NUMERICAL ANALYSIS AND MATHEMATICAL METHODS
BANACH CENTER PUBLICATIONS, VOLUME 24
PWN-POLISH SCIENTIFIC PUBLISHERS
WARSAW 1990

OPTIMAL SOLUTION OF SOME TWO-POINT BOUNDARY VALUE PROBLEM

BOLESŁAW Z. KACEWICZ

Institute of Informatics, University of Warsaw, Warsaw, Poland

We consider a numerical solution of a two-point boundary value problem $u''(x) = f(x)u(x)$, $x \in [0, T]$, $u(0) = c$, $u'(T) = 0$, where c, $T \in \mathfrak{R}$ $(T > 0)$ and f is a nonnegative function with r continuous derivatives. We prove a sharp lower bound on the error of *any* algorithm that uses n values of f or its derivatives to approximate u. We show the algorithm ϕ^s with best convergence properties, based on a spline approximation of the function f. The algorithm uses n values of f, and has the error $o(n^{-r})$, as $n \to +\infty$. A multiple shooting method is used to implement the algorithm ϕ^s. The number of arithmetic operations needed to compute approximations to the solution u at n points is proportional to n, i.e., ϕ^s has almost minimal cost. We report some numerical experiments which confirm theoretical properties of ϕ^s.

Introduction

In this paper we deal with a numerical solution of the following problem. Let $f \in C^r([0, T])$ $(r \geqslant 0)$ be a nonnegative function, c, T be real numbers and $T > 0$. We wish to find a function $u = u_f$ such that

$$(1.1) \qquad u''(x) = f(x)u(x), \quad x \in [0, T], \quad u(0) = c, \; u'(T) = 0.$$

There exists a unique solution u_f to this problem. Equivalently, the function u_f is a unique solution of some optimal control problem (see (2.1)).

There are many well-known numerical methods for solving (1.1), so that from a practical point of view the problem (1.1) causes no difficulties. However, some theoretical questions remain open, even for this simple problem. Any method for solving (1.1) is based on some *information* about f, which usually consists of the values of f or its derivatives. For example, this kind of information is used in the finite element method in which integrals of f appearing in the formulation are replaced by quadrature formulas.

In this paper, we ask what is the best use that can be done of this information. More precisely, we are concerned with the following question:

What is the maximal speed of convergence of an algorithm that uses n values of f or its derivatives, as $n \to +\infty$?

By the error of an algorithm we mean the maximum norm error in $[0, T]$, and we study its asymptotic behavior at each f. In a different setting, a similar problem was recently considered in [7], where optimal properties of the finite element method were shown.

Before giving the answer to our question, we study a spline algorithm ϕ^s, which will be shown to have the best convergence properties. In this algorithm, we first replace f by a spline g^s which interpolates f at n points from $[0, T]$. Next, we approximate the solution u^s of the problem

$$u''(x) = g^s(x)\,u(x), \quad x \in [0, T],\ u(0) = c,\ u'(T) = 0. \tag{1.2}$$

The idea of substituting an original problem by a simpler one is commonly used in numerical analysis, for instance, for an integration problem (interpolation quadratures), or for the problem of solving nonlinear equations (Newton's method). Expected advantages in our case are due to the fact that g^s has a simple piecewise polynomial form, in contrast with a possibly very complex form of the function f. Therefore, we can easily compute information about g^s (e.g. derivatives or integrals of g^s) which may be very expensive or even impossible to compute for the function f (see Section 3.1).

The spline g^s which substitutes f in (1.2) is given in Section 3.2 using the values of f at n (possibly equidistant) points from $[0, T]$. We show that ϕ^s is well defined, and that its error is $o(n^{-r})$, as $n \to +\infty$, for any nonnegative function $f \in C^r([0, T])$.

The algorithm ϕ^s is not meaningful in practice unless we show a way of computing the function u^s from (1.2). Following the suggestions of [6] for one-dimensional problems, we apply to (1.2) the multiple shooting method. To obtain $o(n^{-r})$ approximations to u_f at m points from $[0, T]$, $m = \Theta(n)$, we need to solve a linear system of $2m$ equations. The matrix of this system is five diagonal (the bandwidth is independent of r), and its condition number is $O(m)$. It is to note that in difference or finite element methods the bandwidth of a matrix depends on r, and the condition number is usually $O(m^2)$. We compute this matrix with the cost proportional to $r^2 m$ arithmetic operations.

To solve the linear system, we use a stable Gaussian elimination, which requires $8m-4$ arithmetic operations (independently of r). The total cost of the algorithm ϕ^s is therefore proportional to n, i.e., proportional to the number of evaluations of f which are used as initial data. Thus, ϕ^s has almost minimal cost.

In Section 3.4 we describe numerical tests which confirm theoretical properties of the algorithm ϕ^s.

Finally, we address the main question: is it possible to define an algorithm with convergence faster than that of ϕ^s? We show that the answer is negative. It is proven in Section 4 that the convergence of *any* algorithm based on n values of f or its derivatives cannot be faster than $\delta_n n^{-r}$, where the sequence $\{\delta_n\}$ may tend to zero arbitrarily slowly. Furthermore, the set of functions for which the convergence is no faster than $\delta_n n^{-r}$ is dense. This means that the algorithm ϕ^s enjoys best convergence properties.

2. Formulation of the problem

Let f: $[0, T] \to \mathfrak{R}$ be a continuous nonnegative function and $D = \{u\colon [0, T] \to \mathfrak{R}\colon u(0) = c,\ u \in C^2([0, T])\}$, where $T > 0$ and $c \neq 0$. The problem considered in this paper arises, for instance, when dealing with the following optimal control problem:

Find $u = u_f \in D$ such that

$$\int_0^T [(u'(x))^2 + f(x)(u(x))^2]\, dx \to \min \qquad (\text{in } D). \tag{2.1}$$

It is known that the unique solution $u = u_f$ of (2.1) is also the unique solution of our boundary value problem

$$\begin{cases} u''(x) = f(x)u(x), & x \in [0, T], \\ u(0) = c, & u'(T) = 0 \end{cases} \tag{2.2}$$

(see e.g. [6], p. 532).

