Numerische Mathematik 11, 111—125 (1968)

# Iterated Deferred Corrections for Nonlinear Boundary Value Problems

Victor Pereyra*

Received June 26, 1967

## Introduction

In [11, 13, 14] we have developed the general theory of the iterated deferred corrections method (IDC), an extension of Fox's difference correction method.

In this paper we intend to clarify and illustrate some of the numerical problems that appear when this method is applied in practice. We will deal with nonlinear boundary value problems for ordinary differential equations, both with two-point and periodic boundary conditions.

We will not insist on the precise conditions under which the method is valid (besides smothness of the data, which can readily be checked), since the research on this matter is still in a lively and changing stage as it is shown by the variety of existence theorems with different sets of sufficient conditions, which in general are either too stringent or too difficult to verify (cf. [2, 7, 10], and references therein).

Let us say that generally, conditions which are sufficient for the convergence of the basic method are also sufficient to ensure the asymptotic improvement expected from the IDC method (cf. [11]).

We will then concentrate our efforts on explaining how to automate the construction of the correction terms needed in IDC, and in general on how to implement the whole procedure on a digital computer. We hope that this detailed presentation will be of help in the design of programs for other applications of the IDC procedure.

## § 1. The Continuous Problem and its Discretization

We will treat first the case of two-point boundary conditions, i.e.,

$$\begin{aligned} y'' &= f(x, y, y'), \\ y(a) = 0, &\quad y(b) = 0, \end{aligned} \tag{1.1}$$

where $f(x, y, z) \in C^\infty([a, b] \times R^2)$.

Non-homogeneous boundary conditions can be reduced to (1.1) by a simple change of variables. As it is clear from [11] the smoothness requirement can be weakened by asking $f$ to be in $C^M$ for a large $M$; this will just complicate the technical details without adding any real insight into the problem.

* Sponsored by the Mathematics Research Center, United States Army, Madison, Wisconsin, under Contract No.: DA-31-124-ARO-D-462.

Under these hypotheses, if a solution exists it must belong to $C^\infty[a, b]$. Let us then take $D$ the Banach space of twice continuously differentiable functions on $[a, b]$ which satisfy homogeneous boundary conditions and with the norm

$$|||v||| = \max\left(\max_{x\in[a,b]} |v(x)|, \frac{b-a}{4} \max_{x\in[a,b]} |v'(x)|\right).$$

The operator $F(y) \equiv y'' - f(x, y, y')$ will map $D$ into $E \equiv C[a, b]$. Let us subdivide the interval $[a, b]$ into $n$ equal parts by defining $x_i = a + ih$ ($i = 0, 1, \ldots, n$; $h = (b-a)/n$).

Let $D_h$ be the $(n-1)$-dimensional linear space of $(n+1)$-component vectors $V$ with $V_0 = V_n = 0$, and let $E_h = R_{n-1}$. In $D_h$ we consider the norm

$$|||V||| = \max\left(\max_{i=1,\ldots,n-1} |V_i|, \frac{b-a}{4} \max_{i=1,\ldots,n-1} \frac{|V_{i+1} - V_{i-1}|}{2h}\right)$$

and in $E_h$ just the maximum norm ($\|\cdot\|$).

Now we can define for every $v \in D$, $w \in E$ the discretization mappings

$$\varphi_h v = \{v(x_i)\}_{i=0,\ldots,n}, \qquad \varphi_h^0 w = \{w(x_i)\}_{i=1,\ldots,n-1}.$$

Finally we introduce the discretized version of $F(y) = 0$:

$$[\Phi_h(Y)]_i = h^{-2}(-Y_{i-1} + 2_i - Y_{i+1}) + f\left(x_i, Y_i, \frac{Y_{i+1} - Y_{i-1}}{2h}\right), \quad (i = 1, \ldots, n-1), \tag{1.2}$$

which is defined for every $Y \in D_h$.

For any $u \in C^\infty[a, b] \subset E$ we have the following asymptotic expansion (easily obtained by TAYLOR's formula)

$$\Phi_h(\varphi_h u) = \varphi_h^0 \left\{F(u) + \sum_{j=1}^{N} h^{2j} \left[\frac{-2}{(2j+2)!} u^{(2j+2)} + g_{2j}\left(u, u^{(3)}, \ldots, \frac{\partial^j f}{\partial z^j}\right)\right]\right\} + O(h^{2N+1}), \tag{1.3}$$

where the functions $g_{2j}$ can be obtained by reordering

$$\begin{aligned} \sum_{\nu=1}^{N} \frac{1}{\nu!} \frac{\partial^\nu f}{\partial z^\nu} \left[\sum_{j=1}^{N} \frac{h^{2j}}{(2j+1)!} u^{(2j+1)}\right]^\nu &= h^2 f_z \frac{u^{(3)}}{3!} + h^4 \left[f_z \frac{u^{(5)}}{5!} + f_{zz} \frac{(u^{(3)})^2}{2 \cdot 3!}\right] \\ &+ h^6 \left[f_z \frac{u^{(7)}}{7!} + f_{zz} \frac{u^{(3)} u^{(5)}}{3!\,7!} + f_{zzz} \frac{(u^{(3)})^3}{3!\,3!}\right] + h^8 \ldots. \end{aligned} \tag{1.4}$$

If $f(x, y, z)$ is linear in $z$ then (1.3) can be written explicitly as

$$\Phi_h(\varphi_h u) = \varphi_h^0 \left\{F(u) + \sum_{j=1}^{N} h^{2j} \left[\frac{-2}{(2j+2)!} u^{(2j+2)} + f_z \frac{u^{(2j+1)}}{(2j+1)}\right]\right\} + O(h^{2N+1}). \tag{1.5}$$

Under additional conditions on $f$ we can ensure the existence of a unique solution of $\Phi_h(Y) = 0$ (for sufficiently small $h$) and furthermore, if $Y(h)$ is such a solution and $y^*$ is the unique solution of (1.1) then we have discrete convergence of order $h^2$, i.e.

$$|||Y(h) - \varphi_h y^*||| = O(h^2). \tag{1.6}$$

For instance, a sufficient condition for (1.6) is that $\Phi_h$ be uniformly stable on the family of neighborhoods $\mathscr{S}(\varphi_h y^*, \varrho) \equiv \{Y \in D_h \colon |||Y - \varphi_h y^*||| \leqq \varrho\}$, where $\varrho$ does not depend on $h$; i.e.: for each $V_1, V_2 \in \mathscr{S}(\varphi_h y^*, \varrho)$ we have

$$|||V_1 - V_2||| < c\|\Phi_h(V_1) - \Phi_h(V_2)\|, \quad c > 0 \text{ independent of } h, \tag{1.7}$$

(cf. [14]).

