Computing, Suppl. 5, 211–226 (1984)

Deferred Corrections Software and Its Application to Seismic Ray Tracing

V. Pereyra, Caracas, Venezuela

Abstract

We give first a historical account of the various stages of development of iterated deferred corrections software, mainly for ordinary two-point boundary value problems, but mentioning also some work on partial differential equations. Then we describe the latest code on the PASVA series (No. 4), which extends the earlier one to problems with discontinuous data and mixed systems of differential and algebraic conditions. Finally, an example of application of this code to two-point ray tracing on piecewise continuous media is given.

1. Introduction

During the Winter of 1972–1973, while enjoying the hospitality of Gene Golub at Stanford University, I wrote a Report [36] that contains in great detail many of the ingredients of current programs for solving nonlinear boundary value problems for ordinary differential equations, by means of iterated deferred corrections and automatic grid refinements.

The problem class considered in that Report was fairly restricted – smooth single second order problems of the form $y''=f(t,y)$, $y(a)=\alpha$, $y(b)=\beta$ – but both the theory and explicit practical implementation of iterated deferred corrections, asymptotic error estimations, and automatic (uniform) mesh refinements, with FORTRAN programs and numerical results, were included.

One of these programs was later improved by Daniel and Martin [12], and constitutes probably one of the most efficient and robust pieces of software for solving the class of problems for which it was designed.

One important aspect of modern solvers, automatic non-uniform mesh refinements, was not considered until later in collaboration with Granville Sewell. In [5] automatic and fast generation of weights for high order approximation to ordinary differential operators were introduced. This is an essential part of iterated deferred (and some implementations of defect) corrections, and also for high order interpolation between different meshes, and it has to be considered carefully since it is one of the most time consuming operations in these procedures.

After the Stanford Report was finished as a pilot project, we were ready to tackle general first order systems, for which a complete discretization theory had been

published in H. B. Keller's book [18]. The first program, PASUNI (also known as SYSSOL), was written under my direction as an undergraduate thesis by M. Lentini, at Universidad Central de Venezuela [22, 23, 37]. This had essentially the same structure as the program in [36], extended to general first order systems of the form:

$$y' = f(t, y), \quad B_a y(a) + B_b y(b) = c.$$

The basic discretization used was the trapezoidal rule on a uniform mesh. Iterated deferred corrections and global error estimations were incorporated. The error estimations helped to decide if higher order corrections were pertinent, and if not, an automatic halving of the mesh was triggered. Of course, we performed a compatible high order interpolation of the last good solution on the coarse mesh to initiate the approximations on the finer one.

For solving the discrete nonlinear equations we used Newton's method, and the resulting linear equations were solved by a recursive algorithm described in Keller's book. This algorithm turned out to be unstable, and it was replaced in later versions. The nonlinear equations were solved only to a precision compatible with the expected global error.

This program worked well within a restricted class of smooth problems for which uniform meshes were adequate. However, it was clear that many important problems were left out, and that the algorithm was not efficient for problems with sharp gradients and boundary or internal layers. These problems required obviously nonuniform meshes, and the success of variable order – variable step codes for IVP's further suggested that approach.

Keller's theory allowed non-uniform meshes, and the fact of using a one step method for first order equations made their implementation fairly simple. The automatic fast deferred correction generator could, and was generalized to non-uniform meshes, and the only remaining obstacle for a successful generalization of PASUNI was the automatic selection of the non-uniform mesh.

So we came naturally to the question of automatic and efficient non-uniform mesh generation. A set of circumstances, including important conversation with H. B. Keller and C. De Boor, and the fact that G. Sewell was working at the time in Caracas, led us to the results reported in [45] and to the first complete implementation of a variable order (via IDC), variable step algorithm with asymptotical global error estimation for first order systems with nonlinear boundary conditions: PASVAR [25]. There, we also made comparisons including an implementation of Successive Richardson Extrapolations [27]. PASVAR still used the unstable solver for the linear equation systems. It is good to point out here that the instability of this algorithm was not catastrophic, since due to the adaptive characteristics of the code, and those of Newton's method itself, a lot of ill-conditioning would get masked and could be tolerated.

However, it was not too elegant, and one could really get into troubles, so the linear equation solver was replaced, in a major overhauling of the code, by the one in Varah [61]. See also [60, 62, 59]. By the way, Bruce Simpson of the University of Waterloo was Lam's advisor, and has also worked on mesh refinements [52], and

has therefore been present in many of these developments without receiving, in our opinion, as wide a recognition as his work deserves.

The first widely distributed code of the PASVA series, PASVA2, was written during the Summer of 1975, while we were visiting the Lawrence Berkeley Laboratory by invitation of Paul Concus. After that, researchers around the world and ourselves used PASVA2 in a wide number of applications, a number of which stem from our colleagues at the Applied Mathematics Department of Caltech, where we spent the period from 1976 to 1978. This combined experience resulted in some further modifications and improvements, and in 1977 appeared PASVA3 [39], which was eventually incorporated to the IMSL, Harwell and NAG Libraries (adapted respectively by G. Sewell, I. Duff and I. Gladwell), and to those of a number of Universities (adapted by J. Bolstad and R. Le Veque among others) and Research Centers.

In this paper we would like to report on the latest version of our code PASVA4 [26], that extends the applicability of the earlier ones to problems with discontinuous right hand sides, and also allows for the simultaneous solution of differential equations with additional algebraic equations and parameters.