We wish to approximate u_f from (2.2) for functions f from the class

$$F_r = \{f\colon f \in C^r([0, T]),\ f(x) \geqslant 0\}, \qquad r \geqslant 0.$$

We assume that information about f is provided by n values of f or its derivatives at certain points from $[0, T]$. More specifically, by *information* we mean a sequence $N = \{N_n\}_{n=1}^{\infty}$, where the mapping N_n: $C^r([0, T]) \to \mathfrak{R}^n$ is defined by

$$N_n(f) = [f^{(i_1^n)}(t_1^n), \ldots, f^{(i_n^n)}(t_n^n)]^T, \tag{2.3}$$

for points $t_i^n \in [0, T]$ and $0 \leqslant i_1^n, \ldots, i_n^n \leqslant r$. The numbers i_1^n, t_1^n are given, and i_j^n and t_j^n may be chosen as functions of the previously computed values $f^{(i_1^n)}(t_1^n), \ldots, f^{(i_{j-1}^n)}(t_{j-1}^n)$ $(j = 2, 3, \ldots, n)$, so that information $N = \{N_n\}_{n=1}^{\infty}$ may be *adaptive*. Various choices of i_j^n, t_j^n $(j = 1, 2, \ldots, n)$ define different information sequences N.

By an *algorithm* ϕ using information N we mean a sequence $\phi = \{\phi_n\}_{n=1}^{\infty}$, where

$$\phi_n\colon N_n(F_r) \to C([0, T]) \tag{2.4}$$

is an arbitrary mapping. This means that the algorithm ϕ gives a sequence $\{\phi_n(N_n(f))\}_{n=1}^{\infty}$ of continuous functions which approximate the solution u_f. The nth approximation $\phi_n(N_n(f))$ is based on the values (2.3). We stress that we do not impose restrictions on the mappings ϕ_n, so that a large class of algorithms is considered.

The *error* of an algorithm ϕ at f is defined as

$$e_n(\phi, N, f) = \|u_f - \phi_n(N_n(f))\|_{\infty}, \quad f \in F_r, \tag{2.5}$$

where $\|u\|_{\infty} = \sup_{x \in [0,T]} |u(x)|$ for $u \in C([0, T])$. We are interested in algorithms with minimal errors and minimal cost. They are studied in subsequent sections.

3. Algorithm ϕ^s

In this section we define an algorithm ϕ^s with the error of order $o(n^{-r})$, as $n \to +\infty$, which requires $O(n)$ arithmetic operations to compute approximations to u_f at n points from $[0, T]$.

3.1. Preliminary remarks. Denote the problem (2.2) by $P(f)$. We apply the following idea for approximating the solution of $P(f)$:

(S.1) replace f in (2.2) by a continuous function g of a simple form, using available information on f,

(S.2) approximate a solution u_g of $P(g)$.

That is, we replace the original problem $P(f)$ by its approximation $P(g)$. We shall see that for our problem, a suitable choice of g will lead to the minimum error approximation to u_f (see Sections 3.2 and 4). Possible advantages of the procedure (S.1), (S.2) are the following:

(a) g can be chosen in a convenient simple form. It is then possible to compute easily information for g which is not available (or is expensive) f (see Sections 3.2, 3.3). It is therefore more convenient to deal with a problem $P(g)$ instead of $P(f)$.

(b) Information about f is used only in the first step (S.1). The step (S.2) does not involve extra evaluations of f, no matter what method is used to solve $P(g)$.

Standard considerations show the following properties of the perturbed problem:

LEMMA 3.1. *Let $f \in C([0, T])$ be a nonnegative function. There exists a positive number $\delta = \delta(f)$ such that for any $g \in C([0, T])$ with $\|g-f\|_{\infty} \leq \delta$ we have:*

(i) *$P(g)$ has a unique solution u_g;*

(ii) *$|u_g(x) - u_f(x)| \leq D_1 \|g-f\|_{\infty}$, $x \in [0, T]$, where D_1 is a constant which depends on $\|f\|_{\infty}$, c and T.* ∎

It follows from (i) that the procedure (S.1), (S.2) is well defined for $f \in F_r$, provided that $\|g-f\|_\infty$ is sufficiently small. In the next section, we use (ii) to define an algorithm with the error $o(n^{-r})$, based on a spline approximation of f.

3.2. Spline algorithm ϕ^s. To define the algorithm ϕ^s, we first approximate the function f by an interpolating spline g^s. Let the points $0 = x_0 < x_1 < \ldots < x_m = T$ $(m \geq 1)$ define a partition of $[0, T]$ and let $h_m = \max(|x_{i+1} - x_i|\colon\ i = 0, 1, \ldots, m-1)$. We assume that $h_m = O(m^{-1})$, as $m \to +\infty$. Let $r \geq 0$ and $f \in C^r([0, T])$. In each subinterval $[x_i, x_{i+1}]$ choose $k = \max(r, 1)+1$ distinct points z_j^i, $x_i = z_1^i < z_2^i < \ldots < z_{k-1}^i < z_k^i = x_{i+1}$ $(i = 0, 1, \ldots, m-1)$. We define g^s in $[x_i, x_{i+1}]$ as the interpolating polynomial of degree $\leq k-1$ satisfying

$$g^s(z_j^i) = f(z_j^i), \qquad j = 1, 2, \ldots, k.$$

The spline g^s is then continuous in $[0, T]$. Using twice the error formula for Lagrange interpolation, one can show that

$$|g^s(x) - f(x)| \leq \varepsilon(f, r, m) m^{-r}, \qquad x \in [0, T], \tag{3.1}$$

where $\varepsilon(f, r, m) \to 0$ as $m \to +\infty$.

To construct g^s we need information about f given by $n = m(k-1)+1$ evaluations of f

$$N_n^s(f) \tag{3.2}$$
$$= [f(z_1^0), f(z_2^0), \ldots, f(z_{k-1}^0), \ldots, f(z_1^{m-1}), f(z_2^{m-1}), \ldots, f(z_{k-1}^{m-1}), f(T)]^T.$$

To complete the definition of N_n^s for all n, we set $N_1^s = N_2^s = \ldots = N_{k-1}^s = 0$ and $N_{n+1}^s = N_{n+2}^s = \ldots = N_{n+k-2}^s = N_n^s$ for $n = m(k-1)+1$, $m = 1, 2, \ldots$.

