]
\end{aligned}$$

and

$$e_{k,k+1} = \int_{a_{k-1}}^{a_{k+2}} [r_\sigma(t)z_k'(t)z_{k+1}'(t) - p_\sigma(t)z_k(t)z_{k+1}(t)]dt$$

$$= \int_{a_{k-1}}^{a_k} [r_\sigma(t)z_k'(t)z_{k+1}'(t) - p_\sigma(t)z_k(t)z_{k+1}(t)]dt$$

$$+ \int_{a_k}^{a_{k+1}} [r_\sigma(t)z_k'(t)z_{k+1}'(t) - p_\sigma(t)z_k(t)z_{k+1}(t)]dt$$

$$+ \int_{a_{k+1}}^{a_{k+2}} [r_\sigma(t)z_k'(t)z_{k+1}'(t) - p_\sigma(t)z_k(t)z_{k+1}(t)]dt$$

$$= \int_{a_k}^{a_{k+1}} r(a_k^*)\left(-\frac{1}{\sigma}\right)\left(\frac{1}{\sigma}\right)dt - \int_{a_k}^{a_{k+1}} p(a_k^*)\left[\frac{1}{\sigma}(a_{k+1} - t)\right]\left[\frac{1}{\sigma}(t - a_k)\right]dt$$

$$= -\frac{r(a_k^*)}{\sigma} - \frac{1}{6}\, p(a_k^*)\sigma.$$

In practice it is more convenient to normalize our problem so that

(5a) $$e_{k,k} = r(a_{k-1}^*) + r(a_k^*) - \frac{\sigma^2}{3}\,[p(a_{k-1}^*) + p(a_k^*)] \quad \text{and}$$

(5b) $$e_{k,k+1} = -r(a_k^*) - \frac{\sigma^2}{6}\, p(a_k^*).$$

This is due to the fact that $J(x; \sigma)$ is quadratic, hence homogeneous of degree two.

THE ALGORITHM

The motivation for our work is that we wish to construct the "Euler Lagrange equations" for $D(\sigma)$. In the continuous case the bilinear form for (2) is

$$J(x,y) = \int_a^b (rx'y' - pxy)\, dt = -\int_a^b [(rx')' + px]y\, dt + r(t)x'(t)y(t)\Big|_a^b$$

Hence in trying to make $J(x, y) = 0$ for all $y(t)$ vanishing at $t = a$ and $t = b$ we set $L(x) = (rx')' + px = 0$ and "wait" until $y(b) = 0$. The discrete analogue is to try to find $\hat{c} = (c_1, c_2, \ldots,)^T$ so that $J(x, y; \sigma) = \hat{c}^T D(\sigma)\hat{c} = 0$ when the function $\eta(t) = d_\alpha z_\alpha(t)$ becomes zero (where repeated indices are summed). Thus we try to solve $D(\sigma)\hat{c} = 0$ and "wait" until $\eta(t)$ becomes zero. This cannot be done, but we come "as close as we can" if we define the sequence $\{c_i\}$ $(i = 1, \ldots, N)$ by

(6a) $$c_1 e_{11} + c_2 e_{12} = 0$$

(6b) $$c_1 e_{21} + c_2 e_{22} + c_3 e_{23} = 0$$

(6c) $\quad c_{k-1}e_{k,k-1} + c_k e_{k,k} + c_{k+1}e_{k,k+1} = 0 \quad (k = 3,4,5,\ldots,N)$

and the extremal solution $x_\sigma(t) = c_\alpha z_\alpha(t)$. Notice that (6) gives a sequence $\{c_k\}$ such that "$D(\sigma)c = 0$ except for the last term."

THEOREM 2. *Define $e_{k,k}$ and $c_{k,k+1} = e_{k+1,k}$ as in (5) and $\{c_i\}$ by (6). Let $x_0(t)$ be a nonzero solution to $L(x) = 0$, $x_0(a) = 0$ and $x_\sigma(t) = c_\alpha z_\alpha(t)$, where c_1 is chosen so that $x_0'(a) = x_\sigma'(a)$, that is $c_1 = x_0(a + \sigma)$. Then*