This type of problems could be solved by PASVA3 and similar programs like COLSYS (Ascher, Christiansen and Russell [11]), only by resorting to some ingenious tricks, some of which have been collected and described in detail in [2, 40]. Unfortunately, this indirect approach tended to be both cumbersome and inefficient. PASVA4 was designed with the desire to facilitate and make more efficient the solution of those problems, preserving as much as possible upper compatibility with PASVA3. The main motivation for making this considerable step forward in complexity was the work I had been doing in recent years with W. H. K. Lee, H. B. Keller and G. Wejcik on two-point ray-tracing and its application to a variety of problems in computational geophysics [41 – 44, 46], and also by the enthusiasm of C. Wilts, of the Electrical Engineering Department of Caltech, who was very interested in using such a code for modelling the spin wave resonances in ferromagnetic materials, which had an important bearing in understanding the mechanisms of production of bubble memories [57].

The code was essentially finished in 1980, and it has been undergoing tests on practical problems before is released to the general public. One of these applications is our new ray-tracing code for geological regions with material interfaces of which I will talk in Section 6. More recently, Markovitch *et al.* have used PASVA4 with great success to model semi-conductor behaviour, that involves problems with boundary layers, infinite intervals, singularities, and a few other odds and ends [63]. Some results which were generously give to us by the authors are shown in [26].

Finally, I would like to remark that our work has not been restricted to ODE's: we have applied IDC techniques to elliptic problems, both on rectangular as in general curved regions, from very early on [31, 32, 35], including fairly non-trivial theoretical results [38]. In [31, 32] we dealt with the problem of nested iterations, which has resurfaced in recent times. In [37] we have presented numerical results showing the combination of fast elliptic solvers with deferred corrections.

All this work has obvious connections with multigrid algorithms, antecedes by many years defect correction developments, and specially in those applications where the two procedures are identical may still be useful in new research.

2. Discontinuous Interfaces at Known Locations

We consider first problems of the form:

$$\begin{aligned} y'-f(t,y)&=0 \\ g(y(a),y(b))&=0 \end{aligned} \tag{2.1}$$

where $f(t,y)$ has $k\geq 0$ jump discontinuities with respect to t at the known locations:

$$a<d_1<d_2\ldots<d_k<b\,. \tag{2.2}$$

Here y,f,g are m-dimensional vector functions, and we assume that the problem has an isolated solution $y(t)$, which may be discontinuous at the (d_j), but that we assume to have one-sided limits $y(d_j^-)$ and $y(d_j^+)$.

In order for this locally unique solution to exist it is necessary that we have sufficient additional interface conditions to complete the formulation of the problem:

$$D_j(y\,d_j^-),\,y(d_j^+))=0,\quad j=1,\ldots,k, \tag{2.3}$$

where the D_j's are also m-vectors.

Actually problems of this type were considered on an experimental basis in our first code PASUNI, as it was reported in [23]. The obvious limitation in using finite difference approximations in this situation is that care has to be taken not to straddle the discontinuities. This requirement constraints the points d_j to be mesh points in all the meshes considered. Since we will use the trapezoidal rule as the basic discretization, quantities like y,f, etc. may be two-valued at those points.

Thus, we consider non-uniform meshes $\pi=\{t_i\}$, $i=1,\ldots,n$, satisfying:

$$a=t_1<t_2<\ldots<t_{i_j}=t_{i_j+1}<t_{i_j+2}<\ldots<t_n=b \tag{2.4}$$

where the subindices $\{i_j\}$, $j=1,\ldots,k$, mark the positions on the mesh of the discontinuity points d_j. We have introduced k pseudo-mesh-intervals $[t_{i_j},t_{i_j-1}]$ of zero length, in order to accomodate later on the left and right limits of all discontinuous quantities. For completeness, we set $i_0=0$, $i_{k+1}=n$, and therefore we can define the $(k+1)$ subintervals of continuity by: $C_j=(t_{i_j+1},t_{i_j+1})$, $j=0,\ldots,k$.

On this mesh we consider the discretization of problem (2.1)–(2.3).

$$\begin{aligned} E_i=y_{i+1}-y_i-1/2\,h_i(f_i+f_{i+1})&=0,\quad i=1,\ldots,n-1,\ i\neq i_j, \\ g(y_1,y_n)=0,\quad D_j(y_{i_j},y_{i_j+1})&=0,\quad j=1,\ldots,k, \end{aligned} \tag{2.5}$$

where E_i,y_i,f_i are m-vectors. y_i is the sought approximation to $y(t_i)$, while $f_i=f(t_i,y_i)$, and $h_i=t_{i+1}-t_i$.

Let us call $F(y)=0$ to the continuous problem (2.1)–(2.3), and let us classify the boundary conditions in initial, coupled and final ones:

$$g_1(y_1)=0, \quad g_2(y_1, y_n)=0, \quad g_3(y_n)=0,$$

where the dimensions of g_1 are respectively p, r, and q.

Obviously $p+r+q=m$, and we also require $1<p<m$, so that the problem is a genuine boundary value one. Actually the coupled conditions g_2 could be multipoint without adding major difficulties.

We define the ordering of unknowns and equations, and the discrete operator F_π as:

$$Y=[y_1^T, y_2^T, \ldots, y_n^T]^T,$$

$$F_\pi(Y)=[g_1^T, E_1^T, \ldots, E_{i_1}^T-1, D_1^T, E_{i_1}+1, \ldots, E_{i_k}-1, g_2^T, g_3^T]^T=0 \qquad (2.6)$$

where T means vector transposition.

We assume that $f(t, y)$ is C^M on $C_j \times (-\infty, \infty)$, with $M \geq 2$, for each $j=1, \ldots, k$. This implies that F_π is consistent of order 2 with F. We assume also that the D_j and g are twice continuously differentiable with respect to their arguments, and that $\partial D_j/\partial y_{ij+1}$ is non-singular. This implies that y_{ij+1} can also be solved in terms of y_{ij+1}, and therefore a solution satisfying initial conditions at $t=a$ can be continued accross the discontinuities. By standard arguments, the trapezoidal rule will be stable for the initial value problem, and the property gets transfered to the BVP as in [21].

