SIAM J. SCI. STAT. COMPUT.
Vol. 5, No. 4, December 1984

AN EFFICIENT ALGORITHM FOR SOLVING GENERAL LINEAR TWO-POINT BVP*

R. M. M. MATTHEIJ† AND G. W. M. STAARINK‡

Abstract. A new method is described to compute the solutions of linear BVP in an efficient and stable way. The stability is achieved by decoupling the multiple shooting recursion; this means that the choice of output points can be made virtually without regard to restrictions. By fixing the number of integration steps per "shooting" interval and assembling as many of them as is needed to fit the user's requirements, high efficiency is gained. Apart from a mathematical description, we also give a stability analysis of the method. A large number of numerical examples confirm this analysis and illustrate the possibilities of the algorithm.

Key words. linear boundary value problem, multiple shooting, decoupling, adaptivity

AMS (MOS) classification. 65L10

1. Introduction. There exists an ever growing literature on two-point-boundary-value problems that has produced a large number of methods for solving them, see e.g. [1], [2], [4], [5], [6], [8], [9], [17], [18]. Still, even for linear problems there is room for improvements. These are related to questions regarding the efficiency of the method, in particular, for more general boundary conditions (BC), the flexibility with respect to the choice of output points (whether user requested or code determined) and last but not least the problem of how to control the stability. In this paper we shall describe a multiple shooting algorithm which grew out of a number of ideas, developed in part in previous work cf. [11], [12], [13], [15] and matured while attempting to write a general purpose FORTRAN code MUTS. We consider the linear ODE

$$\frac{dx}{dt} = L(t)x + f(t), \qquad \alpha \leqq x \leqq \beta, \tag{1.1}$$

where L is an $n \times n$ matrix function and f an n vector function, and assume that the solution x satisfies the BC

$$M_\alpha x(\alpha) + M_\beta x(\beta) = b, \tag{1.2}$$

where M_α and M_β are $n \times n$ matrices and b is an n vector. We consider mildly stiff ODE only (i.e. $|L|$ has no very large eigenvalues). Since our method employs orthogonalization at the shooting points it is a remote cousin of the Godunov–Conte algorithm [3] (see also the implementation in [18]). We like to emphasize, however, that our motivation is different. While orthogonalization in [3], [18] is used to maintain (or rather "restore") independence of a number of basic solutions, we feel that decoupling of solution spaces into increasing and decreasing modes is the key to get around the inherent (initial value) instability. Therefore, in our opinion, it is the *triangularization* (at least block triangularization) of the incremental matrices, that is crucial for this technique and we shall give some remarkable examples that underline this. Triangularization of the multiple shooting recursion makes the use of (special) sparse matrix solvers, like in [2], or (perhaps not always stable) initial value recursion

* Received by the editors June 15, 1982, and in revised form May 20, 1983.

† Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, New York 12181. On leave from Mathematisch Instituut, Katholieke Universiteit, Toernooiveld, Nijmegen, the Netherlands.

‡ Mathematisch Instituut, Katholieke Universiteit, 6525 ED Toernooiveld, Nijmegen, the Netherlands.

techniques, like in [5], [20], superfluous. Like most codes we employ an adaptive integrator. Another novelty is that we use efficiency arguments to select the so-called minor shooting intervals (cf. [15]). The so-called major shooting points (where the solution will be given as output) are a subset of the previous set and can be specified by the user, either directly or indirectly. Using the decoupling it can be shown that assembly of minor into major intervals does not affect the global errors, thus insuring stability and reasonable flexibility in the choice of output points.

In § 2 we first give a description of the triangularization technique. The actual algorithm is given in § 3. The stability of the method is considered in § 4 and we conclude with a number of numerical examples in § 5.