$$\lim_{\sigma \to 0} \int_a^b (x_0'(t) - x_\sigma'(t))^2 dt = 0. \tag{7}$$

The proof of this result is very technical and "breaks new ground" in the theory of approximation in that we are approximating the nonpositive part of an operator [2]. Furthermore σ may be chosen relatively large if $r(t)$ and $p(t)$ are smooth because our methods depend on construction of negative vectors, i.e., eigenvectors corresponding to negative eigenvalues. Intuitively we find roots by looking at changes of sign instead of a relatively unstable method for finding zeros. In actual fact the reader may verify that if n_j is the j^{th} value of k for which $a_k a_{k+1} \leq 0$ the vectors $\tilde{c}_1 = (c_1, c_2, \ldots, c_{n_1}, 0, 0, \ldots)^T$ and $\tilde{c}_j = (0, 0, \ldots, 0, c_{n_j+1}, c_{n_j+2}, \ldots, c_{n_{j+1}}, 0, 0, \ldots)^T$ for $j = 2, 3, \ldots$ satisfy $\tilde{c}_j^T D(\sigma)\tilde{c}_j \leq 0$.

As an aside we present another new result for the "Elliptic type" matrix $D(\sigma)$ given above. Perhaps more significantly we should note that our results can be generalized to more general banded matrices of this type.

Theorem 3 generalizes the well known result that a symmetric matrix $A_{n\times n} = (a_{ij})$ is positive definite if and only if the determinants p_1, p_2, and p_n are positive where p_k is the determinant of the principal minor

$$A_p = (a_{ij}) \qquad (i, j = 1, \ldots, p).$$

For these results assume p_r is as described above, that is, let $p_0 = 1$, $p_1 = e_{11}$ and $p_r = e_{r,r}p_{r-1} - e_{r,r-1}^2 p_{r-2}$ $(r = 2, 3, 4, \ldots)$.

THEOREM 3. *The number of agreements in sign of two successive members of the sequence $\{p_r\}$ is equal to the number of eigenvalues of $D(\sigma)$ which are greater than zero. Similarly if $n_1, n_2, \ldots, n_j, \ldots$ and $\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_j, \ldots$ are the positive integers and vectors described at the end of the last section, then the n_j^{th} principal minor of $D(\sigma)$ has j eigenvalues less than or equal to zero and c_j is an "approximate" zero eigenvector of $D(\sigma)$ in the sense described above.*

To find the eigenvalues and eigenvectors of $D(\sigma)$ we construct the new tridiagonal matrix $D_\lambda(\sigma) = D(\sigma) - \lambda I$ where λ is any real number and apply Theorem 3 to this new problem. Note that this procedure gives a

great deal of information for numerical problems. It is evident that with the speed of modern day computing machines that these methods are both accurate and extremely fast. The results will be given in a later paper.

TWO METHODS OF COMPUTATION

In this section we will describe a "dynamic" algorithm for initial value problems and a "relaxed" method for boundary value problems. The former method is similar to conventional methods of ordinary differential equations in that our solution at a_{k+2} depends upon the values at a_k and a_{k+1}. The latter method assumes given boundary conditions such as $x(a) = x(b) = 0$ and computes the solution by relaxation methods.

The dynamic method has in fact been essentially described above and given by equations (6). For the usual examples we desire a one dimensional solution with $x(0) = 0$. This can be done by choosing c_1 to be some desired constant or by choosing $c_1 = 1$, solving recursively for c_k by equation (6), normalizing the c_k by dividing each c_k by $\max\{|c_k|\}$ if desired. If we desire an alternate initial condition such as $x'(a) = 0$ we begin with $k = 0$, $c_0 = 1$, $c_{-1} = c_1$ in (6c). This gives $c_0 = 1$, $c_1 = -c_0 e_{0,0}/(e_{0,-1} + e_{0,1})$, and c_k defined recursively by (6) for $k=2,3,4,\ldots$.

For boundary value problems such as $x(a) = x(b) = 0$ we used a relaxed or Gauss-Seidel method on the matrix $D(\sigma) = (e_{\alpha\beta})$ to solve $D(\sigma)c = 0$. We began with an initial guess for c which we call $c^{(0)}$. Let $c^{(n)}$ be the value of the vector $c = (c_0, c_1, \ldots, c_N)^T$ after n iterations and $c_0^{(n)} = c_{N+1}^{(n)} = 0$ for all n. Then from (6c) we have

$$(7) \qquad c_k^{(n+1)} = -(c_{k+1}^{(n)} e_{k,k+1} + c_{k-1}^{(n+1)} e_{k,k-1})/e_{k,k} \quad (k = 1,2,3,\ldots,N).$$

In our test cases we observed an interesting phenomena which agrees with the theory [2]. To be specific we assume our setting is such that $a = 0$ and the first conjugate point (focal point or oscillation point) is at $b = \pi$. If $a_N < \pi$ the matrix $(e_{\alpha\beta})$ is positive definite and hence the iteration (7) gives vectors $c^{(n)}$ which converge to $c^{(\infty)} = 0$. The convergence is monotone in that $0 < a_{N_1} < a_{N_2} < \pi$ implies that $\{c^{(n)}(N_1)\}$ converges to $c^{(\infty)} = 0$ faster than $\{c^{(n)}(N_2)\}$.