Let $f \in F_r$. The algorithm $\phi^s = \{\phi_n^s\}_{n=1}^\infty$ that uses information $N^s = \{N_n^s\}_{n=1}^\infty$ is defined for $n = m(k-1)+1$ $(m = 1, 2, \ldots)$ by

$$\phi_n^s(N_n^s(f)) = u^s, \tag{3.3}$$

where u^s is a solution of the problem

$$u''(x) = g^s(x) u(x), \qquad u(0) = c,\ u'(T) = 0. \tag{3.4}$$

The definition of ϕ_n^s is completed for all n by setting $\phi_1^s = \phi_2^s = \ldots = \phi_{k-1}^s = 0$ and $\phi_{n+1}^s = \phi_{n+2}^s = \ldots = \phi_{n+k-2}^s = \phi_n^s$ for $n = m(k-1)+1$ $(m = 1, 2, \ldots)$. Due to Lemma 3.1, the transformation ϕ_n^s is well defined for sufficiently large n, and we have the following error bound

THEOREM 3.1. *For any* $r \geq 0$ *and* $f \in F_r$,

$$e_n(\phi^s, N^s, f) = o(n^{-r}), \qquad n \to +\infty.$$

Proof. Since $m = \theta(n)$, the desired relation follows from (ii) of Lemma 3.1 and (3.1). ∎

Remark. Information N^s is *nonadaptive* in the sense that the points z_j^i in (3.2) can be chosen independently of the function f. For instance, they can be taken as equidistant points from $[0, T]$.

Though the algorithm ϕ^s has the error of order $o(n^{-r})$, its practical value is small unless we show a way of computing u^s from (3.4). In the next section we study the implementation of ϕ^s, taking advantage of a piecewise polynomial form of g^s.

3.3. Implementation of ϕ^s. Since the function g^s is a piecewise polynomial of low degree, we can easily compute various types of information on g^s (e.g. the values of g^s, its derivatives or integrals). Thus, any commonly used method can be applied to (3.4). We follow suggestions of [6] for one-dimensional problems, and apply the multiple shooting method.

Denote by $u_{j,i}$ the solution of the initial value problem

$$u''(x) = g^s(x)u(x), \quad u(x_i) = \delta_{j,1}, \quad u'(x_i) = \delta_{j,2}, \tag{3.5}$$

where $x \in [x_i, x_{i+1}]$, $\delta_{j,k}$ is the Kronecker delta, $j = 1, 2$ and $i = 0, 1, \ldots, m-1$. Let a vector $[s_0, s_0', s_1, s_1', \ldots, s_m, s_m']^T$ be the solution of a linear system of equations

$$\begin{cases} b_{i+1}s_i + a_{i+1}s_i' - s_{i+1}' = 0, & \\ d_{i+1}s_i + r_{i+1}s_i' - s_{i+1} = 0, & i = 0, 1, \ldots, m-1, \\ s_0 = c, \quad s_m' = 0, & \end{cases} \tag{3.6}$$

where $a_{i+1} = u_{2,i}'(x_{i+1})$, $b_{i+1} = u_{1,i}'(x_{i+1})$, $r_{i+1} = u_{2,i}(x_{i+1})$ and $d_{i+1} = u_{1,i}(x_{i+1})$. The solution u^s of (3.4) then has the form

$$u^s(x) = s_i u_{1,i}(x) + s_i' u_{2,i}(x), \quad x \in [x_i, x_{i+1}], \ i = 0, 1, \ldots, m-1. \tag{3.7}$$

Thus, $u^s(x_i) = s_i$ and $(u^s)'(x_i) = s_i'$, $i = 0, 1, \ldots, m$. See [6], pp. 483–487 and 475.

We wish to approximate the values s_i, which satisfy (see Theorem 3.1):

$$\max_{1 \le i \le m} |s_i - u_f(x_i)| = o(n^{-r}) \ (= o(m^{-r})),$$

as $m \to +\infty$.

In matrix form, (3.6) reads $(m = 3)$

$$\begin{bmatrix} a_1 & 0 & -1 & 0 & 0 & 0 \\ r_1 & -1 & 0 & 0 & 0 & 0 \\ 0 & b_2 & a_2 & 0 & -1 & 0 \\ 0 & d_2 & r_2 & -1 & 0 & 0 \\ 0 & 0 & 0 & b_3 & a_3 & 0 \\ 0 & 0 & 0 & d_3 & r_3 & -1 \end{bmatrix} \begin{bmatrix} s_0' \\ s_1 \\ s_1' \\ s_2 \\ s_2' \\ s_3 \end{bmatrix} = \begin{bmatrix} -b_1 \\ -d_1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} c. \tag{3.8}$$

The matrix A_m of this system is nonsingular for sufficiently large m, since (3.4) has a unique solution.

To compute approximately s_i, we first have to approximate the unknown numbers a_i, b_i, d_i, r_i in (3.6). We use their Taylor expansions $\hat{a}_i$, $\hat{b}_i$, $\hat{d}_i$, $\hat{r}_i$, respectively. We set

$$\hat{b}_{i+1} = \sum_{k=0}^{r+1} \frac{1}{k!} u_{1,i}^{(k+1)}(x_i)(x_{i+1}-x_i)^k,$$

$$\hat{d}_{i+1} = \sum_{k=0}^{r+1} \frac{1}{k!} u_{1,i}^{(k)}(x_i)(x_{i+1}-x_i)^k,$$

$i = 0, 1, \ldots, m-1$.

Replacing $u_{1,i}$ by $u_{2,i}$, we obtain similar formulas for $\hat{a}_{i+1}$ and $\hat{r}_{i+1}$. Using (3.5) to express the derivatives of $u_{j,i}$, we see that the approximations $\hat{a}_{i+1}$, $\hat{b}_{i+1}$, $\hat{d}_{i+1}$, $\hat{r}_{i+1}$ are based on the values $g^s(x_i)$, $(g^s)'(x_i)$, $\ldots$, $(g^s)^{(r)}(x_i)$. Here and above, a derivative at x_i is meant as a right hand derivative. This information is readily available, since g^s is a polynomial in $[x_i, x_{i+1}]$. Observe that for the original problem with f, the Taylor expansion could not be used as above, since the derivatives of f may be unknown (see (3.2)).