## § 2. The IDC Procedure

Under the hypotheses of §1 it is known that the IDC procedure will give improved approximations provided that some correction terms can be constructed with sufficient accuracy ([14]).

These correction terms have to approximate the most significant parts of the local truncation error (1.3) [or (1.5)]. We will discuss in detail the casilinear case (1.5) since we feel that the general cuasilinear case (1.4) cannot be automated in a practicable manner. However we point out that whenever the terms of the expansion (1.4) can be reasonably obtained there is no intrinsic difficulty in applying the IDC method to this more general problem.

Let

$$F_k[u] = \sum_{j=1}^{k} h^{2j}\left[\frac{-2}{(2j+2)!}\, u^{(2j+2)} + f_z \frac{u^{(2j+1)}}{(2j+1)!}\right].$$

If we evaluate $F_k$ for $y^*$ instead of $u$, then we can replace, by using the differential equation, the $\nu$-th derivative of $y^*$ by the $(\nu-2)$-th total derivative of $f$ computed at $y^*, y^{*\prime}$. In this form the $k$-th segment of the local truncation error becomes

$$F_k[y^*] = \sum_{j=1}^{k} h^{2j}\left[\frac{-2}{(2j+2)!}\, f^{(2j)} + f_z \frac{f^{(2j-1)}}{(2j+1)!}\right], \tag{2.1}$$

where $f^{(\nu)} = \dfrac{d^\nu}{dx^\nu} f\big(x, y^*(x), y^{*\prime}(x)\big)$.

According to Theorem 3.3 of [14], all what is needed is a discrete approximation to $F_k[y^*]$ of order $h^{2k+2}$. We shall consider discrete approximations of the form

$$S_k(\varphi_k y^*) = \varphi_h^0 \sum_{\eta=1}^{r} a_\eta(x)\, f\big(x_\eta, y^*(x_\eta), L[\varphi_h y^*](x_\eta)\big), \tag{2.2}$$

where

$$L[\varphi_h y^*](x) = h^{-1} \sum_{\mu=1}^{s} b_\mu(x) y^*(x_\mu), \qquad x_\eta = x + \alpha_\eta h, \qquad x_\mu = x + \beta_\mu h,$$

and the $\alpha_\eta$, $\beta_\mu$ are integers.

These discrete operators must have the property

$$S_k(\varphi_h y^*) - \varphi_h^0 F_k(y^*) = O(h^{2k+2}), \tag{2.3}$$

$$L[\varphi_h y^*] - \frac{dy^*}{dx} = O(h^{2k}). \tag{2.3'}$$

First of all we point out that since $y^* \in C^\infty$, (2.3') can always be achieved by taking $s = 2k+1$ and all the $\beta_\mu$ distinct. The distribution of the abscissae $x_\mu$ around $x$ is of basic importance: more symmetric distribution implies smaller truncation error.

Once (2.3) is obtained, the method of iterated deferred corrections can be described. Taking $S_0 \equiv 0$, any solutions of the inequalities

$$\text{(IDC)} \qquad \Phi_h(U^{(k)}) - S_k(U^{(k-1)}) = O(h^{2N+1}), \qquad (k = 0, \ldots, N)$$

satisfy

$$U^{(k)} - \varphi_h y^* = \varphi_h \sum_{\nu=2k+2}^{2N} e_{\nu k} h^\nu + O(h^{2N+2}).$$

We now show how to obtain the $S_k$ with the desired property (2.3).

**Lemma 2.1.** *Let $u(t) \in C^\infty$ and let $U$ be a discrete approximation to $u(t)$ satisfying*

$$U - \varphi_h u = \varphi_h \sum_{\eta=p}^{M} e_\eta(t) h^\eta + O(h^{M+1}), \tag{2.4}$$

*where the $e_\eta(t) \in C^\infty$ do not depend on $h$.*

*If $M \geqq p + k - 1$ then any derivative of $u(t)$ of order $\leqq k$ can be approximated in terms of $U$ up to order $h^p$.*

*Proof.* Let $L_r$ be a discrete operator of the form

$$L_r[\varphi_h u](t) = \sum_{i=1}^{r+p} w_i(t)\, u(t + \alpha_i(t) h), \tag{2.5}$$

with $\alpha_i(t)$ integers. $L_r$ maps $D_h$ into $D$. There always exist weights $w_i(t)$ and integers $\alpha_i(t)$ (depending on $r$ of course) such that

$$h^{-r} L_r[\varphi_h u] - \frac{d^r u}{dt^r} = \sum_{\eta=p}^{M} g_\eta(t) h^\eta + O(h^{M+1}), \tag{2.6}$$

(cf. [1]). Moreover, the $w_i(t)$ are independent of $h$. From $L_r$ we can define a discrete operator mapping $D_h$ into $E_h$, which for simplicity we also call $L_r$, namely

$$(L_r[U])_j = \sum_{i=1}^{r+p} w_i(t_j)\, U_{i_j}, \quad \text{where} \quad t_{i_j} = t_j + \alpha_i(t_j) h.$$

It is clear that for $L_r[U]$ to be well defined it is necessary that all $t_{i_j} \in [a, b]$.