2. Triangularization of multiple shooting recursions. In a multiple shooting algorithm, the interval $[\alpha, \beta]$ is divided into a number, say N, of subintervals, $[t_i, t_{i+1}]$, $i=0, \cdots, N-1$. On each interval $[t_i, t_{i+1}]$ a fundamental solution, say F_i, and a particular solution, say w_i, is computed. Hence for each i there exists a vector y_i such that

$$x(t) = F_i(t)y_i + w_i(t), \qquad i=0, \cdots, N. \tag{2.1}$$

(N.B. the relation for $i=N$ is added to make the formulae later on look nicer.) By matching the relations (2.1) at t_{i+1} for $i=0, \cdots, N-1$ we obtain the recurrence relation

$$F_{i+1}(t_{i+1})y_{i+1} = F_i(t_{i+1})y_i + w_i(t_{i+1}) - w_{i+1}(t_{i+1}). \tag{2.2}$$

Suppose we choose the fundamental solutions F_i such that $\forall_i\, F_i(t_i) = I$, then

$$A_i := F_i(t_{i+1}) \tag{2.3}$$

is the incremental matrix (Wronskian) on $[t_i, t_{i+1}]$. If we let

$$g_i := w_i(t_{i+1}) - w_{i+1}(t_{i+1}) \tag{2.4}$$

then (2.2) reads

$$y_{i+1} = A_i y_i + g_i, \qquad i=0, \cdots, N-1. \tag{2.5}$$

From (1.2) and (2.1) we therefore see that the sequence $\{y_i\}_{i=0}^{N}$ must satisfy the BC

$$M_\alpha y_0 + M_\beta y_N = c := b - M_\alpha w_0(t_0) - M_\beta w_N(t_N). \tag{2.6}$$

The relations (2.5) and (2.6) together constitute the multiple shooting equations, cf. [5], [8], [12], [16], [17], [20]. Rather than solving them by some linear system solver, cf. [2], [8], [21], we use the recursion (2.5), in a way different from the approaches in [5], [20].

In order to understand the basic idea of our algorithm it is useful to assume that the homogeneous solution space of (1.1) is *dichotomic*, i.e. such that for some k there exists a k dimensional subspace of increasing solutions and an $(n-k)$ dimensional subspace of nonincreasing solutions. This implies that forward recursion of (2.5) is unstable (recall that A_i was the incremental matrix). We therefore apply the decoupling method of [10], [11] in order to compute growing and nongrowing components separately. This goes as follows: let Q_0 be a given orthogonal matrix (see § 3). Then compute recursively a sequence of orthogonal matrices $\{Q_i\}_{i=1}^{N}$ and upper triangular matrices $\{U_i\}_{i=0}^{N-1}$ such that

$$A_i Q_i = Q_{i+1} U_i, \qquad i=0, \cdots, N-1. \tag{2.7}$$

Writing

$$\tilde{y}_i := Q_i^{-1} y_i, \qquad \tilde{g}_i := Q_{i+1}^{-1} g_i, \tag{2.8}$$

we obtain the triangular recursion

$$\tilde{y}_{i+1} = U_i \tilde{y}_i + \tilde{g}_i, \qquad i = 0, \cdots, N-1. \tag{2.9}$$

It has been shown in [10], [11] that this decoupling gives U_i of which the $k \times k$ left upper block represents the increments of the (transformed) increasing solutions and the $(n-k) \times (n-k)$ right lower block the increments of the nonincreasing solutions, if Q_0 is chosen appropriately. (Observe that the rounding error in U_i is $O(\|A_i\| \varepsilon_M)$, where ε_M denotes the *machine epsilon.*) We now partition matrices and vectors as

$$U_i = \begin{pmatrix} B_i & C_i \\ \varnothing & E_i \end{pmatrix}, \qquad \tilde{y}_i = \begin{pmatrix} \tilde{y}_i^1 \\ \tilde{y}_i^2 \end{pmatrix}, \tag{2.10}$$

(where B_i is a $k \times k$ matrix and y_i^1 is a k vector) and employ the following decoupled recursions

$$\begin{aligned} &\text{(a)} \quad \tilde{y}_{i+1}^2 = E_i \tilde{y}_i^2 + \tilde{g}_i^2, \qquad i = 0, \cdots, N-1, \\ &\text{(b)} \quad B_i \tilde{y}_i^1 = \tilde{y}_{i+1}^1 - C_i \tilde{y}_i^2 - \tilde{g}_i^1, \qquad i = N-1, \cdots, 0. \end{aligned} \tag{2.11}$$