By definition, we have for $f \in F_r$ that

$$\text{(3.9)} \qquad \varepsilon_m = \max_{1 \leqslant i \leqslant m} \max(|a_i-\hat{a}_i|, |b_i-\hat{b}_i|, |d_i-\hat{d}_i|, |r_i-\hat{r}_i|) = \begin{cases} o(m-1) & \text{if } r = 0, \\ O(m^{-(r+2)}) & \text{if } r \geqslant 1, \end{cases}$$

as $m \to +\infty$, where the "o" and "O" expressions depend on f, r and T. Replacing a_i, b_i, d_i, r_i in (3.8) by $\hat{a}_i$, $\hat{b}_i$, $\hat{d}_i$, $\hat{r}_i$, respectively, we get a new system with a matrix $\hat{A}_m$. We now show that the new system has a unique solution $[\hat{s}'_0, \hat{s}_1, \hat{s}'_1, \ldots, \hat{s}_{m-1}, \hat{s}'_{m-1}, \hat{s}_m]^T$, and that for $f \in F_r$ we have

$$\text{(3.10)} \qquad \max_{1 \leqslant i \leqslant m} |\hat{s}_i - u_f(x_i)| = o(m^{-r}), \qquad m \to +\infty.$$

We first note that $a_i = 1+o(m^{-1})$, $d_i = 1+O(m^{-2})$, $r_i = x_i - x_{i-1} + o(m^{-2})$ and $b_i = g^s(x_{i-1})(x_i - x_{i-1}) + o(m^{-1})$, which yields, after some calculations, that $\|A_m\|_\infty = O(1)$ and $\|A_m^{-1}\|_\infty = O(m)$, as $m \to +\infty$. Since $\hat{A}_m = A_m(I + A_m^{-1}(\hat{A}_m - A_m))$ and $\|\hat{A}_m - A_m\|_\infty = o(m^{-(r+1)})$ (see (3.9)), $\hat{A}_m$ is nonsingular for sufficiently large m. Furthermore, it is known that (see [8])

$$\max_{1 \leqslant i \leqslant m} |s_i - \hat{s}_i| = O(\|A_m^{-1}\|_\infty \varepsilon_m) = \begin{cases} o(1) & \text{if } r = 0, \\ O(m^{-(r+1)}) & \text{if } r \geqslant 1, \end{cases}$$

which proves (3.10). Thus, the numbers $\hat{s}_i$ approximate $u_f(x_i)$ with the same error as s_i, $i = 1, 2, \ldots, m$.

To compute $\hat{s}_i$, we use Gaussian elimination without pivoting for solving the system with the matrix $\hat{A}_m$. For stability reasons, we first slightly modify

$\hat{A}_m$. Note that the numbers $\hat{a}_i$, $\hat{d}_i$ and $\hat{r}_i$ are positive for sufficiently large m, while $\hat{b}_i$ may assume negative values. It is possible however to show that $\max_{1\leq i\leq m} |\max(\hat{b}_i, 0)-b_i| = o(m^{-(r+1)})$, $m \to +\infty$. Hence, negative numbers $\hat{b}_i$ can be replaced by 0 without affecting (3.10). The modified matrix $\hat{A}_m$ has therefore the form (3.8) with $a_i := \hat{a}_i$, $b_i := \max(\hat{b}_i, 0)$, $d_i := \hat{d}_i$ and $r_i := \hat{r}_i$.

Eliminating superdiagonal nonzero elements we get relations for computing $\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_m$:

$$\text{(3.11)} \quad \begin{cases} p_m = 0, \quad p_{i-1} = (\max(\hat{b}_i, 0)+\hat{d}_i p_i)/(\hat{a}_i+\hat{r}_i p_i), & i = m, m-1, \ldots, 1, \\ \hat{s}_0 = c, \quad \hat{s}_i = (\hat{d}_i - \hat{r}_i p_{i-1})\hat{s}_{i-1}, & i = 1, 2, \ldots, m. \end{cases}$$

Since $\hat{a}_i+\hat{r}_i p_i > 0$ $(i = 1, 2, \ldots, m)$, this algorithm is well defined. Gaussian multipliers used in the elimination (see [8]) are equal to $1/(\hat{a}_i+\hat{r}_i p_i)$ and p_{i-1}. It is not difficult to see that they are nonnegative and bounded from above by a constant independent of m, which indicates stability of the elimination (in the sense of Wilkinson [8]).

We now consider the cost of the algorithm ϕ^s. In the first step, we compute coefficients of the linear system to be solved, that is, we evaluate $g^s(x_i)$, $(g^s)'(x_i), \ldots, (g^s)^{(r)}(x_i)$ and next compute $\hat{a}_i$, $\hat{b}_i$, $\hat{d}_i$ and $\hat{r}_i$ $(i = 1, 2, \ldots, m)$. Since g^s is a polynomial of degree $\leq \max(r, 1)$ in $[x_i, x_{i+1}]$, the number of arithmetic operations in this step is equal to $l(r)m$, where $l(r)$ depends only on r, and is proportional to r^2. In the second step, we solve the linear system of equations by means of (3.11), which requires $5m-2$ multiplications and $3m-2$ additions. Note that this part of the cost is independent of the order r.

The total cost of the algorithm ϕ^s is therefore equal to $(l(r)+8)m-4$ arithmetic operations. Since ϕ^s is based on $n = \max(r, 1)m+1$ pieces of initial data (see (3.2)), its cost depends linearly on n. This yields that the algorithm ϕ^s has almost minimal cost.

Remarks. 1. As pointed out in [5], the algorithm (3.11) is well behaved in the following sense. Denote by $\bar{s}_i$ an approximation to $\hat{s}_i$ computed by (3.11) in t digit floating point binary arithmetic. It is possible to show that the numbers $\bar{s}_i$ are exact results for a slightly perturbed data $\bar{a}_i = \hat{a}_i(1+\alpha_i)$, $\bar{b}_i = \max(\hat{b}_i, 0)$, $\bar{d}_i = \hat{d}_i(1+\beta_i)$, $\bar{r}_i = \hat{r}_i(1+\varrho_i)$, where $\max_{1\leq i\leq m}(|\alpha_i|, |\beta_i|, |\varrho_i|)$ is of order $6\cdot 2^{-t}$.

