Mat. Aplic. Comp., V. 2, nº 2, pp. 103 a 118, 1983

PASVA4: *AN O.D.E. BOUNDARY SOLVER FOR PROBLEMS WITH DISCONTINUOUS INTERFACES AND ALGEBRAIC PARAMETERS*

M. LENTINI
Departamento de Matemáticas
Universidad Simón Bolívar - Apartado 80659
Caracas - Venezuela

V. PEREYRA
Departamento de Matemáticas y Ciencías de la Computación
Universidad Central de Venezuela - Apartado 47002
Caracas 1041 - Venezuela

ABSTRACT

PASVA4, an updated version of our earlier code PASVA3 for solving nonlinear two-point boundary-value problems for first order systems of ordinary differential equations, is presented. The two main new capabilities of PASVA4 are the direct, efficient handling of internal jump discontinuities, and the possibility of solving simultaneously for parameters and additional algebraic equations. The use of the code is exemplified by solving a seismic ray tracing problem and the modelling of semiconductor devices.

RESUMO

Apresentamos neste artigo uma versão atualizada do sistema PASVA3, que se destina à resolução de problemas de contorno não lineares para sistemas de equações diferenciais ordinárias de primeira ordem. Esta versão, denominada PASVA4, tem como principais características o tratamento direto e eficiente de descontinuidades internas do tipo salto e a possibilidade de resolução simultânea com relação a parâmetros e adicionais equações algébricas. Para exemplificar o uso do sistema, resolvemos dois problemas: da determinação de traços sísmicos e da modelagem de semicondutores.

1. INTRODUCTION

Until fairly recently there were no off-the-shelf codes available for the solution of nonlinear two-point boundary value problems.

In general, interested users wrote their own software, usually based on available initial value codes. This situation started to change around 1972 and at present we can find several extensively tested codes, specially designed for the solution of wide classes of two-point boundary value problems (see [14] for state-of-the-art information as of 1978). These codes are characterized for being robust, portable, and widely disseminated.

However, recent experience with off-the-shelf solvers for two-point boundary value problems has shown their inadequacy to attack a variety of applications of pratical interest.

Among applications of interest to physicists we can mention: linear and nonlinear eigenvalue problems (including bifurcation studies), problems with discontinuous interfaces, free boundary problems, and optimal control problems.

Although there are several "recipes" to reduce some of these problems to a form acceptable to the particular solver [7,12], usually the solution of the transformed problem ammounts to a fairly inneficient approach to the solution of the original one.

Because of these reasons we have deemed necessary to produce an updated version of our code PASVA3 [4,5,6], that will be able to cope with some of these problems more graciously than at present. PASVA3 was designed to solve problems of the form y'=f(t,y),g(y(a),y(b))=0, where f is a smooth function of f and y. We have modified this code so that now it is possible to solve directly problems where the right hand side f, and perhaps the solution y,has jump discontinuities with respect to t (Section 3), and also problems which contain parameters and corresponding additional (multipoint) boundary conditions (Section 4).

We have taken special measures in order to make PASVA4 upper compatible with PASVA3. In this way we expect that former users can upgrade their versions without undue violence to production programs. We have also tried to make the documentation available in a "need to know" format *i.e.* the documentation for the different options is organized in such a way that the user needs to read only those parts which are pertinent to his application. This feature is fairly important, since now the operation of the code in its full generality has become somewhat complicated and that may be discouraging for the inexperienced user. In order to alleviate this problem we offer in Section 6 several examples of

usage of the various options which will be of interest to some readers of this journal.

2. A REVIEW OF *PASVA3*

We would like to give here for completeness a brief description of our earlier code PASVA3, so that we can explain the modifications and additions we have made to it. More complete details can be found in [5,6].