On account of the growth behaviour of the E_i and B_i, we expect $\|\prod_0^{i-1} E_j\|$, $\|[\prod_i^{N-1} B_j]^{-1}\| = O(1)$,[1] i.e. the recursions (2.11) are expected to be stable. This decoupled form is now employed to compute some particular solution of (2.9) and also a fundamental solution of the homogeneous part of (2.9). The desired particular solution $\{\tilde{y}_i\}$ then follows by superposition using the BC. The computation of these solutions goes as follows: Let the particular solution $\{\tilde{z}_i\}_{i=0}^N$ satisfy

$$\tilde{z}_0^2 = 0, \qquad \tilde{z}_N^1 = 0. \tag{2.12}$$

Then $\{\tilde{z}_i^2\}_{i=0}^N$ can be found in a stable way using (2.11)(a), and $\{\tilde{z}_i^1\}_{i=N}^0$ using (2.11)(b) (note that the $C_i \tilde{z}_i^2$ terms are known now). Let the fundamental solution $\{\tilde{\Phi}_i\}_{i=0}^N$ satisfy

$$\tilde{\Phi}_0^2 = [\varnothing \quad I_{n-k}], \qquad \tilde{\Phi}_N^1 = [I_k \quad \varnothing]. \tag{2.13}$$

Then $\{\tilde{\Phi}_i^2\}_{i=0}^N$ can be computed from the homogeneous part of (2.11)(a) (i.e. with $\forall_i \tilde{g}_i^2 = 0$) and $\{\tilde{\Phi}_i^1\}_{i=N}^0$ from the homogeneous part of 2.11(b) (i.e. with $\forall_i \tilde{g}_i^1 = 0$).

Clearly, for some fixed vector a we must have

$$\tilde{y}_i = \tilde{z}_i + \tilde{\Phi}_i a, \qquad i = 0, \cdots, N. \tag{2.14}$$

If we substitute this for $i = 0, N$ in the BC, we obtain a simple equation for a, viz

$$Ra = c - M_\alpha Q_0 \tilde{z}_0 - M_\beta Q_N \tilde{z}_N, \tag{2.15}$$

where

$$R := M_\alpha Q_0 \tilde{\Phi}_0 + M_\beta Q_N \tilde{\Phi}_N. \tag{2.16}$$

Hence the solution to the recursion (2.5), satisfying (2.6) is given by

$$y_i = Q_i [\tilde{z}_i + \tilde{\Phi}_i a], \qquad i = 0, \cdots, N. \tag{2.17}$$

The stability of this method will be discussed in § 4.

[1] $\prod_{j=p}^q E_j$ is defined as $E_q E_{q-1} \cdots E_p$ if $q \geqq p$ and as I otherwise.

3. The multiple shooting algorithm. In this section we describe a multiple shooting algorithm that is designed to obtain flexibility with respect to output points and reliability with respect to the stability of all computations involved. It employs the ideas described in [15] and § 2. In § 3.1 we show how the orthogonalization and triangularization is actually implemented. In § 3.2 we indicate how the integration of the ODEs is performed and how this is related to selecting the shooting points. In § 3.3 we consider the question how the output points may be chosen. In § 3.4 we show how an appropriate matrix Q_0 and the proper partitioning of the matrices U_i (which was an important prerequisite for the stability of the recursions (2.11)) can be found. Finally, in § 3.5 we give a special strategy for choosing the particular solutions w_i in order to obtain higher efficiency, in cases where the desired solution is very smooth.

3.1. Computation of the upper triangular recursion. In an actual implementation, the matrices A_i (cf. (2.3), (2.5)) do not appear explicitly as we directly use the orthogonal matrices to define initial values of suitable fundamental solutions. This goes as follows: On $[t_0, t_1]$ we compute a fundamental solution $\hat{F}_0$ say with

$$\hat{F}_0(t_0) := Q_0. \tag{3.1}$$

At t_1, decompose $\hat{F}_0(t_1)$ as

$$\hat{F}_0(t_1) = Q_1 U_0, \quad Q_1 \text{ orthogonal}, \quad U_0 \text{ upper triangular}. \tag{3.2}$$