If $a_N > \pi$ the matrix (from our theory) has a negative eigenvalue and the iteration (7) diverges. As above the divergence is monotone in a_N. If a_N is close to π the divergence is slow while if $\pi < a_{N_1} < a_{N_2}$ the divergence of $\{c^{(n)}(N_2)\}$ is greater than the divergence of $\{c^{(n)}(N_1)\}$. If $a_N = \pi$ the matrix is (numerically) nonnegative definite and our iteration procedure converges to a vector in the null space. This fact follows from theorems in [1] or the theory given in [2].

TWO TEST CASES

In this section we will describe the computer runs for two equations.

The first equation is the differential equation $x'' + x = 0$, $x(0) = 0$. This was normalized so that the solution is sin t. In the appendix we will describe in detail computer runs for the second equation. In all computer runs the test runs of this constant coefficient case are slightly better than the equation described in the next equation.

Our second equation is the differential equation

$$[(2 + \cos t)x'(t)]' + (2 + 2 \cos t)x(t) = 0.$$

The reader can verify that sin t is "the" solution which vanishes at $t = 0$. This equation provides coefficients with reasonable change in values for values of t.

Table 1 is a listing of our dynamic or initial value program. The program can be easily modified to give the solution cos(t) to the initial problem with $x'(0) = 0$ instead of $x(0) = 0$. Table 2 is a listing of our relaxation or boundary value program with $x(0) = x(\pi) = 0$. We note that this program is in single precision arithmetic. Table 3 is our test case with a run of our dynamic program with size s = 1/128. Table 4 is a summary of runs of our dynamic program with different step sizes.

Table 5 is the test case of our relaxation method with step size $s = \frac{\pi}{100}$. In this run we initialized three values at 1, the remaining 98 values at 0 which is certainly "far removed" from the correct value of sin t. After 1000 steps we normalized the solution so that $x(\pi/2) = 1$. The differences of normalized values is given as $f(x_k) = \sin x_k - c_k/c_{51}$ since c_{51} is our largest value.

Since most of our test runs were performed on a terminal-time sharing arrangement exact run time is not possible. All the runs given in Table 4 including massive amounts of input and compiling time was 17 seconds. Hence our time for the test case in Table 3 would be a maximum of one second. Similarly the runs in Table 5 where we iterated 5000 times required 24 seconds.

For completeness we include a sketch of the piecewise approximation of our solution by our basis vectors in Table 6 where $a = 0$, $b = \pi$. We note also that double precision "Quartic Runge-Kutta" methods were applied in order to compare results with those of Table 4. As might be expected in this case (see Reference 3, p. 269) the methods were comparable in that the greater time and "cost" of the latter method were offset by increased accuracy.

REFERENCES

[1] George E. Forsythe and Wolfgang R. Wasow, 1960. Finite-Difference Methods for Partial Differential Equations, John Wiley and Sons, Inc., New York.

[2] John Gregory, Numerical algorithms for oscillation vectors of second order differential equations including the Euler Lagrange equation for symmetric tridiagonal matrices, The Pacific Journal of Mathematics, Volume 76, No. 2, June 1978, 397-406.

[3] Peter A. Starks, 1972, Introduction to Numerical Methods, The Macmillan Company, New York.

ACKNOWLEDGMENT

The authors would like to thank Joseph R. Beehenbach, Jr. for the sketch in Table 6 and Robert J. McClinn for his patience in programming support.