This yields that the error $\max_{1\leq i\leq m} |\bar{s}_i-\hat{s}_i|$ of the computed approximation is of order $6\cdot 2^{-t}\, \|\hat{A}_m^{-1}\|_\infty \max_{1\leq i\leq m} |\hat{s}_i|$.

2. Iterative refinement can be used to improve accuracy of the solution of (3.8) obtained in t digit floating point binary arithmetic. Recall that for the system $\hat{A}_m s = b$, one step of iterative refinement is defined as follows:

– compute an approximation s^1 to the exact solution s using some algorithm A,

– compute the residual $r = \hat{A}_m s^1 - b$,
– solve $\hat{A}_m d = r$ using the algorithm A to get an approximation d^1 to d,
– compute the new approximation $\bar{s} = s^1 - d^1$.

To guarantee that $\bar{s}$ is more accurate than s^1, usually higher precision has to be used to compute the residual r (see [8]). It has been however observed in [2] that for some linear systems derived by discretization of differential equations a sufficient accuracy of the computed residual can be obtained using only *single* precision. As pointed out in [5], the same idea can be adapted to the system (3.8). In single precision, the residual for (3.8) can be computed with an absolute (maximum norm) error of order $m^{-1}2^{-t}$. If A is Gaussian elimination algorithm (as in (3.11)), such accuracy of the computed residual guarantees that for sufficiently large m the error $\|\bar{s}-s\|_\infty$ is of order $(2^{-t}+m^2 2^{-2t})\|s\|_\infty$. Recall that for the algorithm (3.11) without refinement, the error $\|\bar{s}-s\|_\infty$ of the computed solution is of order $m2^{-t}\|s\|_\infty$.

3.4. Numerical results. We report in this section numerical results obtained by using the algorithm ϕ^s with $r = 1$ for problems $u''(x) = f(x)u(x)$, $x \in [0, 1]$, $u(0) = 1$, $u'(1) = 0$, where (accordingly):

(i)
$$f(x) = \frac{1+\sin^2 x}{\cos^2 x},$$

with the solution

$$u_f(x) = \frac{1}{\cos x} - \frac{\sin 1}{\cos^3 1 + \cos 1 + \sin 1}\left(\sin x + \frac{x}{\cos x}\right);$$

(ii)
$$f(x) = \frac{2}{(x+2)(x+1)^2},$$

with the solution

$$u_f(x) = \frac{1}{3+4\log 3 - 4\log 2}\left(2\frac{x+2}{x+1}\log\frac{3}{x+2} + x + 3\right);$$

(iii)
$$f(x) = \frac{1}{1 - xe^{1-x} + e^{-x}},$$

with the solution

$$u_f(x) = \tfrac{1}{2}(e^x - ex + 1).$$

We have used information given by evaluation of f at equidistant points from $[0, 1]$,

$$N_n^s(f) = [f(x_0), f(x_1), \ldots, f(x_m)]^T, \qquad n = m+1,$$

where $x_i = ih$ and $h = 1/m$. Since we take $r = 1$, the function g^s is linear in $[x_i, x_{i+1}]$ and we have in (3.11)

$$\hat{a}_i = \hat{d}_i = 1+\tfrac{1}{2}h^2 f(x_{i-1}),$$
$$\hat{r}_i = h,$$
$$\hat{b}_i = \tfrac{1}{2}h\big(f(x_{i-1})+f(x_i)\big),$$

for $i = 1, 2, \ldots, m$. It follows from (3.10) that for any $f \in F_1$, the approximations $\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_m$ satisfy

$$e_m = \max_{1 \leqslant i \leqslant m} |\hat{s}_i - u_f(x_i)| = o(m^{-1}), \qquad m \to +\infty.$$

Since the functions f given (i)–(iii) belong to $C^{\infty}([0, T])$, we even have $e_m = O(m^{-2})$, $m \to +\infty$. Indeed, now $\varepsilon(f, r, m) = O(m^{-1})$ in (3.1) and $\varepsilon_m = O(m^{-3})$ in (3.9), which yields the desired bound on e_m.

Calculations have been performed on an IBM PC-XT computer with relative computer precision about 10^{-7} (single precision) and 10^{-15} (double precision). Both single (sp) and double (dp) precision results have been computed for m varying from 10 to 20 000, which allows to see the theoretical behavior of the algorithm as well as its sensitivity to rounding errors. The table below contains the values $e_m m^2$ for the problems (i)–(iii).

m	(i) sp	(i) dp	(ii) sp	(ii) dp	(iii) sp	(iii) dp
10	0.47	0.47	0.11	0.11	0.11	0.11
50	0.43	0.43	0.11	0.10	0.099	0.097
100	0.42	0.42	0.11	0.10	0.097	0.096
200	0.42	0.42	0.10	0.10	0.12	0.095
300	0.43	0.42	0.11	0.10	0.13	0.094
450	0.24	0.41	0.18	0.10	0.16	0.094
500	0.79	0.41	0.51	0.10	0.58	0.094
1000	6.8	0.41	4.4	0.10	4.5	0.094
3000	270	0.41	490	0.10	360	0.094
5000	2000	0.41	590	0.10	2200	0.094
9000	4500	0.41	780	0.10	3900	0.094
10 000	2000	0.41	2200	0.10	2100	0.094
20 000	18 000	0.41	9300	0.10	11 000	0.094

The results in double precision (where rounding errors may be neglected) confirm the theoretical properties of ϕ^s. In single precision, the influence of rounding errors becomes significant for $m \geqslant 500$. Since the condition number of $\hat{A}_m$ is of order m, the perturbation due to these errors may be of order $m\,10^{-7}$. The obtained results show that for our problems the influence of rounding errors is sligtly below this level. In all examples, the minimum error in single precision is obtained for $m = 450$, and $e_{450} < 1.6 \cdot 10^{-6}$.

