Acta Cient. Venezolana 19: 128-135, 1968.

Iterated deferred corrections for initial value problems

JAMES W. DANIEL, VÍCTOR PEREYRA * and LARRY L. SCHUMAKER

Sponsored by the Mathematics Research Center, University of Wisconsin, Madison, Wisconsin, under Contract Nº: DA-31-124-ARO-D-462.

1. INTRODUCTION

THE PURPOSE of this paper is to study the application of iterated deferred corrections (IDC) to the numerical solution of initial value problems (IVP's) for systems of ordinary differential equations. The idea of difference correction was introduced by Fox [4]; the extension of IDC for operator equations has been examined in detail by Pereyra in [7, 8, 9] where a historical account with references to most of the earlier work is given.

Our interest in the problem was stimulated by a method of Loscalzo and Schoenberg [6] for solving IVP's; we began by studying the effect of substituting numerical approximations for certain explicit differentiation operations in their formulas. This turned out to be equivalent to applying IDC to a discrete variable method based on a Hermite expansion and suggested that IDC could be employed to derive high order schemes from other discrete variable formulas.

In the next section we state the basic problem and present the general theory of IDC for initial value problems. In later sections we describe a number of particular discrete variable methods which are amenable to IDC, show how to implement the IDC procedure for these specific formulas, and discuss our numerical experience with them.

2. STATEMENT OF THE PROBLEM AND GENERAL THEORY

We wish to solve numerically the initial value problem for a system of ordinary differential equations

$$\begin{aligned} y'(t) &= f(y(t)) \qquad\qquad t \in [a, b] \\ y(a) &= y_0, \end{aligned} \tag{2.1}$$

where y_0 is a prescribed initial value and $[a, b]$ is a fixed closed subinterval of the real line. We have restricted ourselves to autonomous systems for convenience since a general system can always be converted to one of the form (2.1) by introducing an equation for the time variable. Time dependent equations can be treated directly but the added notation merely clouds the relevant issues. For systems of equations y, y_0 and f are vectors, but in the sequel we shall not take special pains to remind the reader of this fact.

Throughout the following we asume that $f \in C^\infty$ for $-\infty < \|y\| < \infty$, although the analysis can be carried out whenever f is sufficiently differentiable. If a priori estimates for the solution of (2.1) are known, then the continuity requirement need only be satisfied for an appropriate (smaller) set of y-values. In adition, we assume that f satisfies a Lipschitz condition with respect to y with a uniform constant for all y. These assumptions imply, in particular, that the IVP (2.1) possesses a unique solution $y^*(t)$.

In order to simplify notation we take $a=0$, $b=1$. The interval $[0, 1]$ is discretized by subdividing it into N equal parts, where N is a positive integer. Let $h=1/N$ and let $[t_i = i \cdot h]_0^N$ denote the points of the subdivision. We now define a discretization mapping $\varphi_h : C[0,1] \to E^N$ by the relation $\varphi_h z = (z(t_1), z(t_2), \ldots, z(t_N))^T$. Now suppose Φ_h is a mapping of δ into E^N, where δ is the subspace of $C[0,1]$ consisting of all piecewise linear functions with "knots" at the points $[t_i]_1^{N-1}$. In the sequel we will often apply Φ_h to functions y in $C[0,1]$ but not in δ; this should be taken to mean the value of Φ_h when applied to that element of δ agreeing with y at $[t_i]_0^N$. Here Φ_h is to be regarded as being intimately connected with a discrete variable method, i.e. a soluction $Y \in \delta$ of $\Phi_h(Y) = 0$, if it exists, provides an approximation to $y^*(t)$ at the grid points $[t_i]_0^N$.