Table 1. Listing of program for initial value problem $(p(t)x')' + q(t)x = 0$ where $p(t) = 2 + \cos t$ and $q(t) = 2 + 2\cos t$.

```
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION C(3)
    C NUMBER OF DIVISIONS OF INTERVAL ARE READ IN, IF S=0 STOP
  100 READ (5,150)S
  150 FORMAT (D10.4)
      IF(S.EQ.0.D0) GO TO 401
      XS= S*7.D0
      N=IDINT(XS)
      S=1.D0/S
      B=S/2.D0
      BB=S+S/2.D0
    C E1=DIAGONAL ELEMENT, E2 AND E3 ARE OFF-DIAGONAL ELEMENTS
      E1=DCOS(B)+DCOS(BB)+4.D0-S*S*(2.D0*DCOS(B)+2.D0*DCOS(BB)+4.D0)
     */3.D0
      E2=-DCOS(BB)-2.D0-S*S*(DCOS(BB)+1.D0)/3.D0
    C SOLUTION IS NORMALIZED TO SINX
      C(1)=DSIN(S)
      C(2)=-C(1)*E1/E2
      WRITE (6,200) S,C(1)
  200 FORMAT(T2,'STEP SIZE =',F11.8,T25,'C(1) =',F16.14)
      WRITE (6,201)
  201 FORMAT('1')
      WRITE (6,202)
  202 FORMAT(T2,'X VALUE',T20,'ALGORITHM VALUE',T45,'ACTUAL VALUE',
     *T70,'DIFFERENCE')
      DO 250 I=3,N
      T=I
      B=(T-2.D0)*S+S/2.D0
      BB=(T-1.D0)*S+S/2.D0
      E1=DCOS(B)+DCOS(BB)+4.D0-S*S*(2.D0*DCOS(B)+2.D0*DCOS(BB)+4.D0)
     */3.D0
      E3=-DCOS(BB)-2.D0-S*S*(DCOS(BB)+1.D0)/3.D0
    C C(3) REPRESENTS THE I+2 DYNAMIC VALUE
      C(3)=(-C(1)*E2-C(2)*E1)/E3
      E2=E3
      M=N/100
      C(1)=C(2)
      C(2)=C(3)
      IF (MOD(I,M)) 250,210,250
  210 Y=I
      X=Y*S
      Z=DSIN(X)
      W=Z-C(3)
      WRITE(6,225)(X,C(3),Z,W)
  225 FORMAT(T2,F8.5,T20,F16.14,T45,F16.14,T70,D22.14)
  250 CONTINUE
      WRITE (6,300)
  300 FORMAT ('1')
      GO TO 100
  401 STOP
      END
```

Table 2. Listing of program for boundary value problem $(p(t)s')' + q(t)x = 0$ with $p(t) = 2 + \cos t$ and $q(t) = 2 + 2 \cos t$.

```
      DIMENSION EU(101),ED(101),C(101),F(101)
      P(X)=2.0+COS(X)
      R(X)=2.0+2.0*COS(X)
      N=101
      NN=N-1
    C NN=NUMBER OF DIVISIONS OF THE INTERVAL
      S=3.141592654/NN
      DO 11 I=1,N
 11   C(I)=0.0
      C(20)=1.0
      C(45)=1.0
      C(30)=1.0
      DO 50 K=2,NN
      AK=K*S-S/2.0
      AL=AK-S
      ED(K)=P(AL)+P(AK)-S*S*(R(AL)+R(AK))/3.0
      EU(K)=-P(AK)-S*S*R(AK)/6.0
 50   CONTINUE
      DO 100 ITER=1,5000
      DO 99 K=2,NN
  C   GAUSS-SEIDEL ITERATION STEP
 99   C(K)=-(EU(K-1)*C(K-1)+EU(K)*C(K+1))/ED(K)
      M=MOD(ITER,1000)
      IF(M.NE.0) GO TO 100
      WRITE (6,75) ITER
 75   FORMAT(T2,'NUMBER OF ITERATIONS=',I5)
      WRITE (6,76)
 76   FORMAT(T2,'X-VALUE',T20,'ACTUAL-VALUE',T45,'ALGORITHM-
     *VALUE',T70,'DIFFERENCE',T90,'NORMALIZER')
      DO 98 J=1,N
 98   F(J)=C(J)/C(51)
      DO 70 J=1,N,5
      X=(J-1)*S
      Z=SIN(X)
      D=Z-F(J)
      WRITE(6,80)(X,Z,F(J),D,C(51))
 80   FORMAT(T2,F8.5,T20,F16.8,T45,F16.8,T70,E16.8,T90,F16.8)
 70   CONTINUE
      WRITE (6,110)
110   FORMAT('1')
      IF (ITER.NE.1000) GO TO 100
  C   RE-INITIALIZE SOLUTION TO SIN X
 85   C(L)=F(L)
100   CONTINUE
      STOP
      END
```

Table 3. Test case with step size = 1/128 of the initial value problem $(p(t)x')' + q(t)x = 0$ where $p(t) = 2 + \cos t$ and $q(t) = 2 + 2 \cos t$.