Since $L_r$ is linear we can apply it to (2.4) in order to obtain

$$h^{-r}\{L_r[U] - L_r[\varphi_h u]\} = \sum_{\eta=p}^{M} h^{-r} L_r[\varphi_h e_\eta(t)] h^\eta + O(h^{M+1-r}),$$

and from (2.6) we deduce that

$$h^{-r} L_r[U] - \varphi_h^0 \frac{d^r u}{dt^r} = \varphi_h^0 \sum_{\eta=p}^{M} \tilde{g}_\eta(t) h^\eta + O(h^{M+1-r}), \tag{2.7}$$

where $\tilde{g}_\eta(t)$ is an appropriate combination of $g_s(t)$ and $\frac{d^r e_\sigma(t)}{dt^r}$. Therefore, if $r \leqq M + 1 - p$ the Lemma follows from (2.7).

**Lemma 2.2.** *Set $g(x) = f(x, y^*(x), y^{*\prime}(x))$, and $G = \varphi_h f(x, y^*(x), L[\varphi_h y^*])$* (see (2.2) and (2.3′) for the definition of $L$). *Let $L_r$ be discrete operators as in Lemma 2.1 approximating $h^r \frac{d^r g}{dx^r}(x)$ up to the order $h^{2k+2}$ $(r = 1, \ldots, 2k)$.*

*Then*

$$S_k(\varphi_h y^*) = \sum_{j=1}^{k} \frac{-2}{(2j+2)!} L_{2j}(G) + h\,\varphi_h^0 f_z(x, y^*(x), L[\varphi_h y^*(x)]) \otimes \frac{L_{2j-1}(G)}{(2j+1)!} \quad (2.8)$$

*satisfies* (2.3) ($\otimes$ denotes componentwise multiplication of the two vector arguments).

*Proof.* According to the remark made in Lemma 2.1 with reference to formulas (2.5) and (2.6), the $L_r$ exist and they all use the same number of ordinates: $2k+2$. In addition

$$G - \varphi_h g = \varphi_h \tilde{f}_z \otimes \left(L[\varphi_h y^*] - \frac{dy^*}{dx}\right) = O(h^{2k}),$$

where $\varphi_h \tilde{f}_z$ is a vector whose components are $f_z$ computed at some intermediary points. Using Lemma 2.1 this leads to the fact that the $L_r(G)$ have the property

$$\begin{aligned} &L_1[G] - L_1[\varphi_h g] = O(h^{2k+1}),\\ &L_j[G] - L_j[\varphi_h g] = O(h^{2k+2}), \qquad (j = 2, \ldots, 2k), \end{aligned}$$

and also that

$$\begin{aligned} &\varphi_h f_z(x, y^*(x), L[\varphi_h y^*(x)]) - \varphi_h f_z\left(x, y^*(x), \frac{dy^*}{dx}(x)\right)\\ &\quad = \varphi_h \tilde{f}_{zz} \otimes \left(L[\varphi_h y^*(x)] - \varphi_h \frac{dy^*}{dx}(x)\right) = O(h^{2k}). \end{aligned}$$

From the definitions we have

$$\begin{aligned} S_k(\varphi_h y^*) - \varphi_h^0 F_k[y^*] &= \sum_{j=1}^{k} \frac{-2}{(2j+2)!} [L_{2j}(G) - h^{2j} \varphi_h^0 g^{(2j)}]\\ &+ \frac{h}{(2j-1)!} \left[\varphi_h^0 f_z(x, y^*(x), L[\varphi_h y^*(x)]) \otimes L_{2j-1}(G) - \varphi_h^0 f_z\left(x, y^*, \frac{dy^*}{dx}\right) h^{2j-1} g^{(2j-1)}\right]. \end{aligned}$$

The first term of this sum is clearly $O(h^{2k+2})$. The second term can be written as

$$\begin{aligned} &\varphi_h^0 f_z(x, y^*, L[\varphi_h y^*]) \otimes (L_{2j-1}(G) - h^{2j-1} \varphi_h^0 g^{(2j-1)}(x))\\ &\quad + \varphi_h^0 \left[f_z(x, y^*, L[\varphi_h y^*]) - f_z\left(x, y^*, \frac{dy^*}{dx}\right)\right] \otimes h^{2j-1} \varphi_h^0 g^{(2j-1)}(x)\\ &\quad = \begin{cases} O(h^{2k+1}), & j = 1,\\ O(h^{2k+2}), & j = 2, \ldots, k. \end{cases} \end{aligned}$$

Thus, since there is still a factor $h$ multiplying this last term, it follows that

$$S_k(\varphi_h y^*) - \varphi_h^0 F_k[y^*] = O(h^{2k+2})$$

as was to be shown.

In view of the smoothness of all functions involved it is clear that using Taylor's formula the terms $O(\cdot)$ can be replaced by expansions in powers of $h$ (probably containing *all* powers) up to the order $2N$ with a final term $O(h^{2N+1})$.

The numerical problem is now reduced to obtaining, for each $k$, the weights for the operators $L$ of (2.3′), and $L^\nu$ of (2.8).

## § 3. Computer Generation of Correction Formulae

The problem of obtaining the weights for formulae of the form (2.8) where the individual operators $L^{\nu}$ have the form (2.5) has been studied in [1]. An ALGOL program for the basic routine can be found in [6].

Therefore our present task is to enumerate precisely the different cases that appear according to the position of the central point $x$, and also to describe the parameters attached to each case.

As usual, $S_k(U^{(k-1)})$ means that wherever $y^*(x_i)$ appears it must be replaced by $U_i^{(k-1)}$. In this form, $S_k(U^{(k-1)})$ is a $(n-1)$-vector whose components are linear combinations of the components of $U^{(k-1)}$, with weights depending on $k$, $x_i$, and the coefficients of $L_\nu$ in (2.8). More precisely,

$$[S_k(U^{(k-1)})]_i = \sum_{j=i-t_i}^{i+s_i} w_{j-i+t_i+1} U_j^{(k-1)}, \quad (i=1, \ldots, n-1), \tag{3.1}$$

where it is assumed that the points $x_{i-t_i}, \ldots, x_{i+s_i}$, are mesh points in $[a, b]$.