3. PASVA4, Part I

With the choice of the ordering (2.6), the resulting nonlinear system has the same block structure as in the continuous case, and the same solver of [39] can be used.

This solver has two components. A Newton algorithm with step control to make it more robust, and special sparse solver for the resulting linear systems. This are well tested modules in PASVA3, and in order to diminish development costs it is a good idea to use them in this more general code. Clearly, still some substantial changes are necessary in order to intercalate the jump conditions D_j in the original data structures.

The main objective of defining the subintervals of continuity C_j however, is to avoid straddling the discontinuities during the processes of error estimation, deferred corrections, and mesh selection. All these steps can easily be restricted to each C_j, treating the mesh points near to the inner and outer boundaries via unsymmetric formulae. This is a case in point where it is not possible to extend the solution outside the interval of integration in order to be able to use the same correction formulae throughout. Although this deteriorates somewhat the asymptotic properties of the correction formulae, both practice and the work of Christiansen and Russell [11] show that an efficient and robust algorithm results.

Problems with discontinuities have been solved in the past with programs like PASVA3 and COLSYS by a technique called "multiplexing" (see [2, 17, 40]), which is essentially the same idea as that of Keller's formulation for multiple shooting. A copy of the differential system is solved in each interval C_j, which is first mapped into a common interval, say $[0, 1]$. Then this $m \times (k+1)$ system of differential equations is solved by means of a conventional program.

The dimensionality of the system affects most directly the cost of solving the linearized equations. As we showed for instance in [20], the number of arithmetic operations for solving an $m\times(k+1)$ system on a mesh with l points in $[0,1]$ is Ops. $=l\times(m\times(k+1))^3$.

Observe that, because of the multiplexing, the l points get mapped back into $l\times(k+1)$ mesh points on the original interval, since copies of the $[0,1]$ mesh will appear in each interval C_j. If we consider $n=l\times(k+1)$ mesh points, the new algorithm in PASVA4 will take Ops $=l\times(k+1)\times m^3$, with a saving of $(k+1)^2$ operations per solve of the linear equations (and a factor of $(k+1)$ saving in storage).

Furthermore, in the case of multiplexing, the mesh selected on $[0,1]$ will be copied to all the subintervals C_j, although its choice will correspond to the worst behaviour of the solution. If this bad behaviour is concentrated in only a small part of the region, and probably a particular sub-interval, as in the case of boundary or interior layers, PASVA4 will treat it independently, since there is no communication between the C_j in the mesh selection process, and it will not polute the other sub-intervals as with multiplexing. In fact, artificial sub-intervals can be introduced to isolate such regions and deal with them more efficiently, which amounts to simulating the independent variable stretchings common in singular perturbation techniques (as it has actually been done with excellent results in [63]).

4. Discontinuities at Unknown Locations

Consider now problem (2.1) – (2.3) in the case, fairly common in the applications, in which the location of the discontinuities is not known a priori. Let us call these unknowns positions $\{\tau_j\}$, $\tau_1<\tau_2<\ldots<\tau_k$.

We add for completeness $\tau_o=a$, $\tau_{k+1}=b$.

In order to complete the description of this problem it is necessary to introduce switching functions, which will indicate when a discontinuity is traversed. These are functions of the form $\phi_j(t,y)$, $j=1,\ldots,k$.

The surfaces in the (t,y) space defined by $\phi_j(t,y)=0$, are the boundaries separating the regions of continuity of the trajectory $y(t)$. Thus,

$$t\in C_j \quad \text{iff} \quad \text{(say)} \quad \phi_j(t,y(t))>0 \quad \text{and} \quad \phi_{j-1}(t,y(t))<0,$$

here we have defined $\phi_o(t,y)=t-a$, and $\phi_{k+1}(t,y)=t-b$.

Therefore, at the discontinuity points we must have the additional conditions

$$\phi_j(\tau_j,y(\tau_j))=0, \quad j=1,\ldots,k, \tag{4.1}$$

which will account for the additional unknown parameters τ_j.

To use PASVA4 in this problem we require the following pre-processing:

(I) Map each interval of continuity $C_j=(\tau_j,\tau_{j+1})$, $j=0,\ldots,k$, into (say) the known interval $[j,j+1]$.

This will introduce the unknown parameters τ_j into the differential equations, but now the resulting format will be that of Section 2. The treatment of the problem with

parameters and additional internal boundary conditions like (4.1) will be considered in the next Section.

Clearly, if this process is successful, we would have solved simultaneously for the unknown functions $y(t)$ and for the position of the discontinuities.

Currently we are investigating the feasibility of using these facilities of PASVA4 for solving nonlinear hyperbolic problems with propagating discontinuous fronts by shock tracking (see for instance [10]).

Observe that we must know a priori the order in which the switching functions will be activated, and of course the number of switching points.

The choice of the name switching is not casual, since these are the same type of functions that appear in optimal control problems, and PASVA4 can be used with profit in such problems if an Euler-Lagrange formulation is employed, and the above restrictions are satisfied.

5. PASVA4, Part II. Algebraic Parameters and Conditions

In the last Section we saw an instance of a problem with mixed differential and algebraic conditions. There are many more applications of this type, including free boundary, linear and nonlinear eigenvalue and control problems.

In the past, we (and many others) have solved these problems with available software by introducing artificial differential equations of the form $\lambda' = 0$, whose solutions would be the constant unknown parameters.

In the discrete version that is actually solved for, this artifice amounts to considering as many additional equations and unknowns as there are mesh points (for each λ), and therefore is fairly wasteful.