PASVA3 is a finite difference solver for first order nonlinear system of ordinary differential equations with nonlinear two point boundary condition. The basic discretization used is the trapezoidal rule on a general mesh

$$\pi = \{t_i\} \ , \ a = t_1 < t_2 < \ldots < t_n = b \ , \ h_i = t_{i+1} - t_i$$

$$y_{i+1} - y_i = h_i(f(t_i,y_i) + f(t_{i+1},y_{i+1}))/2 \qquad (1)$$

$$g(y_1,y_n) = 0 \ , \quad i = 1,\ldots,n-1 \ .$$

This $O(h^2)$ method is enhanced by deferred corrections. There is also a procedure for automatically choosing an appropriate mesh π, which is based on the (sub) equidistribution of an estimate of the local truncation error.

The variable order, variable step procedures mesh together in order to produce a fairly robust, efficient and versatile program which has been widely used in the past few years (see [8,9] for comments on performance).

For the higher order corrections to be effective it is necessary that the solution y(t), and therefore the data f(t,y), be sufficiently smooth. Problems where f(t,y) has end-point singularities can be handled by the methods of Lentini and Keller [2,3,13] without degradation in performance.

On the other hand, problems with internal jump discontinuities require different techniques. A frequently advocated one is "multiplexing",which amounts to writing a new copy of the differential system for each interval of continuity. Thus, a problem with m differential equations and k discontinuities can be transformed into one with (k+1)m equations and no discontinuities. In order to make possible this transformation, each interval between successive discontinuities is mapped into [0,1] (say). Thus, solving the transformed problem on a mesh in [0,1] is equivalent to solving the original problem with this mesh mapped back into each subinterval of continuity. In principle, the critical para-

meter (number of differential equations times number of mesh points) remains constant, although a more careful check into the structure of the system of discrete equations shows that the amount of work and storage is increased by a factor (k+1), the number of subintervals of continuity.

Also, the mesh produced will correspond to the worst possible behavior of the solution. Although this behavior might be concentrated only in one subinterval, the mesh will get copied to all other intervals in virtue of the transformation performed.

3. DISCONTINUOUS DATA

In this Section we consider problem $y'=f(t,y)$, $g(y(a),y(b))=0$, where $f(t,y)$ has a number of jump discontinuities with respect to t at the known locations

$$a < td_1 < td_2 < \ldots < td_k < b \quad , \quad i \leq k \quad . \tag{2}$$

The solution y(t) may, or may not be discontinuous at those points, but it is assumed to have one-sided limits $y(td_i^-)$, $y(td_i^+)$. In all cases the problem description is completed by giving appropriate jump conditions:

$$D_i(y(td_i^-),y(td_i^+)) = 0 \quad , \quad i = 1,\ldots,k \quad , \tag{3}$$

where the vectors D_i have dimension m.

The transformation of Section 2 is at all necessary in PASVA3 because of the fact that a finite difference scheme should never straddle a discontinuity. As it was indicated in [4], the trapezoidal rule itself will not be affected by jump discontinuities provided these are located at mesh points. One has only to put up with the mild inconvenience of maybe having two different side limits at such points. Even that inconvenience could be avoided by considering the midpoint rule instead.

However, the higher order approximations will involve more than one mesh interval, and unless special precautions are taken they will involve mesh points at both sides of the discontinuities, which will usually prove to be disastrous. Thus, we have modified our deferred correction and error estimation procedure so that this fact is taken into account. The program will assume that the user has indicated the discontinuity points by repeating them in the mesh vector, *i.e.* each td_i must appear not once, but twice, in consecutive locations in the

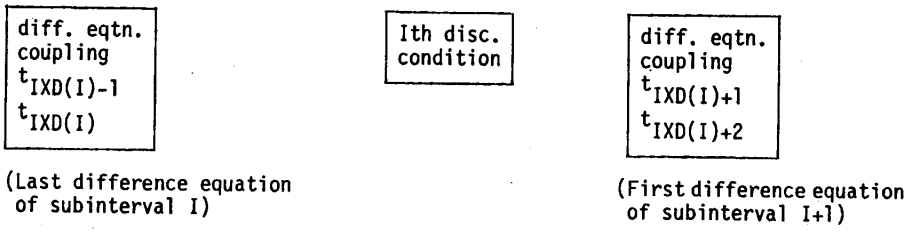
mesh. The indices of the first appearance of each td_i are recorded in the integer array IXD. That information will be enough to drive the rest of the computation.