Combining Theorems 2.1, 2.2, and 3.3 of [9] we obtain

Theorem 2.1. Suppose

$$\Phi_h(y) - \Psi_h(y' - f) = \Psi_h \sum_{j=p}^{Q-1} c_j \frac{h^j}{j!} y^{(j+1)}(t) + O(h^Q) \tag{2.2}$$

* Present addres: Departamento de Computación, Facultad de Ciencias, Universidad Central, Caracas, Venezuela.

for $y \in C^\infty$, where p and Q are positive integers and Ψ_h is a linear mapping from C into E^N. For each y the $O(h^Q)$ is to be interpreted as holding uniformly for all $t \in [t_i]_0^N$. For $z \in C^\infty$ define

$$- F_{h,J}(z) = \Psi_h \sum_{j=p}^{J-1} c_j \frac{h^j}{j!} z^{(j)}(t), \quad p < J \leqslant Q,$$

and suppose there exists an operator $S_{h,J}: C \to E^N$ such that

$$F_{h,J}(z) = S_{h,J}(z) + \Psi_h \sum_{j=J}^{Q-1} \gamma_{j,J} z^{(j)} \frac{h^j}{j!} + O(h^Q). \tag{2.3}$$

In addition assume

$$\Phi_h(Y) = b \text{ has a solution } Y \in \delta \text{ for every } b \in E^N; \tag{2.4}$$

$$\|\varphi_h(Y - y^*)\| \leqslant c \|\Psi_h(Y) - \Phi_h(y^*)\| \text{ if } Y(0) = y^*(0), \tag{2.5}$$

is independent of h;

$$\|S_{h,J}\| \leqslant B < \infty, \quad B \text{ independent of } h; \tag{2.6}$$

$$\begin{aligned} &\Phi_h'(y^*)\, e - \Psi_h(e' - f_y(y^*) \cdot e) \\ &\quad = \Psi_h \sum_{j=p}^{Q-1} c_j \frac{h^j}{j!} e^{(j+1)}(t) + O(h^Q), \end{aligned} \tag{2.7}$$

where Φ_h' is the Fréchet derivative of Φ_h, $e \in C^\infty[0,1]$, and $f_y(y^*)$ is the Jacobian of f at y^*; and

$$\Phi_h'(y^*) \text{ is non-singular.} \tag{2.8}$$

Then if $Y^k \in \delta$ is defined by the IDC algorithm

$$\begin{aligned} &\Phi_h(Y^1) = 0, \quad Y^1(0) = y_0 \\ &\Phi_h(Y^k) = S_{h,kp}(f(Y^{k-1})) \quad 2 \leqslant k \leqslant [Q/p] \\ &Y^k(0) = y_0 \end{aligned} \tag{2.9}$$

we have

$$\varphi_h(Y^k - y^*) = O(h^{kp}) \quad 1 \leqslant k \leqslant [Q/p]. \tag{2.10}$$

The conclusion of Theorem 2.1 indicates that an asymptotic improvement of order p can be accomplished at each step of the IDC procedure. This observation has apparently been overlooked in the earlier work on IDC. As a further application of this remark one of us has performed experiments for the two-point boundary value problem using IDC on a basic method with $p = 4$ (namely the Mehrstellenverfahren of Collatz [3]). The resulting higher accuracy procedure yielded excellent results (see [10]).

The operators $S_{h,J}$ will be constructed in the following section and the mappings Ψ_h and Φ_h as well as the parameters in the expansion (2.2) for a number of explicit discrete variable schemes are identified in § 4. For specific Φ_h's, the assumptions (2.2)—(2.9) of Theorem 2.1 follow directly from the hypotheses on f, from known results about the stability of Φ_h (cf. [5]), and from the construction of the $S_{h,J}$.