X VALUE	ALGORITHM VALUE	ACTUAL VALUE	DIFFERENCE
0.0625	0.062459	0.062459	-0.542D-09
0.3125	0.307438	0.307438	-0.680D-07
0.5625	0.533303	0.533302	-0.386D-06
0.8125	0.726009	0.726008	-0.111D-05
1.0625	0.873577	0.873574	-0.233D-05
1.3125	0.966830	0.966826	-0.405D-05
1.5625	0.999971	0.999965	-0.617D-05
1.8125	0.970940	0.970931	-0.850D-05
2.0625	0.881540	0.881529	-0.107D-04
2.3125	0.737331	0.737318	-0.127D-04
2.5625	0.547279	0.547264	-0.142D-04
2.8125	0.323199	0.323184	-0.152D-04
3.0625	0.079026	0.079010	-0.158D-04
3.3125	-.170060	-.170076	-0.163D-04
3.5625	-.408571	-.408588	-0.169D-04
3.8125	-.621679	-.621696	-0.176D-04
4.0625	-.796132	-.796150	-0.183D-04
4.3125	-.921085	-.921104	-0.186D-04
4.5625	-.988769	-.988787	-0.181D-04
4.8125	-.994976	-.994993	-0.165D-04
5.0625	-.939320	-.939334	-0.136D-04
5.3125	-.825263	-.825272	-0.974D-05
5.5625	-.659894	-.659899	-0.486D-05
5.8125	-.453497	-.453497	0.569D-06
6.0625	-.218904	-.218898	0.610D-05
6.3125	0.029299	0.029310	0.112D-04
6.5625	0.275681	0.275696	0.153D-04
6.8125	0.504923	0.504941	0.181D-04

Table 4. Error range of solution values of the initial value problem $(p(t)x')' + q(t)x = 0$ where $p(t) = 2 + \cos t$ and $q(t) = 2 + 2 \cos t$ with step size = s on the range 0 to 2π.

step size	maximum error	minimum error
1/32	0.301D-03	0.661D-07
1/64	0.761D-04	0.207D-08
1/128	0.190D-04	0.543D-09
1/256	0.475D-05	0.165D-09
1/512	0.119D-05	0.451D-10
1/1024	0.295D-06	0.118D-10
1/2000	0.798D-07	0.317D-11
1/4000	0.285D-07	0.823D-12

Table 5. Differences of normalized values of solution to the boundary value problem $(p(t)x')x' + q(t)x = 0$ where $p(t) = 2 + \cos t$ and $q(t) = 2 + 2\cos t$ with step size = $\pi/100$ on the range 0 to π.

X-VALUE	1000 iterations normalized value = 0.05646	2000 iterations normalized value = 1.00609	3000 iterations normalized value = 1.00529	4000 iterations normalized value = 1.00423	5000 iterations normalized value = 1.00318
0.15708	-0.110E-01	-0.477E-03	-0.289E-04	-0.607E-05	-0.476E-05
0.31416	-0.209E-01	-0.905E-03	-0.556E-04	-0.120E-04	-0.965E-05
0.47124	-0.286E-01	-0.124E-02	-0.807E-04	-0.211E-04	-0.168E-04
0.62832	-0.336E-01	-0.146E-02	-0.100E-03	-0.312E-04	-0.255E-04
0.78540	-0.352E-01	-0.154E-02	-0.111E-03	-0.390E-04	-0.329E-04
0.94248	-0.335E-01	-0.147E-02	-0.108E-03	-0.404E-04	-0.340E-04
1.09956	-0.285E-01	-0.125E-02	-0.920E-04	-0.345E-04	-0.290E-04
1.25664	-0.207E-01	-0.914E-03	-0.658E-04	-0.252E-04	-0.202E-04
1.41372	-0.109E-01	-0.482E-03	-0.341E-04	-0.144E-04	-0.102E-04
1.57080	0.0	0.0	0.0	0.0	0.0
1.72788	0.110E-01	0.471E-03	0.252E-04	0.542E-05	0.536E-05
1.88496	0.211E-01	0.900E-03	0.455E-04	0.971E-05	0.822E-05
2.04204	0.293E-01	0.124E-02	0.645E-04	0.140E-04	0.125E-04
2.19911	0.347E-01	0.148E-02	0.825E-04	0.211E-04	0.191E-04
2.35619	0.367E-01	0.157E-02	0.889E-04	0.237E-04	0.212E-04
2.51327	0.351E-01	0.150E-02	0.854E-04	0.219E-04	0.198E-04
2.67035	0.299E-01	0.128E-02	0.702E-04	0.413E-04	0.136E-04
2.82743	0.217E-01	0.929E-03	0.483E-04	0.780E-05	0.762E-05
2.98451	0.114E-01	0.486E-03	0.246E-04	0.351E-05	0.351E-05
3.14159	0.627E-06	0.627E-06	0.627E-06	0.627E-06	0.627E-06