The modification for the deferred correction procedure is very simple and it was already used experimentally in an earlier version of the code [4]. It amounts to calling the current procedure for each subinterval of continuity. Of course, this requires that there are enough points in the interior of these subintervals in order to carry out the desired correction.

The repetition of the discontinuity points in the mesh implies that for a given number n of physical mesh points, there will be (n+k) positions on the working mesh. This, of course, is reflected in all the other arrays that are associated with the mesh, like Y, F and its Jacobian matrix. Thus, we have in this way provided a natural place to store the right and left limits of all discontinuous quantities.

The last problem to be solved is where to put the discontinuity conditions (3) so that the linear equation solver is not unduly disturbed. We recall that the longest and, by far, most complicated part of PASVA3 is the special linear equation solver used at each step of Newton's method to solve the linearized discrete equations. These equations are casted as a block tridiagonal system with some additional sparse structure in the off-diagonal blocks. An alternating pivoting strategy is used in order to preserve this sparsity, while still producing an stable LU decomposition.

It turns out that if we merge each m-block equation (3) at the IXD(I) position, then we do not disturb the structure of the matrix at all, and the same solver can be used on this enlarged matrix. These equations take the place of a difference equation block for the pseudo interval (of length 0) $[t_{IXD(I)}, t_{IXD(I)+1}]$. Graphically



(Last difference equation of subinterval I)

(First difference equation of subinterval I+1)

Most of these changes have been made transparent to the user, and all that he

has to do is to provide the minimum information necessary to describe his problem.

The important case of movable discontinuities, *i.e.* dependent upon t and y, will be treated in Section 5.

4. PARAMETERS AND ALGEBRAIC EQUATIONS

In this Section we consider the problem

$$\begin{aligned} y' &= f(t,y;\lambda) \\ g(y(a),y(b);\lambda) &= 0 \quad , \end{aligned} \tag{4}$$

where $\lambda \in \mathbb{R}^{\ell}$, $1 \leq \ell$. Here $y,f \in \mathbb{R}^{m}$, while $g \in \mathbb{R}^{m+\ell}$, and it is assumed partitioned as (g_1,g_2,g_3,g_4), where

g_1 contains the purely initial conditions;

g_2 contains coupled conditions;

g_3 contains purely final conditions;

g_4 contains the additional conditions necessary to determine the parameters λ. These can be of any of the three types above, and they have been separated for reasons which will become apparent below.

Our main objective here is to be able to use the principal module of PASVA3, the linear equation solver, without changes.

Let L_π represent the matrix of the standard linearized equations associated with (1). Systems of this type are the ones solved by the pair of subroutines DECOMP-SOLVE of PASVA3, once the matrix L_π has been set up by SYSLIN.

For the new linearized system, if the equations are ordered as indicated above, and the unknowns are ordered as $[y,\lambda]$, then, schematically, the format will be the following

$$\tilde{L}_\pi \begin{bmatrix} \Delta y \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} L_\pi & f_\lambda \\ g_{4y} & g_{4\lambda} \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} . \tag{5}$$

Provided L_π and $\tilde{L}_\pi$ are non-singular, we can solve this system by one call

to DECOMP to obtain the LU decomposition of L_π, and repeated use of SOLVE, a few matrix operations, and the solution of a small $\ell\times\ell$ system, namely:

a) get the LU decomposition of L_π;

b) solve $L_\pi z = r_1$;

c) solve $L_\pi B = f_\lambda$;

d) form $G = g_{4\lambda} - g_{4y}B$ and $\beta = r_2 - g_{4y}z$;

e) solve $G\,\Delta\lambda = \beta$;

f) compute $\Delta y = z - B\,\Delta\lambda$.