4. Optimality of ϕ^s

In this section we prove the main result of this paper. We show that the error of any algorithm cannot be essentially less than $\delta_n n^{-r}$, as $n \to +\infty$, where the sequence $\{\delta_n\}$ may tend to zero arbitrarily slowly. This will indicate the optimality of the algorithm ϕ^s.

We first state the theorem. Let a norm in $C^r([0, T])$ be defined by $\|f\| = \sum_{i=0}^{r} \|f^{(i)}\|_\infty$.

THEOREM 4.1. *Let N be any information given by* (2.3) *and ϕ be any algorithm using N. Then for any sequence $\{\delta_n\}$, $\delta_n \to 0^+$, the set*

$$B = B(\phi, N, \{\delta_n\}) = \left\{ f \in F_r \colon \lim_{n \to +\infty} \frac{e_n(\phi, N, f)}{\delta_n n^{-r}} = 0 \right\}$$

has empty interior in F_r, i.e., $\overline{F_r - B} = F_r$. ∎

Before presenting the proof, we comment on this theorem. Consider any algorithm $\phi = \{\phi_n\}_{n=1}^{\infty}$ (with no restriction imposed on the mappings ϕ_n) that uses evaluations of f or its derivatives. For functions f from a dense subset of F_r, the error of ϕ cannot converge to zero faster than $\delta_n n^{-r}$ $(n \to +\infty)$. More specifically, for any $f \in F_r$ there exists $g \in F_r$ with derivatives $g, g', \ldots, g^{(r)}$ arbitrarily close in $[0, T]$ to those of f, such that $\limsup_{n \to +\infty} e_n(\phi, N, g) \times \times (\delta_n n^{-r})^{-1} > 0$. This holds for any positive sequence $\{\delta_n\}$, arbitrarily slowly convergent to zero.

Theorems 3.1 and 4.1 together with considerations in Section 3.3 lead to the following conclusion.

COROLLARY 4.1. *The algorithm ϕ^s has best convergence properties and almost minimal cost in the class of all algorithms that use any information of the form* (2.3). ∎

The remaining part of this section is devoted to the proof of Theorem 4.1. Note first that $C^r([0, T])$ is a Banach space under the norm defined above, F_r is a closed subset of $C^r([0, T])$, and u depends continuously on f. For $f \in F_r$ define

$$d_\alpha(N_n, f) = \sup \{ \|u_g - u_f\|_\infty \colon g \in F_r,\ \|g - f\| \leqslant \alpha,\ N_n(g) = N_n(f) \}. \tag{4.1}$$

The number $d_1(N_n, f)$ measures the maximal distance between the solutions for f and for a function from the unit ball around f, which shares the same information with f. We call it the *local diameter of information* N_n. We now recall the theorem from [3], which we use to prove Theorem 4.1. By $N_n(f) \subset N_{n+1}(f)$ we mean below that the first n components of $N_{n+1}(f)$ are the same as the components of $N_n(f)$.

THEOREM 4.2 *Let N be any information given by* (2.3) *such that $N_n(f) \subset N_{n+1}(f)$ for all f, $n = 1, 2, \ldots$. Let ϕ be any algorithm given by* (2.4). *If*

(a) $d_1(N_n, f) < +\infty$ *for* $f \in F_r$ *and sufficiently large* n,

(b) *for any* $f \in F_r$ *there exists a positive constant* $C(f)$ *such that*

$$d_\alpha(N_n, f) \geqslant \alpha C(f) d_1(N_n, f),$$

for $\alpha \in [0, 1]$ *and sufficiently large* n,
then for any sequence $\{\delta_n\}$, $\delta_n \to 0^+$, *the set*

$$A = \left\{ f \in F_r \colon \lim_{n \to +\infty} \frac{e_n(\phi, N, f)}{\delta_n d_1(N_n, f)} = 0 \right\}$$

has empty interior in F_r, *i.e.*, $\overline{F_r - A} = F_r$. ∎

We need to show that (a) and (b) hold for our problem. To do this we need an upper bound on the distance between two solutions of (2.2).

LEMMA 4.1. *Let* f, $g \in C([0, T])$ *be nonnegative functions. Then for* $x \in [0, T]$

$$|u_g(x) - u_f(x)| \leqslant D_2(\|g\|_\infty, x) \sup_{x \in [0,T]} \left| \int_T^x \int_T^t h(\theta) u_f(\theta) \, d\theta \, dt \right|, \tag{4.2}$$

where $h = g - f$ *and for* $a \geqslant 0$

$$D_2(a, x) = \tfrac{1}{2}\left(\tfrac{1}{2}(e^{\sqrt{a}T} + e^{-\sqrt{a}T}) + 1\right)\left(e^{\sqrt{a}(T-x)} + e^{-\sqrt{a}(T-x)}\right).$$

Proof. The proof is typical for this kind of results, and we present it only for completeness. Let $y = y_g(x, s)$ be the solution of the initial value problem $y''(x) = g(x) y(x)$, $x \in [0, T]$, $y(T) = s$, $y'(T) = 0$, where $s \in \mathfrak{R}$. From Lemma A1 we have for $x \in [0, T]$

$$\begin{aligned} u_g(x) - u_f(x) = {} & \int_T^x \int_T^t g(\theta)\left(u_g(\theta) - u_f(\theta)\right) d\theta \, dt + \int_T^x \int_T^t h(\theta) u_f(\theta) \, d\theta \, dt \\ & - m_g(0)^{-1} \left(\int_T^0 \int_T^t g(\theta)\left(y_g(\theta, s_f^*) - u_f(\theta)\right) d\theta \, dt + \int_T^0 \int_T^t h(\theta) u_f(\theta) \, d\theta \, dt \right), \end{aligned} \tag{4.3}$$

where $m_g(0) = \partial y_g(0, 0)/\partial s \geqslant 1$ and $s_f^* = u_f(T)$. We need an upper bound on $|y_g(x, s_f^*) - u_f(x)|$, $x \in [0, T]$. From (A.1) of Lemma A1 and Lemma A3 we get

$$|y_g(x, s_f^*) - u_f(x)| \leqslant \tfrac{1}{2}\left(\exp\left(\sqrt{\|g\|_\infty}(T-x)\right) + \exp\left(-\sqrt{\|g\|_\infty}(T-x)\right)\right) B \tag{4.4}$$

for $x \in [0, T]$, where

$$B = \sup_{x \in [0,T]} \left| \int_T^x \int_T^t h(\theta) u_f(\theta) \, d\theta \, dt \right|.$$

Using this in (4.3), we obtain the inequality

$$\begin{aligned} |u_g(x) - u_f(x)| \leqslant {} & \|g\|_\infty \int_T^x \int_T^t |u_g(\theta) - u_f(\theta)| \, d\theta \, dt \\ & + B\left(\tfrac{1}{2}\left(\exp(\sqrt{\|g\|_\infty}\, T) + \exp(-\sqrt{\|g\|_\infty}\, T)\right) + 1\right), \quad x \in [0, T]. \end{aligned} \tag{4.5}$$

Relation (4.2) now follows from (4.5) and Lemma A3, which completes the proof. ■

We also need the following lower bound on the distance between two solutions of (2.2).