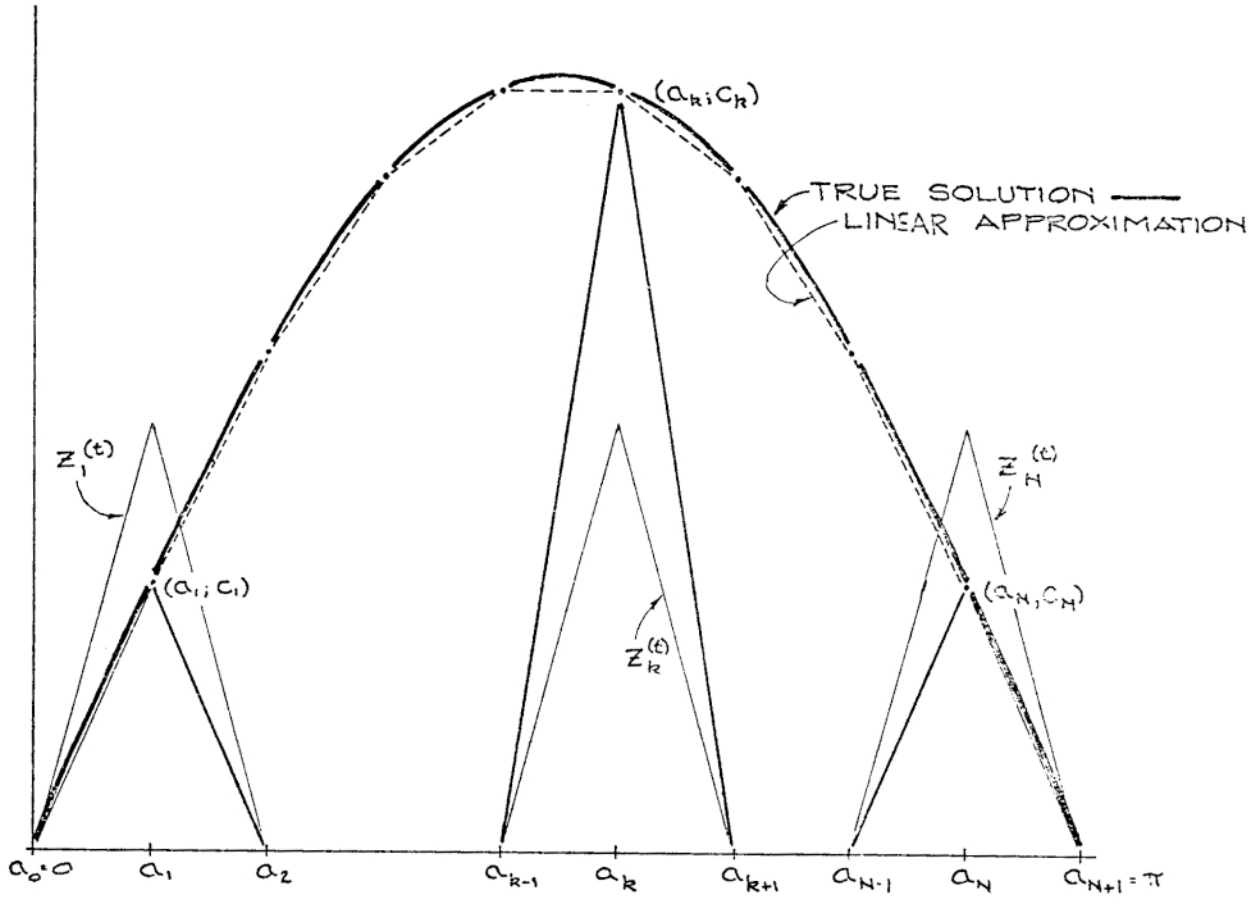

TABLE SIX- DIAGRAM OF SOLUTION AND APPROXIMATION