Observe that the matrix g_{4y} is very sparse and therefore (d) requires only some very short matrix multiplications. The fact that G is nonsingular is a consequence of the assumed non-singularity of L_π and $\tilde{L}_\pi$ (by Schur's Complement Theorem).

The discontinuous case can now be handled in the same way, combining the procedure of Section 3 and the one just explained. The only new feature is that we will have to intercalate in f_λ, at the appropriate spots, the Jacobian matrices of the discontinuity conditions with respect to λ. We also allow in this case more general algebraic conditions of multipoint type:

$$g_4(y(a),y(td_1^-),y(td_1^+),\ldots,y(td_k^+),y(b)) = 0 \quad , \tag{6}$$

which imply a more complex structure in g_{4y} and therefore in the sparse matrix multiplication in step (d) of the new linear solver (SYSNEW).

From the method of solution it is easy to see that if r_1 represents the local truncation error of the stable method $L_\pi y=0$, and $r_2=0$, then Δy, $\Delta\lambda$ will be the principal terms of the global error. If $r_1=\mathcal{O}(h^\alpha)$ them from (b) and the stability of L_π it follows that $z=\mathcal{O}(h^\alpha)$, which by (d) and (e) implies that $\beta=\mathcal{O}(h^\alpha)$, and therefore $\Delta\lambda=\mathcal{O}(h^\alpha)$. Finally (f) implies that $\Delta y=\mathcal{O}(h^\alpha)$, and both y and λ get computed with accuracy of order h^α. Observe that G has also a bounded inverse uniformly with respect to $h=|\pi|$, under minimal conditions on f_λ, $g_{4\lambda}$, and g_{4y}.

An extension of the theoretical results of [1,4,5] to cover rigorously the present case is straightforward (with a few surprises), and a student of ours is filling some of the gaps.

5. MOVABLE DISCONTINUITIES

A very important problem in the applications (see Section 6), not contemplated so far, is the one in which the jump discontinuities are determined by certain conditions on the trajectory y(t). These conditions are usually represented by the zeroes of given switching functions $\phi_i(t,y(t))$. This situation arises, for instance, in problems with material interfaces, in some optimal control problems, and in free boundary problems.

The type of problems we consider in this Section is of the form:

$$y' = \begin{cases} f^{(i)}(t,y(t)) & \text{if} \quad \phi_i(t,y(t)) < 0 \\ f^{(i+1)}(t,y(t)) & \text{if} \quad \phi_i(t,y(t)) \geq 0 \end{cases} \tag{7}$$

$$g(y(a),y(b)) = 0 \quad ,$$

and thus the jump discontinuities will be located at the points td satisfying

$$\phi_i(td,y(td)) = 0 \quad , \tag{8}$$

where y(t) is the solution to problem (7).

We cannot solve this problem by the methods described earlier because the values of the td are unknown. Ideally we would like to be able to determine in the course of the computation how many switches do occur, but this will not be possible with our technique. Thus, we must assume *a priori* knowledge of the number of switches k, and also which ones of the switching functions are activated.

The trick to make this problem conform to PASVA4 format is a standard one. Simply, we map each subinterval $[td_i,td_{i+1}]$ into the known subinterval [i,i+1] (say), assuming for completeness that $td_0=a$, $td_{k+1}=b$, and we consider each td_i as an unknown parameter. Putting $\lambda_i=td_i$, (7) is transformed into

$$\begin{aligned} &y' = (\lambda_i-\lambda_{i-1})f^{(i)}(t,y) \quad , \quad i-1 \leq t < i \quad , \\ &g(0,k+1) = 0 \qquad\qquad , \quad i=1,\ldots,k+1 \end{aligned} \tag{7'}$$

where $\lambda_0=a$, $\lambda_{k+1}=b$ are not considered as unknowns. The switching conditions provide the additional equations (6)

$$g_{4i} = \phi_i(i,y(i^-)) = 0 \quad .$$

6. APPLICATIONS

SEISMIC RAY TRACING ON PIECEWISE CONTINUOUS, ISOTROPIC MEDIA. An important problem in seismology is the study of the propagation of elastic waves in general media.