LEMMA 4.2. *Let functions $f, g \in C([0, T])$ be such that solutions u_f and u_g of* (2.2) *exist. Then*

$$\|u_g - u_f\|_\infty \geq D_3(\|g\|_\infty) \sup_{x \in [0,T]} \Big|\int_T^x \int_T^t h(\theta) u_f(\theta)\, d\theta\, dt\Big|, \tag{4.6}$$

where $h = g - f$ and $D_3(a) = \frac{1}{2}(1 + \frac{1}{2}T^2 a)^{-1}$ for $a \geq 0$.

Proof. From (2.2) we have for $x \in [0, T]$

$$u_f(x) - u_g(x) = -\int_0^x \int_T^t h(\theta) u_f(\theta)\, d\theta\, dt + \int_0^x \int_T^t g(\theta)\big(u_f(\theta) - u_g(\theta)\big)\, d\theta\, dt,$$

which gives

$$\|u_g - u_f\|_\infty \geq \big(1 + \tfrac{1}{2}x(2T - x)\|g\|_\infty\big)^{-1} \Big|\int_0^x \int_T^t h(\theta) u_f(\theta)\, d\theta\, dt\Big|. \tag{4.7}$$

It is easy to see that

$$\sup_{x \in [0,T]} \Big|\int_0^x H(t)\, dt\Big| \geq \tfrac{1}{2} \sup_{x \in [0,T]} \Big|\int_T^x H(t)\, dt\Big|,$$

for any $H \in C([0, T])$. Using this in (4.7), we get (4.6). ■

We are ready to show that Theorem 4.2 can be applied to our problem.

LEMMA 4.3. *Assumptions* (a) *and* (b) *of Theorem* 4.2 *are satisfied for the problem* (2.2) *and information* (2.3).

Proof. Inequality (a) (for $n = 1, 2, \ldots$) follows from (4.2) and Lemma A2. We show that (b) holds. Let $f, g \in F_r$, $\|g - f\| \leq 1$ and $h = g - f$. From Lemma 4.1 we get

$$\|u_g - u_f\|_\infty \leq D_2(\|f\|_\infty + 1, 0) \sup_{x \in [0,T]} \Big|\int_T^x \int_T^t h(\theta) u_f(\theta)\, d\theta\, dt\Big|. \tag{4.8}$$

Let $g_\alpha = f + \alpha h$ for $\alpha \in [0, 1]$. Then $g_\alpha \in F_r$ and $\|g_\alpha - f\| \leq \alpha$. From Lemma 4.2 with $g := g_\alpha$ we have

$$\|u_{g_\alpha} - u_f\|_\infty \geq D_3(\|f\|_\infty + 1)\alpha \sup_{x \in [0,T]} \Big|\int_T^x \int_T^t h(\theta) u_f(\theta)\, d\theta\, dt\Big|. \tag{4.9}$$

Combining (4.8) and (4.9) we obtain for $\alpha \in (0, 1]$

$$\|u_g - u_f\|_\infty \leq \frac{D_2(\|f\|_\infty + 1, 0)}{D_3(\|f\|_\infty + 1)\alpha} \| u_{g_\alpha} - u_f\|_\infty. \tag{4.10}$$

Assume now that $N_n(g) = N_n(f)$. Since N_n is a linear operator, we have $N_n(g_\alpha) = N_n(f)$. Hence, (4.10) yields

$$\|u_g - u_f\|_\infty \leqslant (C(f)\alpha)^{-1} d_\alpha(N_n, f), \tag{4.11}$$

where $C(f) = D_3(\|f\|_\infty + 1) D_2(\|f\|_\infty + 1, 0)^{-1}$. Since (4.11) holds for any $g \in F_r$ such that $\|g - f\| \leqslant 1$ and $N_n(g) = N_n(f)$, the proof of (b) is completed. ∎

We now use Theorem 4.2. It states that a lower bound on the error of an arbitrary algorithm ϕ can be obtained as a lower bound on $d_1(N_n, f)$. We now estimate $d_1(N_n, f)$ from below.

LEMMA 4.4. *Let $r \geqslant 0$ and $f \in F_r$. There exists a positive constant $D_4(f)$ such that, for any information N_n given by* (2.3),

$$d_1(N_n, f) \geqslant D_4(f) n^{-r}, \qquad n = 1, 2, \ldots \tag{4.12}$$

Proof. Choose a nonnegative function $h \in C^r([0, T])$ such that

(a) $\|h\| \leqslant 1$;

(b) $N_n(h) = 0$, where the numbers i_j^n and t_j^n $(j = 1, 2, \ldots, n)$ in (2.3) are computed for the function f;

(c) $\int_{T/2}^{T} h(t)\,dt \geqslant K n^{-r}$, $n = 1, 2, \ldots$, where K is a positive constant independent of n. See [4], Lemma 3.2, which allows to construct a function h with these properties.