Thus, it is appropriate to consider special methods for problems of the form:

$$\begin{aligned} y' &= f'(t, y, \lambda) \\ g(y(a), y(b), \lambda) &= 0 \\ h((y(a), y(b), \lambda) &= 0, \end{aligned} \tag{5.1}$$

where as before, f and y are m-dimensional, while λ is l-dimensional. g and h are respectively m and l-vectors. We assume that this problem has an isolated solution pair $(y^*(t), \lambda^*)$ that we are seeking.

Let $Y(t)$ be the fundamental solution of system (5.1) linearized around (y^*, λ^*). A sufficient condition for (y^*, λ^*) to be isolated is that the matrices:

$$\bar{Q} = [g_{y(a)}(y^*(a), y^*(b), \lambda^*) + Y(b) g_{y(b)}(y^*(a), y^*(b), \lambda^*)]$$

and

$$h_\lambda(y^*(a), y^*(b), \lambda^*)$$

be nonsingular.

We consider now discretization (2.5) for the extended system of equations (5.1) and unknowns $[y(t), \lambda]$. We shall apply Newton's method to this enlarged system (see (2.6)):

$$F_\pi \begin{bmatrix} Y \\ \lambda \end{bmatrix} = 0 \tag{5.2}$$

$$h(y_1, y_n, \lambda) = 0$$

That is, starting from an initial guess $[Y^0, \lambda^0]$, we iterate according to

1) For $\nu = 0, 1, \ldots$, solve

$$\Omega \begin{bmatrix} \Delta Y \\ \Delta \lambda \end{bmatrix} = \begin{pmatrix} \mathbb{A}^\nu & \beta^\nu \\ \gamma^\nu & h_\lambda^\nu \end{pmatrix} \begin{bmatrix} \Delta Y \\ \Delta \lambda \end{bmatrix} = \rho^\nu \tag{5.3}$$

where $\mathbb{A}^\nu$ is the Jacobian of F_π with respect to Y, evaluated at (Y^ν, λ^ν) $\beta^\nu = F_{\pi\lambda}$, $\gamma^\nu = (h_{y_1}, 0, \ldots, h_{y_n})$, and ρ^ν is the negative residual.

2) Correct with step control, $0 < \mu^\nu < 1$:

$$\begin{pmatrix} Y^{\nu+1} \\ \lambda^{\nu+1} \end{pmatrix} = \begin{pmatrix} Y^\nu \\ \lambda^\nu \end{pmatrix} + \mu^\nu \begin{pmatrix} \Delta Y \\ \Delta \lambda \end{pmatrix}.$$

We observe that $\mathbb{A}^\nu$ is exactly the same matrix as in the case without parameters, and therefore it would be convenient to be able to use the same linear solver we already have developed.

Unfortunately, the fact that the block matrix Ω is non-singular (obtained through standard arguments, say as in [19], Theorem 2.31), does not automatically guarantee that $\mathbb{A}^\nu$ will be non-singular. In fact, it may be necessary to interchange some of the equations in g and h in order to achieve this nonsingularity. That there is always such a choice stems from the fact that rank

$$\begin{pmatrix} \mathbb{A}^\nu \\ \gamma^\nu \end{pmatrix}$$

must be full for the whole matrix to be nonsingular.

We will assume that these interchanges have been made beforehand, and therefore that $\mathbb{A}^\nu$ itself is non-singular. We propose then to solve (5.3) by two-by-two block Gaussian elimination, which will involve the solution of $(l+1)$ systems with the sparse matrix of coefficients $\mathbb{A}^\nu$, and a small $(l \times l)$ full system with matrix $S^\nu = (h_\lambda^\nu - \gamma^\nu \mathbb{A}^{-\nu} \beta^\nu)$, the Schur complement of $\mathbb{A}^\nu$, which is non-singular from the fact that Ω and $\mathbb{A}^\nu$ are non-singular.

This then completes the extension of PASVA3 to problems with discontinuous righthand sides for systems of mixed differential and algebraic equations and additional parameters. Conditions g and h can be multi-point, provided the internal boundaries are contained in the set of discontinuity points.

With this extension we can now solve the problem of Section 4 fairly efficiently, as it will be shown in detail on a non-trivial example in the next Section.

We should mention before finishing this Section, that the linear equation solver of PASVA3 actually performs a sparse LU decomposition of $\mathbb{A}$, and it is separated in the two usual phases of decomposition and solution. Thus, it is very cheap to solve successively systems with the same matrix $\mathbb{A}$ once this has been decomposed. This makes the above algorithm fairly efficient, and it is also used in the deferred correction process by freezing the Jacobian after the first solution on a given mesh is obtained. This strategy anticipated the excellent results of Böhmer [65, 66] which rigorously proves that it will work.

Other applications of this facility occur in the continuation process (to calculate Euler initial approximations), in calculating the geometrical spreading for seismic rays [46], and in general, in calculating derivatives of the solution with respect to parameters as in [41, 43].

6. Three Dimensional Two-point Ray Tracing

Wave propagation on general isotropic media is important in many applications in acoustics, seismology, seismic prospecting and medicine, to name a few.

Geometrical optics provides a good, economical, high frequency approximation to the wave equation, that is used in practice because it gives the information required at low cost, compared to direct full field integration. Besides of its predictive value, this type of calculation furnishes a useful interpretative tool for complex data analysis or even to analyze the results of a full wave calculation (which can be as complex as field data itself).

Instead of calculating the full field, one computes a discrete set of rays, from which it is possible to derive very useful information in the form of ray-paths, travel time curves and synthetic seismograms.