3. CONSTRUCTING $S_{h,J}$ (ver galeano y Pereyra , Pub. 69-06)

As (2.3) indicates, the $F_{h,J}(z)$ which we wish to approximate by an operator $S_{h,J}$ consists of a linear combination of certain derivatives of z. Thus at each grid point t_i, $0 \leqslant i \leqslant N-1$, it is natural to consider expressions of the form

$$(3.1)\qquad [S_{h,J}(z)]_i = \sum_{j=i-\alpha_i}^{i+\beta_i} w_{j-i+\alpha_i+1}(t_i) z_j \ ,$$

where α_i and β_i are integers, $z_j = z(t_j)$, and the $[t_j]$ are grid points in $[a,b]$. It is well known (see [3], pp. 161-162) that weights $[w_\nu(t_i)]$ can be chosen so that the differential expression $F_{h,J}$ is approximated to order h^J by $S_{h,J}$ at the point t_i while the difference has an expansion

$$(3.2)\qquad [F_{h,J}(z) - S_{h,J}(z)]_i = \sum_{j=J}^{Q-1} \gamma_{i,j,J} z^j(t_i) \frac{h^j}{j!} + O(h^Q) \ .$$

To apply Theorem 2.1 we need to have *fixed* constants $\gamma_{j,J}$ independent of t_i and h such that (2.3) holds at all of the grid points $[t_j]_0^{N-1}$. One way to accomplish this is to set $\alpha_i \equiv 0$, $\beta_i \equiv J-1$, and use constant weights $[w_\nu]_1^J$ throughout. This requires a slight modification of the algorithm indicated in Theorem 2.1 inasmuch as it is necessary to integrate the differential equations a few grid points past $t = 1$.

Clearly it would be desirable to use symmetric expressions (3.1), (i.e. with $\alpha_i = \beta_i$), since this leads to the smallest truncation errors. An obvious practical idea is to use unsymmetric formulas for the few grid points near the left and right end points of the interval $[a, b]$, and a fixed symmetric formula for the bulk of the grid points in the middle. Unfortunately in this case we have not been able to prove that hypothesis (2.3) of Theorem 2.1 is satisfied or that the conclusions of the theorem remain valid. The algorithm displayed in (2.9) can still be implemented, of course, and in a later section we discuss our numerical experience with this technique.

A mixture of unsymmetric and symmetric expressions for $S_{h,J}$ at the points $[t_i]_0^{N-1}$ does fulfill hypothesis (2.3) if $S_{h,J}$ is required to approximate $F_{h,J}$ to order $O(h^Q)$ at each of the grid points- i.e. all of the coefficients $\gamma_{i,j,J}$ in (3.2) vanish for $J \leqslant j \leqslant Q-1$ and $0 \leqslant i \leqslant N-1$.

For any prescribed $F_{h,J}(z)$ and order of accuracy the weights of the corresponding approximation $S_{h,J}$ at a grid point t_i can be computed by a numerical algorithm described in detail in[1].

4. PARTICULAR Φ_h's

(I). Trapezoidal Rule: Here $p = 2$ and

$$\Phi_h(y) = h^{-1}[\varphi_h(I - T_{-h})y] - \Psi_h f(y) \ ,$$

where T_h is the shift operator defined by $T_h z(t) = z(t+h)$ and $\Psi_h = \frac{1}{2}\varphi_h(I + T_{-h})$. The coefficients c_j appearing in (2.2) are given by

$$c_2 = -\frac{1}{6}, \quad c_{2j-1} = 0, \quad c_{2j} = \frac{1}{2}\left[\frac{1-2j}{1+2j} - \sum_{i=1}^{j-1} c_{2i}\binom{2j}{2i}\right] \quad \text{for}$$

$$1 \leqslant j \leqslant [(Q-1)/2] \ .$$

(II). *Taylor* — p: The order is p and

$$\Phi_h(y) = h^{-1}\varphi_h(I-T_{-h})y - \Psi_h\left(\sum_{j=0}^{p-1} \frac{D^j f\, h^j}{j!}\right),$$

where $D^j f$ is the total derivate of $f(y(t))$ with respect to t, and $\Psi_h = \varphi_h T_{-h}$. The coefficients in (2.2) are