As we said above the necessary precision will be achieved if the number of points $t_i+s_i+1=2k+2$. In the symmetric case $(s_i=t_i)$ we only need $2k+1$ points. For instance, the second derivative can be obtained with precision $O(h^2)$ using three points if they are symmetrically distributed, but four will be necessary if they are not.

It is clear that close to the boundary points it will be impossible to use symmetric formulae. If we let $\mathscr{L}_k$ to be the set of points $x_1, \ldots, x_k$, and $\mathscr{R}_k$ the set consisting of the points $x_{n-k+1}, \ldots, x_{n-1}$, then these are precisely the sets of points at which unsymmetric formulae are necessary. We will insist in taking in $S_k$ (for $x_i \in \mathscr{L}_k \cup \mathscr{R}_k$) all points up to the closest boundary. Thus, each of the formulae for points in $\mathscr{L}_k$ will use as base the points $x_0, \ldots, x_{2k+1}$, while similarly for $\mathscr{R}_k$ we will use the points $x_{n-2k-1}, \ldots, x_n$.

The ALGOL program we mentioned above will produce the weights $w_s(x_i)$ in (3.1) provided we give the following information:

(a) $N=s_i+t_i+1$: the number of points;

(b) $NP=s_i+1$: (integer) relative position of $x_i$ with respect to the leftmost base point;

(c) $B$: a $N$-dimensional array containing (in this case):

$$B_1=0,$$
$$B_{2j}=h f_z(x_i, U_i, (L[U])_i)/(2j+1)\cdot 2j,$$
$$B_{2j+1}=-1/j\cdot(2j+1), \quad (j=1, \ldots, k),$$
$$B_{2k+2}=B_N=0.$$

Here it has been assumed that $L[U]$, the $O(h^{2k})$ approximation to $\varphi_k y^{*\prime}$, was already available. This is of course obtained by a previous application of the subroutine with:

$$\begin{aligned}
&(\mathrm{a}') && N' = s_i' + t_i' + 1;\\
&(\mathrm{b}') && NP' = s_i' + 1;\\
&(\mathrm{c}') && B_1 = 0,\\
& && B_2 = 1/h,\\
& && B_j = 0, \quad (j = 2, \ldots, N').
\end{aligned}$$

The values of $t_i'$, $s_i'$, $t_i$, $s_i$ are given in Table 1. The set $\mathscr{S}_k$ is that at which symmetric formulae can be used, i.e. $\mathscr{S}_k = \{x_{k+1}, \ldots, x_{n-k}\}$.

Table 1

| | $\mathscr{L}_k$ | $\mathscr{S}_k$ | $\mathscr{R}_k$ |
|---|---|---|---|
| $t_i'$ | $i$ | $k$ | $2k+i-n$ |
| $s_i'$ | $2k-i$ | $k$ | $n-i$ |
| $t_i$ | $i$ | $k$ | $2k+i+1-n$ |
| $s_i$ | $2k+1-i$ | $k$ | $n-i$ |

## § 4. Periodic Boundary Conditions

Let us now consider the following boundary value problem with periodic boundary conditions:

$$\begin{aligned}
&y'' = \gamma(x, y)\, y' + \delta(x, y),\\
&y(0) = y(2\omega), \qquad y'(0) = y'(2\omega),
\end{aligned} \tag{4.1}$$

for $\omega > 0$, where $\gamma$ and $\delta$ are periodic functions of period $2\omega$ in $x$, i.e.,

$$\gamma(x + 2\omega, y) = \gamma(x, y), \qquad \delta(x + 2\omega, y) = \delta(x, y).$$

We assume that (4.1) has a unique periodic solution of period $2\omega$, and we will see how the IDC method can be applied to this case. The interval $[a, b]$ is now $[0, 2\omega]$.

Everything is as before except that we have to add a new equation since the boundary values, though equal, are unknown. The equation is

$$[\Phi_h(U)]_n = h^{-2}(-U_{n-1} + 2U_n - U_1) + \gamma(x_n, U_n)\frac{U_1 - U_{n-1}}{2h} + \delta(x_n, U_n) = 0, \tag{4.2}$$

where we have used the periodicity condition:

$$PC\colon \ \textit{for any integer } p\colon \ U_p \equiv U_{(p \bmod n)}.$$

It is this periodicity condition which enable us to use symmetric formulae everywhere in the IDC procedure; thus $\mathscr{S}_k$ is now the set of all mesh points: $\mathscr{S}_k = \{x_1, \ldots, x_n\}$.

If the functions $\gamma$ and $\delta$ satisfy the additional symmetry conditions

$$\gamma(x, y) = \gamma(-x, -y), \qquad \delta(x, y) = -\delta(-x, -y), \tag{4.3}$$

then we can reduce the amount of computation by one-half. In fact, it is clear now that the periodic solution will also satisfy the boundary conditions

$$y(0) = -y(\omega), \quad y'(0) = -y'(\omega), \tag{4.4}$$

and then we can extend it as an odd function to the whole interval $[0, 2\omega]$, and finally to the whole line by periodicity. Now Eq. (4.2) and the periodicity condition become

$$[\Phi_h(U)]_n = h^{-2}(-U_{n-1} + 2U_n + U_1) + \gamma_n \frac{(-U_1 - U_{n-1})}{2h} + \delta_n = 0, \tag{4.5}$$
$$PC': \quad U_p = -U_{(p \bmod n)}, \quad (-n < p < 2n).$$

### § 5. Computation of the Approximate Solution

The main characteristics of the IDC procedure are that it yields successively more accurate solutions $U^{(k)}$ on a fixed mesh of size $h$, and that the same type of system of nonlinear equations is to be solved at each step, i.e. the system corresponding to the simplest $O(h^2)$ method (1.2). Of course, the correction terms $S_k(U^{(k-1)})$ have to be constructed, but as we have shown in §3 this can be done in a fully automatic way.