Rays are the orthogonal trajectories to the wave fronts. They satisfy the set of second order equations:

$$\frac{d}{ds}\left[u(\underline{\eta})\, d\underline{\eta}/ds\right] = \nabla u(\underline{\eta}), \tag{6.1}$$

where $\underline{\eta} = (\eta_1, \eta_2, \eta_3)$ are the Cartesian coordinates of the ray-path, s is the arc-length along it, and $u(\underline{\eta})$ is the refraction index.

$u(\underline{\eta}) = v(\underline{\eta})^{-1}$, where $v(\underline{\eta})$ is the velocity of the waves being considered, which depends upon the ellastic properties of the materials that compose the medium.

In particular, we will consider piece-wise continuous, inhomogeneous, isotropic media, where $v(\underline{\eta})$ is smooth except across surfaces which are boundaries between sharply different materials; these shall be called material interfaces.

There are several problems of interest associated with equations (6.1). Given a source (earthquake, explosion, mechanical vibrator), one may want to trace through the medium the rays that originate from this source. This will be an initial value problem, and we shall refer to it as "shooting rays".

Another important class of problems arises when we are interested only on those rays that starting from the source arrive to given locations (seismophone, seismograph, ...). This is a two-point problem for (6.1), and if the ray traverses material interfaces we will have a problem with the full generality considered in the last Section.

In order to use PASVA4 we need to write (6.1) in first order form. A good set of auxiliary variables is given by:

$$w_{2i-1}=\eta_i,\quad w_{2i}=u(\underline{\eta})\,d\eta_i/ds,\quad i=1,2,3,$$

and by using them (6.1) reduces to:

$$\begin{aligned} dw_{2i-1}/ds &= v(\underline{\eta})\,w_{2i} \\ dw_{2i}/ds &= \hat{c}\,u/\hat{c}\,\eta_i \quad i=1,2,3. \end{aligned} \tag{6.2}$$

The interface surfaces are defined by means of the switching functions $\phi(\underline{\eta})=0$. These surfaces partition the region in subregions. In order to define completely a ray we must give its "signature". This is an ordered sequence of interfaces that we expect the ray to traverse, and a corresponding ordered sequence of subregions. In other words, we must postulate a priori which and how many switching functions will be activated, and in which order. Also, by knowing in which sub-region is the ray, between each pair of interface crossings, we know which material, and therefore which velocity to use.

All this is necessary because usually there are many rays between two given points. Of course, we may state in this fashion a problem with no solution (the receiver is in the shadow zone respect to the source), in which case we except the algorithm to fail graciously, indicating the presence of such a situation.

At interfaces, the rays can be transmitted to a neighbouring region or they can be reflected into the same region where they were. The plane and angle between the incomming and outgoing rays are governed by Snell's law, that establishes: "The derivatives of the wave fronts in the directions of an obstacle's tangents must be equal".

If we call $\nabla\psi^{\nu}=(w_2^{\nu}, w_4^{\nu}, w_6^{\nu})$, with $\nu=o$ for the incomming ray and $\nu=t, r$ for transmitted or reflected rays we have, after a few calculations, that Snell's law translates into the following interface condition:

$$\nabla\psi^{\nu}=\nabla\psi^{o}+\{\pm]\sqrt{u_{\nu}^2-u_o^2+\langle\nabla\psi^{o},\underline{\mu}\rangle^2}-\langle\nabla\psi^{o},\underline{\mu}\rangle\}\,\mu, \tag{6.3}$$

where $\langle\,,\rangle$ denotes vector inner product, $\underline{\mu}=\nabla\phi(\underline{\eta})/\|\nabla\phi(\underline{\eta})\|$, and the $+$ sign corresponds to transmitted rays ($-$ to reflected ones). These three conditions, together with $\underline{\eta}(s)^{o}=\underline{\eta}(s)^{\nu}$, give six interface conditions for each of the transmitted or reflected rays.

Clearly these crossing will occur at unknown values of the independent variable s, and therefore we must perform the transformations indicated in Section 4. In this way, the lenghts of the segments of rays between interface crossings will appear as unknown parameters in the differential equations.

If we count carefully we shall see that there is still one missing condition (corresponding to the last segment length); this is supplied by

$$\| \nabla \psi(0) \|_2^2 = 1 \tag{6.4}$$

which establishes that our independent variable is actually arc length.

Shooting

In the case where we give the initial position $\underline{\eta}(0)$, and initial take-off direction $\underline{\eta}'(0) = \nabla \psi(0) \cdot u(0)$, then the problem would be an initial value one, and it would be completely determined by the interface conditions. However, we would still like to use PASVA4 for this problem, although it is not a BVP.

To do that, we make sure first that the given initial direction is such that (6.4) is satisfied. Thus we can drop that condition and impose a condition at the far end, say "integrate up to a given interface" or "up to given total arc length S", etc. With this we will have a two-point problem that can be solved by PASVA4.

Two-Pointing

When we want a ray between a pair source-station, we have given the six conditions $\underline{\eta}(0) = \underline{\eta}_{so}$ and $\underline{\eta}(S) = \underline{\eta}_{st}$. With (6.4) and the interface conditions the problem is completely determined.

The insistence in formulating the shooting case also as a two-point problem is not simply because of our desire to promote PASVA4. In fact, as we have just seen, the difference between one problem and the other is very small, and a simple logical switch allows to pass from one to the other, and thus avoids the need of an special formulation and the use of another large code.