$$c_j = 1/j+1 \quad \text{for} \quad p \leq j \leq Q-1 \ .$$

5. PRACTICAL USE OF IDC

As mentioned earlier, to meet the assumptions concerning the approximation of $F_{h,kp}$ in Theorem 2.1, in practice one might 1) approximate $F_{h,kp}$ to order h^Q rather than just h^{kp}, or 2) use a fixed length correction term $F_{h,Q}$ instead of $F_{h,kp}$, or 3) use a single fixed (unsymmetric) numerical differentiation formula. The first two methods require a choice of Q in advance, and for all three the resulting numerical differentiation formulas have rather large constants in the truncation error and large weights in the formulas themselves. Although the above approaches appear to work well, we have preferred to use a further modification which avoids these drawbacks. We approximate $F_{h,kp}$ to order h^{kp} using different formulas at the various grid points to insure using as low order and as symmetric numerical differentiation formulas as possible although the resulting method does not fit Theorem 2.1. In order to clarify the nature of this practical procedure, we describe in more detail the method for the second order Taylor formula.

To solve $y'=f(y)$ we assume that $f_y f=g$ is known explicitly; the formula (2.2) at the solution $y=y^*$ becomes

$$\Phi_h(y)_i = \frac{y_{i+1}-y_i}{h} - f(y_i) - \frac{h}{2} g(y_i) = \sum_{j=2}^{Q-1} \frac{h^j}{(j+1)!} g^{(j-1)}(y_i) + O(h^Q) ,$$

where $y_i=y(t_i)$ and we have made use of the fact that $y''=g(y)$ is a known function. At each new correction we approximate

$$\hat{F}_{h,2k}(g(y))_i = \sum_{j=2}^{2k-1} \frac{h^j}{(j+1)!} g^{(j-1)}(y_i)$$

to order h^{2k} by approximating $g^{(2k-2)}$ to order h, $g^{(2k-3)}$ to order $h^2, \ldots,$ and $g^{(1)}$ to order h^{2k-2}, all of which require formulas based on $2k-1$ grid points. Always using "as symmetric" expressions as possible, we write

$$\hat{S}_{h,2k}(g(y))_i = \begin{cases} \sum_{j=0}^{2k-2} w_j^i \, g(y_j) & \text{for } 0 \le i \le k-1 \\ \sum_{j=i-(k-1)}^{i+(k-1)} w_j \, g(y_j) & \text{for } k \le i \le N-(k-1) \\ \sum_{j=N-(2k-2)}^{N} w_j^i \, g(y_j) & \text{for } N-(k-2) \le i \le N-1 \end{cases}$$

as an approximation to $\hat{F}_{h,2k}(g(y))_i$, where the weights w_j^i for the unsymmetric formulas and w_j for the symmetric formula can be computed very easily by an algorithm described in [1] and implemented by the Algol procedure WEIGHTGEN in [11]. The iteration (2.9) then takes the form

$$\Phi_h(Y^1) = 0,\ Y^1(0) = y_0\ ;\ \Phi_h(Y^k) = \hat{S}_{h,2k}(g(y)),\ Y^k(0) = y_0 .$$

From the above description it should be clear that it is simple to program IDC for the second order Taylor method as well as for any other discrete variable method with an expansion as in (2.2). This brings up the question of how to choose Φ_h from among the various possibilities. Clearly either "open" or "closed" Φ_h can be used, but we have found that an open formula involves many times fewer evaluations of f(y) in view of the fact that closed formulas require the solution of a system of nonlinear equations at each correction step. In practice we have computed Y^k, an order h^{kp} solution, by iteration starting with Y^{k-1}, an order $h^{(k-1)p}$ solution. It is probable that by using some other scheme for solving the nonlinear equations fewer iterations would be required. Closed

formulas such as the trapezoidal rule or the Hermite rule (see below) tend to have better stability properties and smaller asymptotic error constants than open formulas. One can use the highly accurate closed formulas of Hermite