A wave front is the set of points in space on which the solution to the elastic wave equation is discontinuous at time t, say $\Psi(\underset{\sim}{\eta})=t$, $\underset{\sim}{\eta}=(x,y,z)$. The orthogonal trajectories to the wave fronts are called the rays. These rays are solutions of the system of ordinary differential equations

$$d(u(\underset{\sim}{\eta})\, d\underset{\sim}{\eta}/ds)/ds = \nabla u(\underset{\sim}{\eta}) \quad , \tag{9}$$

where $u(\underset{\sim}{\eta})=1/v(\underset{\sim}{\eta})$, and $v(\underset{\sim}{\eta})$ is the velocity of propagation of the type of wave being studied. The independent variable s is arc length along the ray, so we require that $d\underset{\sim}{\eta}/ds=\dot{\underset{\sim}{\eta}}$ satisfies $\|\dot{\underset{\sim}{\eta}}\|_2^2=1$, where $\|\cdot\|_2$ is the Euclidean norm of vectors. The two-point ray tracing problem consists in finding the solution of (9) that joins two given points in $\underset{\sim}{\eta}$ space.

In two dimensions $\underset{\sim}{\eta}=(x,z)$, equations (9) can be reduced to

$$\begin{aligned} w_1' &= \lambda \cos w_3 \\ w_2' &= \lambda \sin w_3 \\ w_3' &= \lambda\, v(w_1,w_2)(u_z \cos w_3 - u_x \sin w_3) \end{aligned} \tag{10}$$

where $w_1=x$, $w_2=z$, $w_3=\Phi$ (the "ray angle"), $w_i'=dw_i/d\tau$, with $\tau=s/\lambda$, and λ is the total arc length of the ray within the interval of integration (unknown).

As soon as $v(x,z)$ is non-constant (i.e. the medium is inhomogeneous), eqs. (9) and therefore (10) are very non-linear.

It is quite usual in the applications that the medium described by the function $v(x,z)$ has material interfaces, and therefore $v(x,z)$ is only piece-

wise continuous. The interfaces will be described by (switching) functions $\phi_i(x,z)=0$.

In [10] we have described in all detail how to solve the two-point ray tracing problem in piecewise continuous media with PASVA3, via multiplexing. It is quite clear that PASVA4 has exactly the features necessary to solve this problem more efficiently; we simply have to use the procedures described in Sections 4 and 5. A detailed description of this and related applications can be found in [11].

The algebraic (or interface) conditions are given by Snell's law and the continuity of the ray; if +, - indicate right and left limits respectively, these conditions are

$$w_1(i^-) = w_1(i^+)$$

$$w_2(i^-) = w_2(i^+) \tag{11}$$

$$v_{i+1}(\phi_{ix} \sin \Phi(i^-)-\phi_{iz} \cos \Phi(i^-))-v_i(\phi_{ix} \sin \Phi(i^+)-\phi_{iz} \cos \Phi(i^+))=0 ,$$

and finally by the fact that the ray must touch the interface

$$\phi_i(w_1(i),w_2(i)) = 0 . \tag{12}$$

As a simple example (very hard or impossible to solve by the conventional means commonly used in seismology) we consider a generalized layered two-dimensional model of a dome with an imbedded low velocity lens (see Fig. 1). The curved interfaces considered depend upon three parameters, and have the form

$$z = D_j + C_j/2\ (1-\cos(2\pi(x-\omega_j/2)/\omega_j)) , \quad \text{in some region}$$

$$z = D_j \qquad \text{elsewhere} ,$$

where $D_j \geq 0$ is a constant, the horizontal background, C_j is the maximum departure from it, giving a cos-like dome (or depression), between the tappering points $-\omega_j/2$, $\omega_j/2$.