The system (1.2) can be solved by NEWTON's method which in turn requires the solution of systems of linear equations. Such systems are easily inverted by Gaussian elimination since they are tridiagonal (cf. [8, 13]). In the periodic case (4.1) this is not quite the case and some further manipulations are needed in order to take advantage of the structure of the resulting equations. We explain this below.

The Fréchet derivative of $\Phi_h(U)$ in (1.2) applied to an element $E \in D_h$, $E = \{E_i\}$, is:

$$\begin{aligned}[\Phi_h'(U)\,E]_i &= h^{-2}(-E_{i-1} + 2E_i - E_{i+1}) + \delta(x_i, U_i)\,E_i \\ &\quad + \gamma(x_i, U_i)\,\frac{(E_{i+1} - E_{i-1})}{2h}, \quad (i = 1, \ldots, n-1).\end{aligned} \tag{5.1}$$

In the periodic case we have to add (see (4.2)):

$$\begin{aligned}[\Phi_h'(U)\,E]_n &= h^{-2}(-E_{n-1} + 2E_n - E_1) + \delta(x_n, U_n)\,E_n \\ &\quad + \gamma(x_n, U_n)\,\frac{(E_1 - E_{n-1})}{2h},\end{aligned} \tag{5.2}$$

and to use throughout the periodicity condition on $E$. If $U_0$ is a starting value, the Newton iteration for IDC ($k = 0, \ldots, N$) is then described by

$$U^{j+1} = U^j - [\Phi_h'(U^j)]^{-1}\{\Phi_h(U^j) - S_k(U^{(k-1)})\}, \tag{5.3}$$

or equivalently,

(a) solve the linear system

$$\Phi_h'(U^j)\,E^j = -\{\Phi_h(U^j) - S_k(U^{(k-1)})\}; \tag{5.4}$$

(b) put

$$U^{j+1} = U^j + E^j;$$

(c) stop when

$$\frac{\|E^j\|}{\|U^j\|} \leq \max(c \cdot h^{2N+1}, 10^{-l}); \tag{5.5}$$

where $c$ is a given (small) constant, and $l$ is the machine number length,

(d) put $U^{(k)} = U^{j+1}$ ($j \equiv$ last iterate above).

An obviously good starting vector at the $k$-th correction ($k > 0$) is $U^0 = U^{(k-1)}$.

The $n \times n$ matrix $\Phi_h'(U)$ in the periodic case has the following block structure

$$\Phi_h'(U) = \begin{pmatrix} & & & a_{1n} \\ & & & \vdots \\ & A_{11} & & 0 \\ & & & \vdots \\ & & & a_{n-1,n} \\ a_{n1} \,..\, 0 \,..\, & & a_{n,n-1} & a_{n,n} \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

where $A_{11}$ is a $(n-1) \times (n-1)$ tridiagonal matrix. We are interested in the solution of systems of the form

$$\Phi_h'(U) \begin{pmatrix} \bar{E} \\ E_n \end{pmatrix} = \begin{pmatrix} \bar{B} \\ B_n \end{pmatrix}, \tag{5.6}$$

where $\bar{E}$, $\bar{B}$ are $(n-1)$-vectors.

In this case the standard Gaussian elimination technique for tridiagonal matrices cannot be applied, but nonetheless we can reduce the inversion of the system (5.6) to that of two tridiagonal systems having $A_{11}$ as their coefficient matrices, plus a few matrix-vector multiplications and additions. If we put

$$[\Phi_h'(U)]^{-1} = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

with the same block structure as that of $\Phi_h'(U)$, then it is well known (cf. [9], p. 78) that

$$\begin{aligned} C_{22} &= \left(A_{22} - A_{21}(A_{11}^{-1}A_{12})\right)^{-1}, \\ C_{12} &= -(A_{11}^{-1}A_{12})\,C_{22}, \\ C_{21} &= -C_{22}A_{21}A_{11}^{-1}, \\ C_{11} &= A_{11}^{-1}(I - A_{12}C_{21}). \end{aligned} \tag{5.7}$$

Therefore the solution of the system of Eqs. (5.6) is given by

$$\begin{aligned} \bar{E} &= C_{11}\bar{B} + C_{12}B_n, \\ E_n &= C_{21}\bar{B} + C_{22}B_n. \end{aligned} \tag{5.8}$$

Now we show the steps in which this solution can be computed.

i) Solve $\quad A_{11}\bar{D} = A_{12}$;

ii) compute $\quad C_{22} = (A_{22} - A_{21}\bar{D})^{-1}$;

iii) solve $\quad A_{11}\bar{W} = \bar{B}$;

iv) compute $\quad v = C_{21}\bar{B} = -C_{22}A_{21}(A_{11}^{-1}\bar{B}) = -C_{22}A_{21}\bar{W}$.

Since

$$\bar{E} = A_{11}^{-1}[\bar{B} - A_{12}(C_{21}\bar{B})] - (A_{11}^{-1}A_{12})\,C_{22}B_n,$$

it follows from (i)—(iv) that,

$$\begin{aligned} \bar{E} &= \bar{W} - (v + C_{22}B_n)\bar{D}, \\ E_n &= v + C_{22}B_n. \end{aligned}$$

The stopping criterion (5.5) can be modified by requiring that the residual be only of the order $h^{2k+2}$, thus varying at each step instead of requiring the full precision all the time. This has been done in practice with excellent results, despite the fact of lacking theory on this point. In any case, this modification does not make too much difference due to the high rate of convergence of NEWTON'S method, specially with the very accurate initial guesses which are provided after the first correction. In other problems, however, it may make a considerable difference and it would be worthwhile to study the precise conditions under which it is safe to use this weaker stopping condition. It will be sufficient to show that the iteration for solving the nonlinear equations (NEWTON or any other) preserves the asymptotic expansions even when prematurely stopped.