On $[t_1, t_2]$ we proceed with the fundamental solution $\hat{F}_1$, satisfying

$$\hat{F}_1(t_1) := Q_1. \tag{3.3}$$

In general, we compute on $[t_i, t_{i+1}]$ the fundamental solution $\hat{F}_i$ with

$$\hat{F}_i(t_i) := Q_i, \tag{3.4}$$

and decompose

$$\hat{F}_i(t_{i+1}) = Q_{i+1} U_i. \tag{3.5}$$

Obviously the $\hat{F}_i$ are nothing but the transformed F_i of § 2, i.e. there holds

$$\hat{F}_i(t) = Q_i F_i(t). \tag{3.6}$$

As for the particular solution, we choose

$$w_i(t_i) := 0, \qquad i = 0, \cdots, N. \tag{3.7}$$

This means that

$$\tilde{y}_i = Q_i^{-1} x(t_i). \tag{3.8}$$

The transformed matching recursion then reads

$$\tilde{y}_{i+1} = U_i \tilde{y}_i + Q_{i+1}^{-1} w_i(t_{i+1}). \tag{3.9}$$

3.2. Adaptive integration and optimal complexity. A basic part of the algorithm is the integration of the particular and the fundamental solutions by some adaptive method. At present there are no codes available that are specially designed for integration both of increasing solutions and of nonincreasing solutions. Nevertheless, if the problem is not too stiff (in forward and backward direction) most currently available methods perform quite well. On account of its simplicity a fourth-fifth-order Runge–Kutta–Fehlberg method [7] is used in MUTS. A more detailed discussion of the use of this method for a nearly optimal shooting strategy can be found in [15]; we

use the results obtained there. First, the adaptivity feature is only employed to compute the particular solutions w_i on some grid on which the *number of points* is fixed (say 5). Then the fundamental solution is integrated by the fourth order method on the same grid. The matrix at the last grid point is decomposed into an orthogonal and an upper triangular matrix as described in § 2.

In order to understand why this strategy makes sense, we first remark that each solution (homogeneous or inhomogeneous) is likely to contain a multiple of the most dominant mode. Since this mode essentially dictates the steplength, there is no need to use the adaptivity feature for more than one solution. Second, the QU-decomposition is relatively cheap compared to an integration step (about a factor $1/5$) and therefore the complexity is mainly governed by the costs of integration. On account of the presence of increasing modes, larger intervals tend to make the integration less efficient; small intervals can make the overhead due to initialization etc. too high. Therefore a small (fixed) number of integration steps per shooting interval is preferred. This has two additional advantages. In the first place the shooting points are equidistributed (regarding the growth of the dominant mode) and in the second place we can choose our output points from a fairly dense grid. We remark that usually this strategy leads to a larger number of points than one may be interested in. In the next subsection we return to the latter question.

3.3. Assembly of minor shooting intervals. The strategy of § 3.2 not only gives us more points than desired, usually, but also more than desirable from a storage point of view. Indeed, in principle we have to store the matrices Q_i, U_i and the vectors $\tilde{g}_i$ at each shooting point. We propose two criteria for picking a subset suitable as output points. The first one is to choose points that correspond to an interval on which the most dominant mode does not grow more than a preset value; this results in a global equidistribution of these (major) shooting points. The second criterion is to take just a prescribed number of points. Both criteria can be used while marching from t_0 to t_N by checking either the increments or the interval length. In composing such a *major shooting interval* we compute an updated incremental matrix and an inhomogeneous term at each step. Hence the number of incremental matrices to be stored is equal to the (smaller) number of major shooting intervals.

Suppose we want the incremental growth to be bounded by M. This assembly of (*minor*) shooting intervals then goes as follows. Let t_{i_j} be the initial point of a major shooting interval. Define

$$W_0 := U_{i_j}, \qquad \tilde{G}_j := \tilde{g}_j. \tag{3.10}$$

If $\|W_0\| \geqq M$, then $t_{i_{j+1}} := t_{i_j+1}$, i.e. the minor interval is a major interval. Suppose this is not the case. Then for $s = 1, 2, \cdots$ we compute

$$W_s := U_{i_j+s} W_{s-1}, \qquad \tilde{G}_s := U_{i_j+s} \tilde{G}_{s-1} + \tilde{g}_{i_j+s}, \tag{3.11}$$

until $\|W_s\| \geqq M$. As a major interval we take (t_{i_j}, t_{i_j+s+1}) and define

$$V_j := W_s, \qquad \tilde{H}_j := \tilde{G}_s \quad \text{(also if } s = 0\text{)}. \tag{3.12}$$