The case of Fig. 1 corresponds to the following values of the parameters.

j	C	D	ω	(km)
1	0	0	0	(free surface)
2	2.0	-0.6	7.0	
3	2.5	-0.3	4.0	
4	2.5	0.3	4.0	

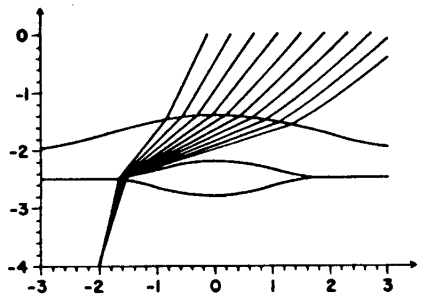


Fig. 1

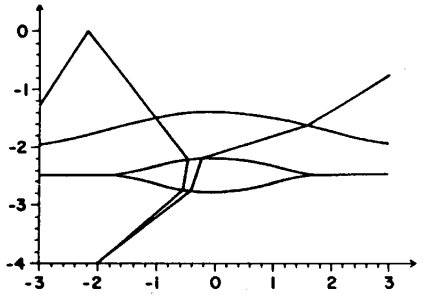


Fig. 2

Ray tracing on curved media

We use in this example a simple velocity model. In the first layer (from the free surface) $v_1(x,z)=1.9\ z^{1/6}$ km/s, which is an empirical law for some kinds of sediments. In the other layers we consider constant velocities, $v_2=2.7$km/s, $v_3=1.0$ km/s (in the lens), $v_4=5.0$ km/s (in the half space).

In Fig. 1 we show a graph of 10 rays calculated with PASVA4. They join a common source located at x=-2, z=-4, with stations located on the free surface at $x_j=-0.1+(j-1) * 0.4$, j=1,...,10. Each one of these rays has four continuous segments, and therefore PASVA4 computes them more than four times faster than PASVA3.

In Fig. 2 we show what happens with this very same problem when one shoots rays: a very small variation in the initial ray angle produces extremely different rays.

MODELLING OF SEMICONDUCTOR DEVICES. The following problem was described to us by Peter Markowich in a private communication. The numerical results and the asymptotic analysis are in [15]. This semiconductor problem demonstrates how well PASVA4 can handle interface and internal layers problems.

The equations that describe the problems, after scaling, are:

$$
\begin{aligned}
\lambda^2\psi'' &= n - p - D(X) \\
n' &= n\psi' + J_n \\
J_n' &= (np-\gamma^4\lambda^4)/4(n+p+2\gamma^2\lambda^2) \\
J_p' &= (np-\gamma^4\lambda^4)/4(n+p+2\gamma^2\lambda^2) \\
p' &= -p\psi' - J_p
\end{aligned}
\tag{13}
$$

for $-1 \leq X \leq L$, subjected to the boundary conditions

$$
\begin{aligned}
\psi(-1) &= \ell n[2\gamma^2\lambda^2/(-D(-1)+\sqrt{D(-1)^2+4\gamma^4\lambda^4})] + U_A/U_T \\
\psi(1) &= \ell n[(D(1)+\sqrt{D(1)^2+4\gamma^4\lambda^4})/2\gamma^2\lambda^2] + U_C/U_T \\
n(1) &= [D(1)+\sqrt{D(1)^2+4\gamma^4\lambda^4}]/2 \\
n(-1) &= [D(-1)+\sqrt{D(-1)^2+4\gamma^4\lambda^4}]/2 \\
P(1) &= [-D(1)+\sqrt{D(1)^2+4\gamma^4\lambda^4}]/2 \\
P(-1) &= [-D(-1)+\sqrt{D(-1)^2+4\gamma^4\lambda^4}]/2 \quad .
\end{aligned}
\tag{14}
$$