Example:

We consider a three dimensional non-homogeneous medium, with velocity varying only with depth (η_3), but with a material discontinuity described by the surface:

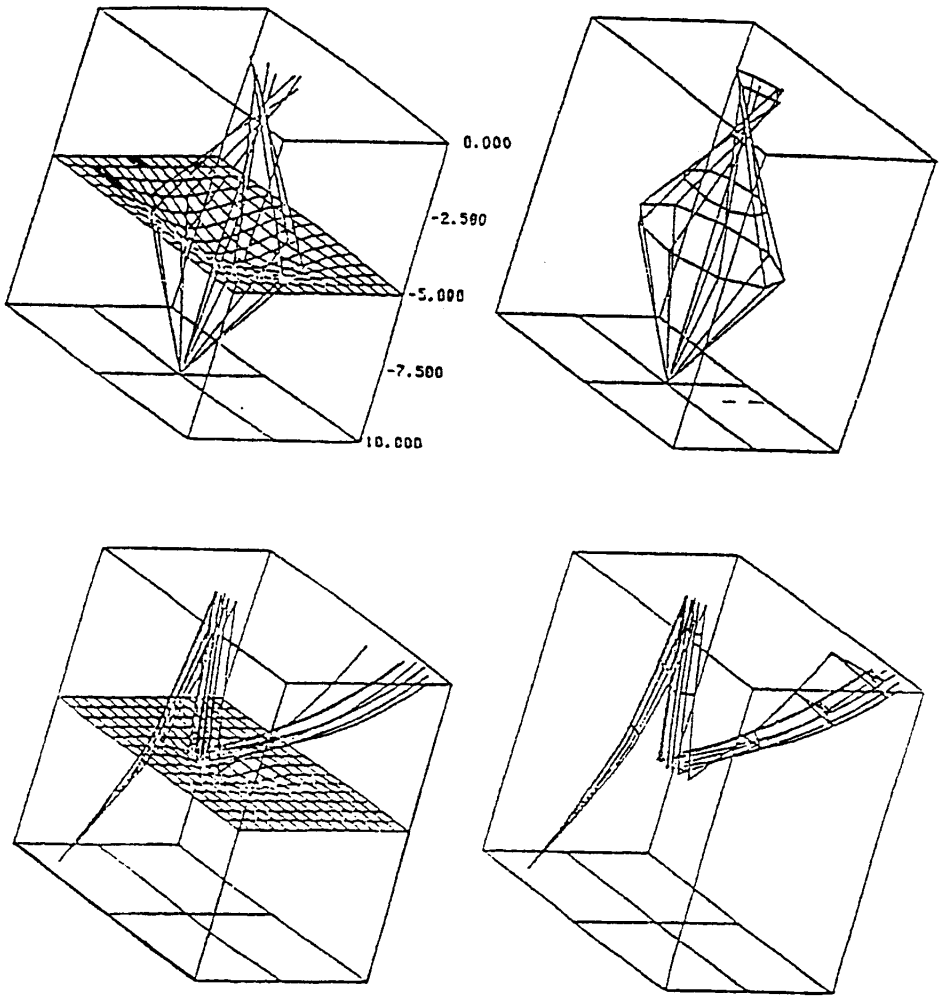
$$\phi(\underline{\eta}) = \eta_3 - d - c\,x\,\{1 - \cos[2\pi(\eta_1 - \omega/2)/\omega]\} \times \{1 - \cos[2\pi(\eta_2 - \omega/2)/\omega]\} \tag{6.5}$$

where the parameters c, d, ω determine the shape of this cos-like depression: d is an horizontal background, $(4c + d)$ is the minimum point of the surface, and $\omega/2$ is where this surface tapers off in the η_1 and η_2 directions (see Fig. 1).

The velocities above and below the interface are respectively

$$v_1 = \alpha_1 + \beta_1 \eta_3, \quad v_2 = \alpha_2 + \beta_2 \eta_3.$$

We show in Figs. 1–4, the graphical output of a 3 d ray-tracing code based on the ideas described earlier, for various choices of the parameters and options. Typically, each ray takes under 0.1 seconds of CPU on a VAX 11/780.



Figs. 1–4

7. Future Developments

I would like to close this presentation of past and current software developments of iterated deferred corrections algorithms by pointing out to some areas in which further contributions would be welcomed.

A bridge between the mild and tame boundary layer problems and the really tough ones would be quite useful. This might appear as a program that switches from one technique to another as necessary, in the way it is been done in stiff IVP's at the

present time. In that way one may try to couple current efficient solvers with techniques like those of Kreiss and Nichols [72] for reliably choosing meshes in hard cases.

Bifurcation diagrams are needed in many applications. The techniques of Keller and others should be integrated to current or new software, in order to provide a ductile package, which should be interactive and with graphic output. We have certainly used PASVA3 in this fashion, but no formal software package for the general public has been issued so far.

Vectorization and other advanced architectures should be considered. R. Schreiber of Stanford University has done some preliminary work in vectorizing PASVA3 with the aim of obtaining a vectorized ray tracing code which would produce (ideally) a bundle of rays for the price of one. More efforts in this direction may have an impact on PDE applications also.

At the other end of the spectrum, specialized short algorithms may be valuable for micro-applications, i.e. applications working on stand-alone personal stations. We are at the present time exploring the possibilities of implementing a multigrid type algorithm for some one-dimensional applications, taking advantage of the compact implementation of this method. Observe that here the main issue may not be the outmost efficiency of the method.

References

[1] Ascher, U., Christiansen, J., Russell, R. D.: A collocation solver for mixed order systems of BVP's. Math. Comp. *33*, 659–679 (1978).

[2] Ascher, U., Russell, R. D.: Reformulation of BVP's into "standard" form. SIAM Rev. *23*, 238–254 (1981).

[3] Berger, M. J.: Adaptive mesh refinements for hyperbolic PDE's. STAN-CS 82-924. Comp. Sc. Dept. Stanford Univ., California, U.S.A., 1982.

[4] Berger, M. J., Oliger, J.: Adaptive mesh refinement for hyperbolic PDE's. NA-83-02, Comp. Sc. Dept. Stanford Univ., California, U.S.A., 1983.