The global (major) recursion for the sequence $\{\tilde{y}_{i_j}\}$, then reads

$$\tilde{y}_{i_{j+1}} = V_j \tilde{y}_{i_j} + \tilde{H}_j. \tag{3.13}$$

At this point one might be suspicious whether we destroyed the advantages of our algorithm, as the updating in (3.11) is nothing but forward recursion! However, there is not a real threat for two reasons. First, we may choose M such that $M\varepsilon_M$ (ε_M

the machine accuracy) is still smaller than the required tolerance (cf. [15]). Second, since the recursions are decoupled the error propagation is very special and we show in § 4.3 that no significant error build-up, due to this assembly, is felt in the final solution approximant, however large M may be chosen.

3.4. Choosing an appropriate Q_0 and the proper partitioning. As we remarked in § 2 we need to choose Q_0 properly in order to make sure that the induced upper triangular matrices U_i have the desired ordering, i.e. such that the left upper blocks represent the incremental values of the increasing modes. Before we show how this is achieved in our algorithm it is useful to investigate how different choices of the initial value matrix Q_0 induce different sequences of orthogonal and upper triangular matrices and how these are related. So let $\hat{Q}_0$ be another initial value matrix and $\{\hat{Q}_i\}_{i=0}^{N}$ and $\{\hat{U}_i\}_{i=0}^{N-1}$ be the induced orthogonal and upper triangular matrices (cf. (2.7)), i.e.

$$A_i\hat{Q}_i = \hat{Q}_{i+1}\hat{U}_i. \tag{3.14}$$

Write (cf. (2.8))

$$\hat{y}_i := \hat{Q}_i^{-1}y_i, \qquad \hat{g}_i := \hat{Q}_{i+1}^{-1}g_i. \tag{3.15}$$

Then $\{\hat{y}_i\}$ satisfies

$$\hat{y}_{i+1} = \hat{U}_i\hat{y}_i + \hat{g}_i. \tag{3.16}$$

The recursions (2.9) and (3.16) may be called *equivalent* since we can define an equivalence relation by introducing

$$S_i := Q_i^{-1}\hat{Q}_i. \tag{3.17}$$

It is easy to see that the following equalities hold

$$\begin{aligned} &\text{(a)}\quad U_iS_i = S_{i+1}\hat{U}_i, \\ &\text{(b)}\quad \hat{g}_i = S_{i+1}^{-1}\tilde{g}_i, \\ &\text{(c)}\quad \hat{y}_i = S_i^{-1}\tilde{y}_i. \end{aligned} \tag{3.18}$$

In our algorithm, we like Q_0 to be chosen such that the B_i reflect the growth of the increasing solutions. This may be measured by inspecting the diagonal elements (= eigenvalues); in particular if the (major) shooting interval is not too small, this seems a suitable strategy. Assuming there is a global dichotomy (i.e. on the entire interval $[t_0, t_N]$) it suffices to make sure that the first assembled matrix V_0 (see § 3.3) has a properly ordered diagonal. This is done as follows. We start at $t = t_0$ with $Q_0 = I$. Due to the tendency of the triangularization to give ordered diagonals in the upper triangular matrices (cf. power method arguments, see also [10], [11]), this will quite often be a satisfactory choice to achieve our goal. Anyway, at t_1 we check whether diag (U_0) is ordered. If this is not the case, the columns of U_0 are reordered according to the absolute magnitude of the diagonal elements. This can be described by a permutation matrix $P_0^1(1)$ say (the one between parentheses indicates that the checking has been performed at $t = t_1$ and the superscript one that it is the first permutation trial). The permuted matrix is again decomposed into an orthogonal and an upper triangular matrix, say

$$U_0(0)P_0^1(1) = P_1^1(1)U_0^1(1), \tag{3.19}$$

where $U_0(0) := U_0$.