$$\Phi_{M,h}(y)_i \equiv \frac{y_{i+1}-y_i}{h} + \sum_{j=0}^{M} d_{M,j}\left[f^{(j)}(y_i) + (-1)^j f^{(j)}(y_{i+1})\right] = \sum_{j=2M+2}^{Q-1} c_{M,j} h^j f^{(j)}(y_i) + O(h^Q) \tag{5.1}$$

which requires only M explicit derivatives to obtain an order h^{2M+2} formula as compared with the $2M+1$ derivatives needed for Taylor — $(2M+2)$. When the cost of obtaining the explicit derivatives is high, IDC provides a simple iterative method for generating highly accurate solutions; for example, a low choice of M, say $M=1$, used with IDC provides successively $h^4, h^8, \ldots$ solutions at little expense.

As was mentioned earlier, this report was stimulated by an attempt to modify the operator $\Phi_{M,h}$ (see (5.1)) used by Loscalzo and Schoenberg in such a way as to avoid explicit differentiation; this is accomplished by letting $\Phi_{0,h}(Y^0)=0$, $\Phi_{k,h}(Y^k)=0$, $k \geqslant 1$, where $\Phi_{k,h}$ is the same as $\Phi_{k,h}$ except that the derivatives $f^{(j)}(Y^k)$, $1 \leqslant j \leqslant k$ are computed numerically from the values of Y^{k-1}. It can be shown that this is equivalent to using IDC on the trapezoidal rule as described in § 4, method (I).

By now it should be apparent that we have avoided mention of multi-point methods, obvious candidates for high order methods. They do not, however, fit into the theory of § 2 (the proof breaks down since certain important relations between Y^k and y^* cannot be demonstrated via present methods), nor does it appear that the theory can be modified to allow for them. In applying IDC to the second order Adams-Bashforth formula, we have found that $Y^1=y^*+O(h^2)$, $Y^2=y^*+O(h^4)$, but $Y^{2+k}=y^*+O(h^{4+k})$ rather than $y^*+O(h^{4+2k})$ as expected; we have not even succeeded in proving that these experimental results should hold. This is disturbing since the performance of the Adams-Bashforth method with IDC is excellent, producing very accurate results with few evaluations of f despite the fact that the order of accuracy increases only one each time; this indicates that a more careful analysis of IDC applied to multi-point methods could well lead to overall schemes superior to those presented here.

6. NUMERICAL EXPERIMENTS AND COMPARISONS

In this section we discuss numerical experiments performed with some of the methods described earlier.

The following basic methods were used in conjunction with the IDC procedure:

I. Trapezoidal rule (TRAP); see § 4;

II. Taylor-p for $p=2, 3$; see § 4;

III. Adams-Bashforth of order 2 (ADB2), where

$$\Phi_{ADB2,h}(y)_i = \frac{y_{i+1}-y_i}{h} - \frac{1}{2}(3f_i - f_{i-1}) = \frac{1}{2}\sum_{j=2}^{Q-1}(2/(j+1) + (-1)^j)\frac{h^j}{j!} f_i^{(j)} + O(h^Q)$$

$$(i = 1, 2, \ldots, N-1)$$

with $y_0=y_0^*$, $y_1=y_1^*$. All of these methods were programmed in Fortran-63 for the CDC-3600 computer at the University of Wisconsin Computing Center; the experiments were carried out in double precision arithmetic ($\sim$25 decimal digits). An Algol program implementing TRAP for a system of equations was tested on the Burroughs-5500 computer at the UWCC, and can be found in the Appendix of [11].

We begin by comparing the performance of the different methods on the equation

$$y' = -y, \quad y(0)=1, \tag{6.1}$$

a standard problem for testing stability. For comparison we also include results using the rational extrapolation method of Bulirsch and Stöer [2, Table 1]. The relative error at $t=5$ are given in Table 1 together with the number of evaluations of the right hand side of the equation as well as the number of corrections used in IDC.