[5] Björck, A., Pereyra, V.: Solution of Vandermonde systems of equations. Math. Comp. *24*, 893–904 (1970).

[6] Bolstad, J.: PhD Thesis, Stanford Univ. California, U.S.A., 1982.

[7] de Boor, C.: Good approximation by splines with variable knots. Lecture Notes in Mathematics *363*, 12–20 (1973).

[8] Burchard, H. G.: Splines (with optimal knots) are better. J. Appl. Anal. *3*, 309–319 (1974).

[9] Childs, B., Scott, M., Daniel, J. W., Denman, E., Nelson, P. (eds.): Codes for BVP's in ODE's. (Lecture Notes Comp. Sc., Vol. 76.) Berlin-Heidelberg-New York: Springer 1979.

[10] Corless, R. M.: A feasibility study of the use of COLSYS in the solution of systems of PDE's. CS-82-32. Dept. Appl. Math. Univ. Waterloo, Ontario, Canada, 1982.

[11] Christiansen, J., Russell, R. D.: Deferred corrections using non-symmetric end formulas. Numer. Math. *35*, 21–33 (1980).

[12] Daniel, J. W., Martin, A. J.: Numerov's method with deferred corrections for two-point boundary-value problems. SIAM J. Numer. Anal. *14*, 1033–1050 (1977).

[13] Daniel, J. W., Pereyra, V., Schumaker, L. L.: Iterated deferred corrections for IVP's. Acta Cient. Venezolana *19*, 128–135 (1968).

[14] Fox, L.: Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations. Proc. Royal Soc. (London) *A 190*, 31–59 (1947).

[15] Hackbusch, W., Trottenberg, U. (eds.): Multigrid Methods. (Lecture Notes in Mathematics, Vol. 960.) Berlin-Heidelberg-New York: Springer 1982.

[16] Henrici, P.: Discrete Variable Methods in ODE's. New York: Wiley 1962.
[17] Itoh, K.: On the numerical solvability of 2 PBVP's in a finite Chebyshev series for piece-wise smooth differential systems. Mem. Numer. Math. *2*, 45–67 (1975).
[18] Keller, H. B.: Numerical Methods for 2PBVP's. Waltham, Mass.: Blaisdell 1968.
[19] Keller, H. B.: Numerical Solution of TPBVP's. Reg. Conf. Series Appl. Math. *24*, SIAM, Philadelphia, Pa., U.S.A., 1976.
[20] Keller, H. B., Pereyra, V.: Difference methods and deferred corrections for OBVP's. SIAM J. Numer. Anal. *16*, 241–259 (1979).
[21] Keller, H. B., White, A. B.: Difference methods for BVP's in ODE's. SIAM J. Numer. Anal. *12*, 791–801 (1975).
[22] Lentini, M.: Correcciones diferidas para problemas de contorno en sistemas de ecuaciones diferenciales ordinarias de primer orden. Pub. 73-04, Dpto. Comp., Fac. Ciencias. Univ. Central, Caracas, Venezuela, 1973.
[23] Lentini, M., Pereyra, V.: A variable order finite difference method for nonlinear multipoint BVP's. Math. Comp. *28*, 981–1003 (1974).
[24] Lentini, M., Pereyra, V.: Boundary problem solvers for first order systems based on deferred corrections. In: Numerical Solution of BVP's for ODE's (Aziz, A. K., ed.). New York: Academic Press 1975.
[25] Lentini, M., Pereyra, V.: An adaptive finite difference solver for nonlinear two-point boundary problems with mild boundary layers. SIAM J. Numer. Anal. *14*, 91–111 (1977).
[26] Lentini, M., Pereyra, V.: PASVA4: An ordinary boundary solver for problems with discontinuous interfaces and algebraic parameters. Mat. Aplicada e Comp. *2*, 103–118 (1983).
[27] Lopez, H., Ruiz, L.: Extrapolaciones sucesivas para problemas de contorno en sistemas no-lineales de EDO's. Trabajo Esp. de Grado. Depto. Comp., Fac. Ciencias. Univ. Central, Caracas, Venezuela, 1974.
[28] Pearson, C. E.: On a differential equation of boundary layer type. J. Math. Phys. *47*, 134–154 (1968).
[29] Pereyra, V.: The difference correction method for non-linear TPBVP's. Techn. Rep. CS18, Comp. Sc. Dept., Stanford Univ. California, U.S.A., 1965.
[30] Pereyra, V.: The correction difference method for non-linear BVP's of class. M. Rev. Unión Matemática Argentina *22*, 184–201 (1965).
[31] Pereyra, V.: Highly accurate discrete methods for nonlinear problems. PhD Thesis, Comp. Sc. Dept., Univ. Wisconsin, Madison, Wis., U.S.A., 1967.
[32] Pereyra, V.: Accelerating the convergence of discretization algorithms. SIAM J. Numer. Anal. *4*, 508–533 (1967).
[33] Pereyra, V.: Iterated deferred corrections for nonlinear operator equations. Numer. Math. *10*, 316–323 (1967).
[34] Pereyra, V.: Iterated deferred corrections for nonlinear BVP's. Numer. Math. *11*, 111–125 (1968).
[35] Pereyra, V.: Highly accurate numerical solution of casilinear elliptic BVP's in n dimensions. Math. Comp. *24*, 771–783 (1970).
[36] Pereyra, V.: High order finite difference solution of differential equations. STAN-CS-73-348, Comp. Sc. Dept. Stanford Univ., California, U.S.A., 1973.
[37] Pereyra, V.: Variable order variable step finite difference methods for NLBVP's. (Lecture Notes in Mathematics, Vol. 363, pp. 118–133.) Berlin-Heidelberg-New York: Springer 1974.
[38] Pereyra, V., Proskurowski, W., Widlund, O.: High order fast Laplace solvers for the Dirichlet problem on general regions. Math. Comp. *31*, 1–16 (1977).
[39] Pereyra, V.: An adaptive finite difference FORTRAN program for first order. NLOBP's. In [9], pp. 67–88 (1979).
[40] Pereyra, V.: Solución numérica de ecuaciones diferenciales con condiciones de frontera. Acta Cient. Venezolana *30*, 7–22 (1979).
[41] Pereyra, V.: Two-point ray tracing in heterogeneous media and the inversion of travel time data. In: Computing Methods in Appl. Sc. and Eng. (Glowinski, R., Lions, J. L., eds.), pp. 553–570. Amsterdam: North-Holland 1980.
[42] Pereyra, V.: Constrained ill-conditioned non-linear least squares. (In preparation.)
[43] Pereyra, V., Keller, H. B., Lee, W. H. K.: Computational methods for inverse problems in geophysics: inversion of travel time observations. Physics of the Earth and Planetary Interiors *21*, 120–125 (1980).