## § 6. Numerical Examples

In what follows we present some of the numerical experiments we have performed. They have been computed in double precision arithmetic on a CDC 3600 computer (84 binary digits ~24 decimal digits) at the University of Wisconsin Computing Center.

The first equation we consider is

$$y'' - (1 - y^2) y' - 4y = -5 \sin t - \cos^3 t \tag{6.1}$$

with the boundary conditions

$$y(0) = y(2\pi), \quad y'(0) = y'(2\pi),$$

which has the periodic solution $y^*(t) = \sin t$.

We present in Table 2 the maximum absolute errors

$$\varepsilon^{(k)}(h) = \max_{i=1,\ldots,n} |U_i^{(k)}(h) - y^*(t_i)|, \quad (k = 0, \ldots, 8),$$

for

$$h_1 = 2\pi/20, \quad h_2 = 2\pi/40, \quad h_3 = 2\pi/80.$$

This is one case in which the symmetry conditions (4.3) are satisfied (as they will be in all the Lienard type equations), and we have computed only on the half period $[0, \pi]$ using the appropriate boundary and periodicity conditions. Symmetric formulae are used everywhere. As usual the notation $a \cdot b(\pm c)$ means $a \cdot b \times 10^{\pm c}$.

Table 2

| $k$ | $h$ | | |
|---|---|---|---|
| | $2\pi/20 \sim 0.32$ | $2\pi/40 \sim 0.16$ | $2\pi/80 \sim 0.08$ |
| 0 | 3.2(−3) | 8.0(−4) | 2.0(−4) |
| 1 | 5.8(−5) | 3.7(−6) | 2.3(−7) |
| 2 | 1.4(−6) | 2.2(−8) | 3.5(−10) |
| 3 | 3.5(−8) | 1.4(−10) | 5.6(−13) |
| 4 | 9.8(−10) | 1.0(−12) | 9.6(−16) |
| 5 | 4.4(−11) | 9.8(−15) | 2.4(−18) |
| 6 | 2.4(−12) | 1.3(−16) | 7.2(−21) |
| 7 | 2.4(−12) | 1.8(−18) | 2.5(−23) |
| 8 | 1.5(−13) | 4.1(−20) | 1.6(−24) |

In order to check the asymptotic behavior of the successive corrections we have computed the ratios $\varepsilon^{(k)}(h_i)/\varepsilon^{(k)}(h_{i+1})$, $(i=1, 2)$, which, according to the theory should be equal to $(h_i/h_{i+1})^{2k+2}=2^{2k+2}$.

In Table 3 we present the computed ratios together with the exact ones. For $k\leqq 6$ the behavior is very nearly as predicted while for $k>6$ it becomes slightly erratic. If we go back to Table 2 we find the reason for this deviation in the fact that we are working within the roundoff level of about $10^{-24}$.

Table 3

| $\nu$ | ratios | | |
|---|---|---|---|
| | $\varepsilon^{(\nu)}\left(\frac{2\pi}{20}\right)\Big/\varepsilon^{\nu}\left(\frac{2\pi}{40}\right)$ | $\varepsilon^{(\nu)}\left(\frac{2\pi}{40}\right)\Big/\varepsilon^{\nu}\left(\frac{2\pi}{80}\right)$ | $2^{2\nu+2}$ |
| 0 | 4 | 4 | 4 |
| 1 | 16 | 15.7 | 16 |
| 2 | 63 | 63.7 | 64 |
| 3 | 250 | 250.0 | 256 |
| 4 | 1,042 | 980.0 | 1,024 |
| 5 | 4,084 | 4,490.0 | 4,096 |
| 6 | 18,000 | 18,500.0 | 16,384 |
| 7 | 72,000 | 133,000.0 | 65,536 |
| 8 | 25,625 | 3,700,000.0 | 262,144 |

The simplest fourth order method (say COLLATZ's Mehrstellenverfahren [5], p. 164, cf. (7.2) of this paper) will behave similarly to our method for $k=1$, and from Table 1 we deduce that in order to achieve an accuracy of about $10^{-22}$ it would require a 16,000 point mesh! We have achieved this accuracy with 40 mesh points and 7 corrections. We have sistematically used as starting values (for computing $U^{(0)}$) a linear combination of the boundary values. On the average it takes about 3 Newton steps in order to obtain $U^{(0)}$ to the desired accuracy. After this the deferred corrected values $U^{(k)}$ are generally obtained with only one Newton iteration since we use very accurate starting values (i.e. $U^{(k-1)}$). Therefore if we count the number of right hand side evaluations and that of the evaluation of the partial derivatives $f_y$, $f_{y'}$, then the fourth order method of COLLATZ will take (assuming that 5 Newton corrections are necessary) $15\times 16{,}000 = 240{,}000$ function evaluations in order to reduce the discretization error below $10^{-22}$. With our procedure we needed $30\times 40=1{,}200$ function evaluations in order to achieve the same precision. Of course we have to add the work of producing the correction terms and that of solving the systems of linear equations. In this last respect we have to compare the number of operations needed to solve three $16{,}000\times 16{,}000$ tridiagonal systems against those necessary to solve ten $40\times 40$ tridiagonal systems. Since the inversion of a $n\times n$ tridiagonal system takes $6n$ divisions and multiplications, and $3n$ additions then we see that in the first case ($n=16{,}000$) 432,000 operations will be necessary while in the second case only 3600 are required.

It will perhaps be fairer to compare IDC with the rational successive extrapolation method of BULIRSCH and STOER [3], or with the Hermite-Ritz type methods of CIARLET, SCHULTZ, and VARGA [4], but unfortunately there is not enough numerical information available at the present time to make such comparisons possible.

Next we consider the following van der Pol equation with a harmonic forcing term

$$y'' - \frac{1}{9}(1-y^2)\,y' + \frac{100}{81}\,y = \frac{10}{27}\sin t, \tag{6.2}$$

and with periodic boundary conditions in the interval $[0, 2\pi]$.