If the matrix $U_0^1(1)$ is not yet ordered, this procedure has to be repeated. One should realize, however, that disordering is caused by the fact that the subspace spanned

by the first k column vectors of Q_0 makes an extremely small angle with an initial value of a nonincreasing mode (cf. [11, § 5.1]); therefore in the (*quite unlikely*!) case of a disordered U_0 we expect such a permutation process to give a desired result after a few steps only, in which we have performed, for $j=1,\cdots,p$ say,

$$U_0^{j-1}(1)P_0^j(1)=P_1^j(1)U_0^j(1). \tag{3.20}$$

Writing

$$S_0(1):=P_0^1(1)\cdots P_0^p(1);\ S_1(1):=P_1^p(1)\cdots P_1^1(1);\ U_0(1):=U_0^p(1), \tag{3.21}$$

we thus have

$$U_0(0)S_0(1)=S_1(1)U_0(1). \tag{3.22}$$

If we indicate the original sequences $\{Q_i\}$ and $\{U_i\}$ by $\{Q_i(0)\}$ and $\{U_i(0)\}$, then using $Q_0(1):=S_0(1)$ as an initial matrix induces sequences of orthogonal matrices $\{Q_i(1)\}$ and upper triangular matrices $\{U_i(1)\}$. Obviously we thereby obtain an equivalent upper triangular recursion of which the transformation matrices are related by (cf. (3.17))

$$S_i(1):=[Q_i(0)]^{-1}Q_i(1). \tag{3.23}$$

(Note that $Q_0(0)=I$.) Since we are building a major shooting step (cf. § 3.3), we next check whether

$$W_1(1):=U_1(1)U_0(1) \tag{3.24}$$

is ordered. If this is not the case we permute columns of $W_1(1)$ and decompose, as we did before with $U_0(1)$,

$$W_1(1)S_0(2)=S_2(2)W_1(2) \tag{3.25}$$

where $S_2(2)$ is orthogonal and $W_1(2)$ is upper triangular. Now $S_0(2)$ induces another equivalent recursion which would follow from the transformations $\{Q_i(2)\}$ with $Q_0(2):=S_0(1)S_0(2)$ and for which

$$S_i(2):=[Q_i(1)]^{-1}Q_i(2). \tag{3.26}$$

At this step the inhomogeneous term is updated like

$$\tilde{G}_1(2):=[S_2(2)]^{-1}\{U_1(1)\tilde{G}_0(1)+\tilde{g}_1(1)\}, \tag{3.27}$$

where $\tilde{g}_j(1):=Q_j^{-1}(1)\tilde{g}_j$ (cf. also (3.18)(b)). Typically at this point we have to store the most recently found initial matrix $Q_0(2)$, and also $Q_2(2)$, $W_1(2)$ and $\tilde{G}_1(2)$. These four arrays are overwritten at each consecutive step by the new initial transformation matrix $Q_0(\cdot)$ (only if reordering of diagonal elements is needed), the most recent orthogonal factorization matrix $Q_{i+1}(\cdot)$, the most recent assembled upper triangular matrix $W_i(\cdot)$ and the most recent assembled forcing $\tilde{G}_i(\cdot)$ respectively. This updating is continued until the endpoint of the first major shooting interval is reached. At subsequent points no reordering is performed; if the solution space is dichotomic this seems quite reasonable. However, if there is no global dichotomy the ordering found at the beginning may not be the appropriate one globally. Therefore, the algorithm finally checks the product of the diagonals of the upper triangular matrices. If this turns out not to be ordered, a new round of permutations is performed giving new transformation matrices at the major shooting points and new upper triangular matrices describing increments between them (carried out as described in (2.7)).

Two remarks must be made. First, due to the tendency of larger increments to be in the left upper block if not by rounding errors then at least by (usually larger) discretization errors, there is a "natural" self-restoring effect (though not always appropriate, see Example 5.2). Second, if the problem is not dichotomic, it cannot be expected to be a very well conditioned one, as follows from [14, Thm. 3.17]; in other words, it is the problem, rather than the algorithm that is to be blamed for possible instabilities! In § 4.1 we indicate how we may monitor this instability. After the global ordering has been found to be satisfactory, a value of k, the dimension of the dominant subspace, is determined from inspection of the diagonal elements.