TABLE 1

Comparison for (6.1)

	B-S h=1/2	TRAP h=1/4	Taylor-2 h=5/16	Taylor-3 h=5/16	ADB2 h=1/4
relative error at t=5	4.0(—11)	3.4(—11)	2.0(—11)	1.6(—11)	6.0(—11)
No. of function evaluations	330	568	161(383)	80(480)	305
No. of difference corrections	—	5	7	4	7

For the Taylor formulas we have recorded in parentheses the number of function evaluations if each evaluation of f, f_x, f_y (Taylor-2) and f_{xx}, f_{yy}, f_{xy} (for Taylor-3) is counted. In many problems it is possible to evaluate all of these functions at less cost than 3 (or 6) times the number of f-evaluations.

A more significant example is provided by

$$y' = \left[\frac{\sin y}{t} + y^2 \cos y \right] / \sqrt{1+t^2} - \frac{[\arctan t]^2}{1+t^2}, \tag{6.2}$$

$$y(1) = \pi/4$$

whose exact solution is y (t) = arc tan t. In Tables 2 and 3 we display the maximum errors on the interval [1,3] for Taylor-2 and ADB2 with different step sizes. The choice of the step-sizes is due to the fact that Taylor-2 requires about twice as much computation per step as ADB2 and we wish to illustrate the results obtained with comparable amounts of work. It is clear from this and other experiments that the most economical method for a given precision is obtained when the largest step size compatible with convergence is chosen.

TABLE 2

Max. error in [1,3] for (6.2), Taylor-2

k \ h	1/10	1/40
1	7.8 (— 4)	4.9 (— 5)
2	9.7 (— 7)	1.1 (— 9)
3	6.9 (— 8)	1.8 (—11)
4	5.9 (—10)	2.9 (—14)
5	5.0 (—10)	7.9 (—16)
6	1.3 (—10)	2.4 (—18)
7	1.8 (—11)	6.1 (—19)
8	2.6 (—12)	1.2 (—20)
9	2.9 (—12)	1.1 (—21)

TABLE 3

Max. error on [1,3] for (6.2), ADB2

k \ h	1/20	1/80
1	4.6 (— 4)	3.0 (— 5)
2	4.6 (— 7)	2.1 (— 9)
3	1.2 (— 8)	1.5 (—11)
4	4.0 (—10)	1.4 (—13)
5	1.9 (—11)	2.0 (—15)
6	1.1 (—12)	3.4 (—17)
7	7.2 (—14)	7.2 (—19)
8	5.3 (—15)	1.7 (—20)
9	4.3 (—16)	4.4 (—22)

Obviously, one of the possible problems with the IDC technique is that the numerical differentiation formulas may, regardless of their high order accuracy in h, be inaccurate for "large" h; this is unfortunate since it is precisely for "large" h that one would like to use the method. Since this phenomenon has not been evident in our earlier examples, we present the results of one final experiment.

We consider the system

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}' = \begin{pmatrix} 1 - y_1 y_3 \\ y_1 - y_2 y_3 \\ y_2 - y_3^2 \end{pmatrix}, \quad y_1(0) = y_2(0) = y_3(0) = 0, \tag{6.3}$$

on $0 \leqslant t \leqslant 10$; the solution is

$$y_1(t) = \frac{\sinh(t) + \sin(t)}{\cosh(t) + \cos(t)}, \quad y_2(t) = \frac{\cosh(t) - \cos(t)}{\cosh(t) + \cos(t)}$$

$$y_3(t) = \frac{\sinh(t) - \sin(t)}{\cosh(t) + \cos(t)}$$

Experiments were performed in single precision ($\approx$12 decimals) using the trapezoidal rule (equivalently, the Hermite rule (5.1) as used by Loscalzo and Schoenberg [6] with numerical differentiation). In Table 4 we display the maximum error in $y_1(t)$, $0 \leqslant t \leqslant 10$, for $h = .4$ and $h = .2$ using k corrections; $k = 0$ corresponds to the usual trapezoidal rule which is of order h^2 while in general the order is h^{2k+2}.