URABE and REITER [15] give an approximate solution to (6.2) in the form of a trigonometrical polynomial with ten terms, which they can prove to be accurate up to $195 \times 10^{-9}$.

We have applied the IDC method with $h=\pi/40$, $N=9$, and the discrete solution thus obtained is shown in Table 4, together with that of URABE and REITER. They coincide up to nine significant figures. We point out that our values coincide with those of the 8-th correction up to 17 significant figures.

If $U^{(k)}(h)$ and $U^{(k)}(h/2)$ are the discrete solutions corresponding to the steps sizes $h$, $h/2$ (at the points of the coarsest mesh), then an estimate of the error is given by (cf. [13], p. 51)

$$\|U^{(k)}(h/2) - \varphi_h y^*\| \sim \frac{\|U^{(k)}(h) - U^{(k)}(h/2)\|}{1-(\frac{1}{2})^{2k+2}}. \tag{6.3}$$

When used in this case with $h=\pi/20$ we obtained for $k=9$

$$\|U^{(9)}(\pi/20) - \varphi_h y^*\| \sim 2.3 \times 10^{-18}$$

which tends to confirm our contention that $U^{(9)}$ is accurate to at least 17 significant figures. The computing time for obtaining both $U^{(9)}(h)$ and $U^{(9)}(h/2)$ was 15 seconds.

Other numerical examples on one- and two-dimensional problems can be found in [11—13].

## § 7. Gaining $h^4$ per Correction

In [6] it has been observed that if the basic discrete method $\Phi_h$ has order $p$ (stable, etc.), then $p$ orders in $h$ can be gained at every deferred correction by taking appropriate segments in the expansion of the local discretization error. In this Section we will explain how this can be implemented in the present situation. For simplicity we only consider problems of class $M$ (see [8], Chapter 7) with unique periodic solutions:

$$\begin{aligned} y'' &= f(x, y), \\ y(0) &= y(2\omega) = 0, \end{aligned} \tag{7.1}$$

$\omega > 0$, $f(x+2\omega, y) = f(x, y)$, $f_y > 0$.

For the discretization we use the fourth order method discussed in [8]:

$$\begin{aligned} &[\Phi_h(U)]_i = h^{-2}(-U_{i-1} + 2U_i - U_{i+1}) \\ &\quad + \tfrac{1}{12}[f(x_{i-1}, U_{i-1}) + 10 f(x_i, U_i) + f(x_{i+1}, U_{i+1})] = 0, \quad (i = 1, \ldots, n-1), \end{aligned} \tag{7.2}$$

which satisfies, for any smooth solution of (7.1),

$$\Phi_h(\varphi_h u) = \varphi_h^0 \left\{ F(u) + \sum_{j=2}^{N} \left[ \frac{1}{(j+1)(2j+1)} - \frac{1}{6} \right] f^{(2j)}(x, u(x)) \frac{h^{2j}}{(2j)!} \right\} + O(h^{2N+1}). \tag{7.3}$$

Table 4

| $t_i$ | $U_i^{(9)}$ | Urabe-Reiter |
|---|---|---|
| 7.854−2 | 3.9624226006960437353(−1) | 3.96242260445(−1) |
| 1.571−1 | 5.1534099533631665963(−1) | 5.15340996134(−1) |
| 2.356−1 | 6.3163730874287509246(−1) | 6.31627309558(−1) |
| 3.142−1 | 7.4423854998715199410(−1) | 7.44238550410(−1) |
| 3.927−1 | 8.5232011535289333850(−1) | 8.52320115183(−1) |
| 4.712−1 | 9.5504030473230497847(−1) | 9.55040304063(−1) |
| 5.498−1 | 1.0516056160497768358(+0) | 1.05160561535(+0) |
| 6.283−1 | 1.1412756538519426575(+0) | 1.14127565367(+0) |
| 7.069−1 | 1.2233768124349275611(+0) | 1.22337681302(+0) |
| 7.854−1 | 1.2973139589330727289(+0) | 1.29731396009(+0) |
| 8.639−1 | 1.3625794840360011193(+0) | 1.36257948519(+0) |
| 9.425−1 | 1.4187592914791505432(+0) | 1.41875929216(+0) |
| 1.021+0 | 1.4655355371545680504(+0) | 1.46553553728(+0) |
| 1.100+0 | 1.5026861750989436638(+0) | 1.50268617470(+0) |
| 1.257+0 | 1.5476787916609334819(+0) | 1.54767879162(+0) |
| 1.335+0 | 1.5555137095835258572(+0) | 1.55551370998(+0) |
| 1.414+0 | 1.5536922971048942046(+0) | 1.55369229763(+0) |
| 1.492+0 | 1.5423810084053333689(+0) | 1.54238100891(+0) |
| 1.571+0 | 1.5217972758382629244(+0) | 1.52179727598(+0) |
| 1.649+0 | 1.4922004734305528415(+0) | 1.49220047335(+0) |
| 1.728+0 | 1.4538837660937926858(+0) | 1.45388376596(+0) |
| 1.806+0 | 1.4071671341186187147(+0) | 1.40716713422(+0) |
| 1.885+0 | 1.3523917595642784701(+0) | 1.35239175986(+0) |
| 1.963+0 | 1.2899158696896060739(+0) | 1.28991586988(+0) |
| 2.042+0 | 1.2201120556730175483(+0) | 1.22011205562(+0) |
| 2.121+0 | 1.1433660227549478985(+0) | 1.14336602229(+0) |
| 2.199+0 | 1.0600766787699503660(+0) | 1.06007667796(+0) |
| 2.278+0 | 9.7065742875395943760(−1) | 9.70657427897(−1) |
| 2.356+0 | 8.7553851037106081987(−1) | 8.75538509848(−1) |
| 2.435+0 | 7.7517017490694281833(−1) | 7.75170174791(−1) |
| 2.513+0 | 6.7002648878560959839(−1) | 6.70026488951(−1) |
| 2.592+0 | 5.6060949932125796088(−1) | 5.60609499487(−1) |
| 2.670+0 | 4.4745347547697359724(−1) | 4.47453475469(−1) |
| 2.749+0 | 3.3112890121031270064(−1) | 3.31128901088(−1) |
| 2.827+0 | 2.1224586883197804486(−1) | 2.12245868751(−1) |
| 2.906+0 | 9.1456497808009063465(−2) | 9.14564980230(−2) |
| 2.985+0 | −3.0544002728320957938(−2) | −3.05440022857(−2) |
| 3.063+0 | −1.5301799423412943239(−1) | −1.53017993710(−1) |
| 3.142+0 | −2.7518811315509881206(−1) | −2.75188112987(−1) |