Here λ is a singular perturbation parameter ($\lambda \to 0^+$), $\gamma^2=0.25$, the thermal voltage $U_T=0.025$ volts, $U=U_A-U_C$ is the applied voltage and $D(X)$ is the scaled doping profile given by

$$
D(x) = \begin{cases} D(-1) & -1 \leq x < X \qquad D(-1) < 0 \\ D(1) & X < x \leq 1 \qquad D(+1) > 0 \end{cases}
$$

for some $X \in (-1, 1)$.

The solution is required to satisfy

i) $n, p \geq 0$ on $[-1, 1]$ (physical assumption)

ii) $n,p,\psi,J_n,J_p \in C'[-1, 1]$

iii) $\psi'' \in C([-1, X) \cup (X, 1])$.

The computations were done for U-values satisfying

$$-1 \text{ volt} \leq U = U_A - U_C \leq 2 \text{ volts} .$$

As usual in singular perturbation theory the solution can be split up asymptotically into a reduced solution (independent of λ) and an interval layer solution (at $x=X$) which is an exponentially decaying function of $\tau=(x-X)/\lambda$ (as $\tau \to \pm\infty$).

Thus we can solve independently the reduced equations (for $\lambda=0$) and the internal layer problem (for $\tau=(x-X)/\lambda$). Then asymptotically we have

$$\psi(x,\lambda) \sim \bar{\psi}(x) + \hat{\psi}\left(\frac{x-X}{\lambda}\right)$$

$$n(x,\lambda) \sim \bar{n}(x) + \hat{n}\left(\frac{x-X}{\lambda}\right)$$

$$p(x,\lambda) \sim \bar{p}(x) + \hat{p}\left(\frac{x-X}{\lambda}\right)$$

$$J_n(x) \sim \bar{J}_n(x)$$

$$J_p(x) \sim \bar{J}_p(x)$$

where the reduced solutions are marked with '$-$' and the layer solutions are marked with '$\sim$'.

In this way it is possible to get a good initial guess to solve the full equations described in (13) and (14). To get a complete description of the reduced and layer equations see [15].

We show in Figures 3-4 the computed solutions for $\lambda^2=4\times10^{-7}$ and

$$D(x) = \begin{cases} -0.5 & -1 \leq x < 1/2 \\ 1 & 1/2 < x \leq 1 \end{cases}$$