TABLE 4

Max. error in [0,10] for (6.3) using TRAP

h \ k	0	1	2	3	4	5	6	7	8
.4	1.8 (—2)	2.4 (—3)	2.3 (—3)	3.2 (—3)	6.0 (—3)	1.5 (—2)	2.3 (—2)	3.1 (—2)	5.2 (—2)
.2	4.3 (—3)	1.1 (—4)	1.6 (—5)	5.0 (—6)	7.0 (—6)	1.9 (—5)	3.2 (—5)	5.5 (—5)	1.3 (—4)

We see that the behaviour of the error is not as expected. Since no such difficulty arises with $h = .1$ or when using the method of [6] with explicit differentiation, it is clear that the error in numerical differentiation destroys the accuracy of the method for $h = .4$, $k > 2$ and $h = .2$, $k > 3$; it is felt that this may be detected during computation by careful monitoring of the succesive approximate solutions Y^k, perhaps by requiring that $\|Y^k - Y^{k-1}\|$ be a decreasing sequence.

Our experiments have indicated that the IDC method described in Theorem 2.1 is a useful tool for producing accurate solutions to IVP's using relatively large step sizes; moreover, the computations described in this section indicate that the modified version of IDC discussed at the beginning of § 5, even though it is not covered by the theorem, performs at least as well as those which are covered while avoiding some of the slight disadvantages. These techniques used with finite difference operators satisfying (2.2) provide a powerful tool for computing numerical solutions of almost arbitrary accuracy with a relatively small amount of effort.

Iterated Deferred Corrections for Initial Value Problems

SUMMARY

A class of methods is presented for computing discrete solutions of succesively higher asymptotic order for initial value problems for ordinary differential equations. Numerical experiments are discussed.

Correcciones diferidas para problemas de valores iniciales

RESUMEN

Se presenta una clase de métodos para calcular soluciones discretas de órdenes asintóticos crecientes en problemas de valores iniciales para sistemas de ecuaciones diferenciales ordinarias. Se discuten experimentos numéricos.

REFERENCIES

1. Ballester, C. and Pereyra, V., "On the construction of discrete approximations to linear differential expressions", Math. Comp., *21*, 297-302 (1967).
2. Bulirsch, R. and Stoer, J., "Numerical treatment of ordinary differential equations by extrapolation methods", Numer. Math., *8*, 1-13 (1966).
3. Collatz, L., *The numerical treatment of differential equations*, 3rd edition, Springer-Verlag, Berlin (1959).
4. Fox, L., "Some improvements in the use of relaxation methods for the solution of ordinary and partial differential equations", Proc. Roy. Soc. London, A190, 31-59 (1947).
5. Henrici, P., *Discrete variable methods in ordinary differential equations*, Wiley, New York (1962).
6. Loscalzo, F. R. and Schoenberg, I. J., "On the use of spline functions for the approximation of solutions of ordinary differential equations", MRC Tech. Sum. Rpt. Nº 723 (1967).
7. Pereyra, V., "On improving an approximate solution of a functional equation by deferred corrections", Numer, Math., *8*, 376-391 (1966).
8. Pereyra, V. "Highly accurate discrete methods for nonlinear problems", MRC Tech. Sum. Rpt. Nº 749 (1967).
9. Pereyra, V. "Iterated deferred corrections for nonlinear operator equations". Numer. Math., *10*, 316-323 (1967).
10. Pereyra, V. "Iterated deferred corrections for nonlinear boundary value problems", Numer. Math. *11*, 111-125 (1968).
11. Daniel, J., V. Pereyra, and L. Schumaker, "Iterated deferred corrections for initial value problems", MRC Tech. Sum. Rep. 808 (1967).

Received: 11-III-68.
Acepted: 30-III-68.

Mejoras: (i) Paso variable

(ii) estimación asintótica de errores y su uso para detener las correcciones