Let $g = f + h$. Then $g \in F_r$, and we have by Lemma 4.2

$$\|u_g - u_f\|_\infty \geqslant D_3(\|f\|_\infty + 1) \left| \int_T^0 \int_T^t h(\theta) u_f(\theta)\, d\theta\, dt \right|. \tag{4.13}$$

Assume without loss of generality that $c > 0$ in (2.2). From Lemma A2, we have $u_f(x) \geqslant u_f(T) > 0$ for $x \in [0, T]$. Since h is nonnegative, (4.13) yields

$$\|u_g - u_f\|_\infty \geqslant D_3(\|f\|_\infty + 1) u_f(T) \int_T^0 \int_T^t h(\theta)\, d\theta\, dt. \tag{4.14}$$

Finally, note that

$$\int_T^0 \int_T^t h(\theta)\, d\theta\, dt = \int_0^T t h(t)\, dt \geqslant \tfrac{1}{2} T \int_{T/2}^{T} h(t)\, dt.$$

Using this in (4.14), we get (4.12) from conditions (a)–(c). ∎

We are ready to prove the main result of this section.

Proof of Theorem 4.1. First assume that for all f $N_n(f) \subset N_{n+1}(f)$, $n = 1, 2, \ldots$. From Lemma 4.4, we find that $B \subset A$, where the set A is defined in Theorem 4.2, and has empty interior. Hence, B has empty interior. Now assume that information N is of general form (2.3). Let ϕ be any algorithm using N and let $\{\delta_n\}$ be any positive sequence converging to zero. As in the proof of Theorem 4.2 from [4], it is possible to construct information N^* such that $N_n^*(f) \subset N_{n+1}^*(f)$ for all f $(n = 1, 2 \ldots)$, an algorithm ϕ^* using N^*, and a sequence $\{\delta_n^*\}$, $\delta_n^* \to 0^+$, for which $B(\phi, N, \{\delta_n\}) \subset B(\phi^*, N^*, \{\delta_n^*\})$. This inclusion and the first part of this proof yield that $B(\phi, N, \{\delta_n\})$ has empty interior, which completes the proof of Theorem 4.1. ∎

Appendix

For completeness, we state in this section standard results concerning (2.2) which have been used in the paper.

LEMMA A1. *Let $f, g \in C([0, T])$. Denote by $y_f = y_f(x, s)$ a solution of the problem $y''(x) = f(x)y(x)$, $y(T) = s$, $y'(T) = 0$, $x \in [0, T]$, $s \in \Re$. Then*

$$\text{(A.1)} \qquad y_g(x, s) - y_f(x, \bar{s}) = \int_T^x \int_T^t g(\theta)\big(y_g(\theta, s) - y_f(\theta, \bar{s})\big)\, d\theta\, dt + \int_T^x \int_T^t (g(\theta) - f(\theta))\, y_f(\theta, \bar{s})\, d\theta\, dt + s - \bar{s},$$

where $x \in [0, T]$.

(A.2) *There exists in $[0, T] \times \Re$ a continuous derivative $m_f(x, s) = \partial y_f(x, s)/\partial s$ satisfying $m_f''(x, s) = f(x) m_f(x, s)$, $m_f(T, s) = 1$, $m_f'(T, s) = 0$. The function m_f is independent of s, $m_f(x, s) = m_f(x)$, and $m_f(x) \geqslant 1$ $(x \in [0, T])$ for nonnegative functions f. Furthermore, $y_f(0, s) = m_f(0)s$.*

(A.3) *If the solutions u_f and u_g of (2.2) exist (i.e., $m_f(0)m_g(0) \neq 0$), then*

$$u_g(x) - u_f(x) = \int_T^x \int_T^t g(\theta)\big(u_g(\theta) - u_f(\theta)\big)\, d\theta\, dt + \int_T^x \int_T^t h(\theta)\, u_f(\theta)\, d\theta\, dt$$
$$- m_g(0)^{-1}\Big(\int_T^0 \int_T^t g(\theta)\big(y_g(\theta, s_f^*) - u_f(\theta)\big)\, d\theta\, dt + \int_T^0 \int_T^t h(\theta)\, u_f(\theta)\, d\theta\, dt\Big),$$

where $h = g - f$ and $s_f^* = u_f(T)$. ∎

LEMMA A2. *Let $f \in C([0, T])$ be a nonnegative function and let $c > 0$. Then the solution u_f of (2.2) satisfies*

$$0 < u_f(T) \leqslant u_f(x) \leqslant c, \qquad \text{for } x \in [0, T].$$

(For $c < 0$ we have $c \leqslant u_f(x) \leqslant u_f(T) < 0$). ∎

The following lemma is a version of Gronwall's lemma.

LEMMA A3. *Let $e = e(x)$ be a continuous nonnegative function in $[a, b]$ such that*

$$e(x) \leqslant L \int_a^x \int_a^t e(\theta)\, d\theta\, dt + \tfrac{1}{2} A(x - a)^2 + B(x - a) + K, \qquad x \in [a, b],$$

where $L,\ A,\ B,\ K \geqslant 0$. *Then we have for* $x \in [a, b]$

$$e(x) \leqslant \frac{1}{2}\left(\frac{A}{L}+\frac{B}{\sqrt{L}}+K\right)\exp\left(\sqrt{L}(x-a)\right) + \frac{1}{2}\left(\frac{A}{L}-\frac{B}{\sqrt{L}}+K\right)\exp\left(-\sqrt{L}(x-a)\right)-\frac{A}{L}.$$

(*The case* $L=0$ *is meant as* $L \to 0^+$). ∎

Acknowledgements. I wish to thank G. Wasilkowski, A. G. Werschulz and H. Woźniakowski for their comments on this paper.

References

[1] P. Hartman, *Ordinary Differential Equations*, Birkhäuser, Boston 1982.
[2] M. Jankowski, H. Woźniakowski, *The accurate solution of certain continuous problems using only single precision arithmetic*, BIT 25 (1985), 635–651.
[3] B. Z. Kacewicz, *Asymptotic error of algorithms for solving nonlinear problems*, J. Complexity 3 (1987), 41–56.
[4] —, *Minimum asymptotic error of algorithms for solving ODE*, to appear in J. Complexity (1988).
[5] A. Kiełbasiński, private communication.
[6] J. Stoer, R. Burlisch, *Introduction to Numerical Analysis*, Springer-Verlag, New York 1980.
[7] A. G. Werschulz, *Optimal solution of the problem of optimal control*, in preparation, 1988.
[8] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford 1965.

Presented to the Semester
Numerical Analysis and Mathematical Modelling
February 25—May 29, 1987