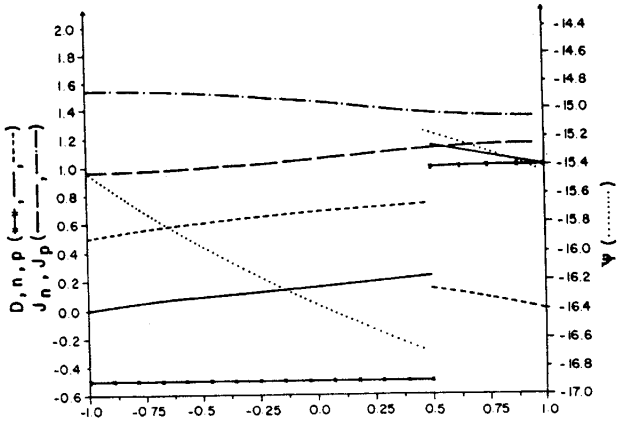


Fig. 3. Reduced solution

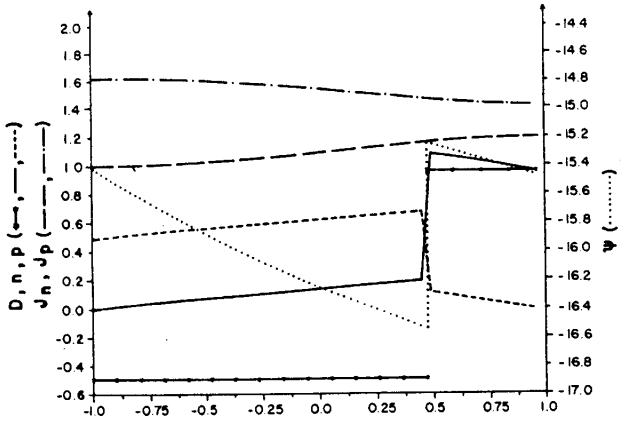


Fig. 4. Full solution

U=0.79 volts

7. REFERENCES

[1] KELLER, H.B. - "Numerical solution of two point boundary value problems", *Reg. Conference Series Appl. Math.*, vol. 24, SIAM, Philadelphia, Penn., 1976.

[2] LENTINI, M. - "Boundary value problems over semi-infinite intervals", *Ph.D. Thesis*, Caltech, Pasadena, Cal., 1978.

[3] LENTINI, M.; KELLER, H.B. - "Boundary value problems over semi-infinite intervals and their numerical solution", *SIAM J. Numer. Anal.*, vol. 17, pp. 577-604, 1980.

[4] LENTINI, M.; PEREYRA, V. - "A variable order finite difference method for nonlinear multipoint boundary value problems", *Math. Comp.*, vol. 28, pp. 981-1003, 1974.

[5] LENTINI, M.; PEREYRA, V. - "An adaptive finite difference solver for nonlinear two-point boundary problems with mild boundary layers", *SIAM J. Anal.*, vol. 14, pp. 91-111, 1977.

[6] PEREYRA, V. - "PASVA3: an adaptive finite difference FORTRAN program for first order nonlinear ordinary boundary problems", *Lect. Notes Comp. Sc.*, vol. 76, pp. 67-88, Springer-Verlag, Berlin, 1978.

[7] PEREYRA, V. - "Solución numérica de ecuaciones diferenciales con valores de frontera", *Acta Cient. Venezolana*, vol. 30, pp. 7-22, 1979.

[8] PEREYRA, V.; RUSSELL, R.D. - "Difficulties of comparing complex mathematical software: general comments and the BVODE case", Submitted for Public. to *Acta Científica Venezolana*.

[9] HEMKER, P.W.; SCHIPPERS, H. and DE ZEEUW, P.M. - "Comparing some aspects of two codes for two-point boundary-value problems", *NW98/80, Math.Centrum*, Amsterdam, 1980.

[10] PEREYRA, V.; LEE, W.H.K. and KELLER, H.B. - "Solving two-point seismic ray tracing problems in a heterogeneous medium. Part 1. A general adaptive finite difference method", *Bull. American Sism. Soc.*, vol. 70, pp. 79-99, 1980.

[11] PEREYRA, V.; WOJCIK, G. - "Interactive ray tracing on complex geological structures", in preparation.

[12] ASCHER, U.; RUSSELL, R.D. - "Reformulation of boundary value problems into 'Standard' form", *SIAM Review*, vol. 23, pp. 238-254, 1981.

[13] LENTINI, M. - "Resolución de ecuaciones diferenciales ordinarias singulares con valores de frontera", *Acta Científica Venezolana*, vol. 31, pp. 381-393, 1980.

[14] CHILDS, B.; SCOTT, M.; DANIEL, J.W.; DENMAN, E. and NELSON, P. (Editors) - "Codes for boundary-value problems in ordinary differential equations", *Lect. Notes Comp. Sc.*, vol. 76, Springer-Verlag, Berlin, 1979.

[15] MARKOWICH, P.; RINGHOFER, Ch.; SELBERHERR, S. and LANGER, E. - "A singularly perturbed boundary value problem modelling a semiconductor device", *MRC Technical Summary Report*, nº 2388, Madison, Wisconsin, 1982.

Recebido em 20/XII/82