[44] Pereyra, V., Lee, W. H. K., Keller, H. B.: Solving two-point ray tracing problems in a heterogeneous medium. Bull. Seism. Soc. America *70*, 79–99 (1980).
[45] Pereyra, V., Sewell, G.: Mesh selection for discrete solution of BVP's in ODE's. Numer. Math. *23*, 261–268 (1975).
[46] Pereyra, V., Wojcik, G.: Interactive seismic ray tracing on complex geological structures. (In preparation.)
[47] Russell, R. D.: Mesh selection methods. In [9], pp. 228–242 (1979).
[48] Russell, R. D., Christiansen, J.: Adaptive mesh selection strategies for solving BVP's. SIAM J. Numer. Anal. *15*, 59–80 (1978).
[49] Sewell, E. G.: Automatic generation of triangulations for piecewise polynomial approximation. PhD Thesis. Purdue Univ. Indiana, U.S.A., 1972.
[50] Sewell, E. G.: An adaptive computer program for the solution of $\mathrm{Div}(P(x,y)\,\mathrm{Grad}\,u)=f(x,y,u)$ on a polygonal region. In: MAFELAP II (Whiteman, J. R., ed.), pp. 543–553. New York: Academic Press 1976.
[51] Sewell, E. G.: TWODEPEP: A small general purpose finite element program. Techn. Rep. 8102, IMSL. Houston Texas, U.S.A., 1981.
[52] Simpson, R. B.: Automatical local refinements for irregular meshes. Research Rep. CS-78-19. Univ. Waterloo, Ontario, Canada, 1978.
[53] Stetter, H. J.: Asymptotic expansions for the error of discretization algorithms for nonlinear functional equations. Numer. Math. *7*, 18–31 (1965).
[54] Stetter, H. J.: Global error estimation in Adams PC.-codes. ACM Trans. Math. Software *5*, 415–430 (1979).
[55] White, A. B.: On the numerical solution of initial/boundary value problems in one space dimension. SIAM J. Numer. Anal. *19*, 683–697 (1982).
[56] White, A. B.: On selection of equidistributing meshes for TPBVP's. SIAM J. Numer. Anal. *16*, 472–502 (1979).
[57] Wilts, C.: Ferromagnetic resonance equations for implanted films. Manuscript. Dept. Electrical Eng., Caltech. Pasadena, Ca., U.S.A., 1980.
[58] Zadunaisky, P. E.: On the estimation of errors propagated in the numerical integration of ODE's. Numer. Math. *27*, 21–40 (1976).
[59] Keller, H. B.: Accurate difference methods for NLTPBVP's. SIAM J. Numer. Anal. *11*, 305–320 (1974).
[60] Lam, D. C. L.: Implementation of the Box scheme and model analysis of diffusion-convection equations. PhD Thesis, Univ. Waterloo, Ontario, Canada (1974).
[61] Varah, J. M.: On the solution of block tridiagonal systems arising from certain finite-difference equations. Math. Comp. *26*, 859–868 (1972).
[62] Varah, J. M.: Alternate row and column elimination for solving certain linear systems. SIAM J. Numer. Anal. *13*, 71–75 (1976).
[63] Markowich, P. A., Ringhofer, Chr. A., Selberherr, S., Langer, E.: A singularly perturbed BVP modelling a semiconductor device. Techn. Summ. Rep. No. 2388. Math. Res. Center, Univ. Wisconsin, Madison, U.S.A., 1982.
[64] Lentini, M., Osborne, M. R., Russell, R. D.: The close relationship between methods for solving TPBVP's. SIAM J. Numer. Anal. (Submitted for publication.)
[65] Böhmer, K.: Discrete Newton methods and iterated defect corrections. Numer. Math. *37*, 167–192 (1981).
[66] Allgöwer, E. L., Böhmer, K., Mc Cormick, S.: Discrete correction methods for operator equations. In: Numerical Solution of Nonlinear Equations (Lecture Notes in Mathematics, Vol. 878), pp. 31–97. Berlin-Heidelberg-New York: Springer 1981.
[67] Lindberg, B.: Error estimation and iterative improvement for discretization algorithms. BIT *20*, 486–500 (1980).
[68] Frank, R., Hertling, J., Ueberhuber, C.: An extension of the applicability of iterated deferred corrections. Math. Comp. *31*, 907–915 (1977).
[69] Stetter, H. J.: The defect correction principle and discretization methods. Num. Math. *29*, 425–443 (1978).
[70] Zadunaisky, P. E.: A method for the estimation of errors propagated in the numerical solution of a system of ODE's. Proc. Astron. Union Symp., No. 25. New York: Academic Press 1966.