3.5. Special choice of the w_i, when the solution is very smooth. One of the main problems in multiple shooting is that the efficiency of the integration is almost always dictated by the most rapidly increasing modes. This is unsatisfactory if the solution x is very smooth (and most multiple shooting codes suffer from this problem). In looking for ways to make our code as efficient as possible, we have experimented with a version that chooses particular solutions w_i in which the dominant mode component is much less influential. The idea is conceptually very simple and is another evidence of the usefulness of decoupling the recursion while marching from α to β. First we define some $\bar{w}_i(t_i)$, equal to $w_{i-1}(t_i)$ except for components that may belong to the dominant modes, by projection of $w_{i-1}(t_i)$ on the dominant solution space. Specifically,

$$\bar{w}_i(t_i) := w_{i-1}(t_i) - Q_i\begin{pmatrix}[Q_i^{-1}w_{i-1}(t_i)]^1\\ \varnothing\end{pmatrix} = Q_i\begin{pmatrix}\varnothing\\ [Q_i^{-1}w_{i-1}(t_i)]^2\end{pmatrix}. \tag{3.28}$$

Because of the smoothness we now expect

$$\tilde{y}_i \approx \tilde{y}_{i-1}, \tag{3.29}$$

so

$$\tilde{y}_i^1 \approx (I - B_{i-1})^{-1}(C_{i-1}\tilde{y}_i^2 + \tilde{g}_{i-1}^2). \tag{3.30}$$

If we are optimistic enough, we may hope that $\tilde{y}_i \approx \tilde{w}_{i-1}(t_i)$, apart from dominated modes. Therefore we propose to use as initial value of w_i at t_i:

$$w_i(t_i) := Q_i\begin{pmatrix}(I - B_{i-1})^{-1}(C_{i-1}[\tilde{w}_{i-1}(t_i)]^2 + \tilde{g}_{i-1}^1)\\ [\tilde{w}_{i-1}(t_i)]^2\end{pmatrix}. \tag{3.31}$$

For a numerical justification, see Example 5.6.

4. The stability of the algorithm. In this section we consider the numerical stability of the different parts of the algorithm. Although the solution of the linear system (2.15) describes the final stage of the method, we analyze this first in § 4.1 which deals with the important notion of well conditioning. Then in § 4.2 we investigate the stability of the recursions (2.11)(a) and (b). Finally in § 4.3 we analyze the effect of assembling the recursion as was described in § 3.3.

4.1. Well conditioning and stability. As was shown in [12], [13] a useful quantity for studying the inherent stability of a BVP with respect to the BC is given by the condition number

$$\mathscr{CN} := \max_t \|F(t)[M_\alpha F(\alpha) + M_\beta F(\beta)]^{-1}\|, \tag{4.1}$$

where F is any fundamental solution. In particular, if we neglect discretization and rounding errors, we have for the fundamental solution Φ of the recursion (2.5), where

$\forall_j \Phi_j := Q_j \tilde{\Phi}_j$:

$$\max_j \|\Phi_j R^{-1}\| \leqq \mathscr{CN}. \tag{4.2}$$

As was shown in [13], the condition number enables us to give fairly straightforward estimates for the global error if the solution space is dichotomic. We can also prove that this condition number is a good measure for the stability of the equation (2.15), see the following theorem.

THEOREM 4.3. *Let* $\|\cdot\| = \|\cdot\|_2$. *Then* $\|R^{-1}\| \leqq 2\mathscr{CN}$.

Proof. Denote $\kappa := \max \|\Phi_j R^{-1}\|$. Then it follows that $\|\tilde{\Phi}_0^2 R^{-1}\| \leqq \|\tilde{\Phi}_0 R^{-1}\| = \|\Phi_0 R^{-1}\| \leqq \kappa$, and similarly $\|\tilde{\Phi}_N^1 R^{-1}\| \leqq \kappa$. Hence

$$\|R^{-1}\| = \left\| \begin{pmatrix} \tilde{\Phi}_N^1 \\ \tilde{\Phi}_0^2 \end{pmatrix}^{-1} \begin{pmatrix} \tilde{\Phi}_N^1 \\ \tilde{\Phi}_0^2 \end{pmatrix} R^{-1} \right\| \leqq \|\tilde{\Phi}_N^1 R^{-1}\| + \|\tilde{\Phi}_0^2 R^{-1}\| \leqq 2\kappa,$$

i.e.

$$\|R^{-1}\| \leqq 2\mathscr{CN}. \qquad \square$$

If we can compute $\tilde{\Phi}$ and $\tilde{z}$ stably, then the problem of computing the vector a from (2.15) is as well conditioned as the BVP itself!