If we let $F_s(u)$ be as before

$$F_s(u) = \sum_{j=2}^{s+1} \left[\frac{1}{(j+1)(2j+1)} - \frac{1}{6}\right] \frac{f^{(2j)}}{(2j)!} h^{2j}, \tag{7.4}$$

and obtain $S_k(U)$, $(k=1, \ldots,)$, such that

$$S_k(\varphi_h y^*) - \varphi_h^0 F_{2k}(y^*) = O(h^{4k}), \tag{7.5}$$

then the IDC procedure will produce approximate solutions $U^{(k)}$ with the property

$$U^{(k)} - \varphi_h y^* = O(h^{4k+4}). \tag{7.6}$$

This method was implemented on the CDC 3600 computer. Double precision was employed throughout, and use of the periodicity was made as indicated in §4. Numerical results for

$$\begin{aligned} y'' &= y^3 - \sin x \cdot (1 + \sin^2 x), \\ y(0) &= y(\pi) = 0, \end{aligned} \tag{7.7}$$

which has the periodic solution $y(x) = \sin x$, are presented in Table 5. There we have combined the maximum absolute errors $\varepsilon^{(k)}(h_i)$, for $h_i = 2^{-i} \times \pi/10$, $i = 0, 1, 2, 3$, with the ratios (when meaningful) $r_i^k = \varepsilon^{(k)}(h_i)/\varepsilon^{(k)}(h_{i+1})$. In this case the exact ratios should be equal to $r^k = 2^{4k+4}$. The agreement is quite remarkable. The total computation time for the four step sizes and some additional computation made to check the asymptotic estimate (6.3) was 33 seconds.

Table 5

| $k$ | $\varepsilon_1$ | $r_1$ | $\varepsilon_2$ | $r_2$ | $\varepsilon_3$ | $r_3$ | $\varepsilon_4$ | $r$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 1.2(−5) | 16.2 | 7.4(−7) | 16 | 4.6(−8) | 16 | 2.9(−9) | 16 |
| 1 | 4.2(−9) | 262 | 1.6(−11) | 258 | 6.2(−14) | 258 | 2.4(−16) | 256 |
| 2 | 2.2(−12) | 4,400 | 5.0(−16) | 4,167 | 1.2(−19) | 3,429 | 3.5(−23) | 4,096 |
| 3 | 3.2(−15) | 50,000 | 6.5(−20) | 26,000 | 2.5(−24) | — | — | 65,536 |
| 4 | 1.8(−17) | 1,200,000 | 1.5(−23) | — | — | — | — | 1,048,576 |

## References

1. Ballester, C., and V. Pereyra: On the construction of discrete approximations to linear differential expressions. Math. Comp. **21**, 297—302 (1966).
2. Bayley, P., and P. Waltman: Existence and uniqueness of solutions of the first boundary value problem for non-linear second order differential equations. Arch. Rational Mech. Anal. **21**, 310—320 (1966).
3. Bulirsch, R., and J. Stoer: Fehlerabschätzungen und Extrapolation mit rationalen Funktionen bei Verfahren vom Richardson-Typus. Numer. Math. **6**, 413—427 (1964).
4. Ciarlet, P., M. Schultz, and R. Varga: Numerical methods of higher accuracy for nonlinear boundary value problems I. Numer. Math. **9**, 394—431 (1967).
5. Collatz, L.: The numerical treatment of differential equations. Third Ed. Berlin-Göttingen-Heidelberg: Springer 1959.
6. Daniel, J., V. Pereyra, and L. Schumaker: Iterated deferred corrections for initial value problems. MRC Tech. Report 808, University of Wisconsin, Madison (1967).
7. Hartman, P.: Ordinary differential equations. New York: Wiley 1964.

8. HENRICI, P.: Discrete variable methods in ordinary differential equations. New York: Wiley 1962.
9. HOUSEHOLDER, A.: Principles of numerical analysis. New York: McGraw-Hill 1953.
10. LEES, M., and M. SCHULTZ: A Leray-Schauder principle for $A$-compact mappings and the numerical solution of non-linear two-point boundary value problems. In: Numerical solution of nonlinear differential equations (ed. by D. GREENSPAN) 167—180. New York: Wiley 1966.
11. PEREYRA, V.: On improving an approximate solution of a functional equation by deferred corrections. Numer. Math. **8**, 376—391 (1966).
12. — Accelerating the convergence of discretization algorithms. MRC Tech. Rep. 687, University of Wisconsin, Madison (1966). To appear in SIAM J. Numer. Anal.
13. — Highly accurate discrete methods for nonlinear problems. Ph. D. Thesis, University of Wisconsin, Madison. Also MRC Tech. Rep. 749 (1967).
14. — Iterated deferred corrections for nonlinear operator equations. MRC Tech. Rep. 763, University of Wisconsin, Madison (1967). Numer. Math. **10**, 316—323 (1967).
15. URABE, M., and A. REITER: Numerical computation of nonlinear forced oscillations by Galerkin's procedure. J. Math. Anal. Appl. **14**, 107—140 (1966).

Dr. VICTOR PEREYRA  
Departamento de Computación  
Facultad de Ciencias  
Universidad Central  
Caracas, Venezuela 105