4.2. The stability of the recurrence relation (2.11). In order to analyze the stability of (2.11), we examine the effects of additive perturbations $\{z_i^2\}$ and $\{z_i^1\}$ in (2.11)(a) and (b) respectively, like was done in [11, § 4]. So let $\{g_i\}$ satisfy the perturbed recursion

$$\begin{aligned} &\text{(a)}\quad g_{i+1}^2 = E_i g_i^2 + z_{i+1}^2, \qquad g_0^2 = z_0^2, \\ &\text{(b)}\quad g_i^1 = B_i^{-1}\{g_{i+1}^1 - C_i g_i^2\} + z_i^1, \qquad g_N^1 = z_N^1, \end{aligned} \tag{4.4}$$

where one may think of $\|z_i^1\|, \|z_i^2\|$ to be of the order of $\|A_i\|\varepsilon_M$ (cf. § 2). We then obtain

$$\begin{aligned} &\text{(a)}\quad g_i^2 = \sum_{l=0}^{i} \left(\prod_{j=l}^{i-1} E_j \right) z_l^2, \\ &\text{(b)}\quad g_i^1 = \sum_{l=0}^{i} \left\{ \Omega_{i,N} \left(\prod_{j=l}^{i-1} E_j \right) z_l^2 \right\} + \sum_{l=i+1}^{N-1} \left\{ \left(\prod_{j=i}^{l-1} B_j \right)^{-1} \Omega_{l,N} z_l^2 \right\} + \sum_{l=i}^{N} \left\{ \left(\prod_{j=i}^{l-1} B_j \right)^{-1} z_l^1 \right\}, \end{aligned} \tag{4.5}$$

where $\Omega_{p,q}$ is a shorter notation for

$$\Omega_{p,q} = -\sum_{l=p}^{q-1} \left\{ \left(\prod_{j=p}^{l} B_j \right)^{-1} C_l \left(\prod_{j=p}^{l-1} E_j \right) \right\}. \tag{4.6}$$

Now if the partitioning is chosen correctly and if there is a dichotomic solution space, then it follows that $\|\prod_{j=l}^{i} E_j\|$ and $\|(\prod_{j=l}^{i} B_j)^{-1}\|$ are of order one. Moreover, if the solutions are directionally well separated, $\|\Omega_{p,q}\|$ will not be large (cf. [10, Rem. 6.13]. Hence contamination of errors will not produce large errors.

It was shown in [11] that a wrong choice for Q_0 (i.e. the span of the first k column vectors being close to an initial value of a decreasing mode) produces large $\|E_j\|$ and large $\|B_j^{-1}\|$ initially and therefore usually large $\Omega_{p,q}$, for p small. This led us to the permutation updating of § 3.4; in Example 5.2 we shall show how important this "optimal" choice for Q_0 may be. If there is no dichotomic solution space, one should hope that $\max_{i,l} \|\prod_{j=l}^{i} E_j\|$ and $\max_{i,l} \|(\prod_{j=l}^{i} B_j)^{-1}\|$ are not so large as to produce

global rounding errors that threaten the desired discretization error. However because of a result given in [14, Thm. 3.17] we are inclined to believe that this can only happen for problems that are inherently unstable.

4.3. The effect of assembling on the global error. The last and perhaps most intriguing part of this stability analysis is to show that the assembly does not influence (at least not significantly) the global rounding error. This can be proven using an adapted version of the error analysis of [13], which we shall give below.

As before, let ε_M denote the machine epsilon. Then a realistic description of the rounding error in a major shooting incremental matrix V_j, assembled going from $t_{i_{j-1}}$ to t_{i_j}, is given by $\varepsilon_M \tilde{\Phi}_{i_j}[\tilde{\Phi}_{i_{j-1}}]^{-1} D_j$ where D_j is some matrix with $\|D_j\| = O(1)$. Similarly the inhomogeneous terms at t_i are perturbed by something like $\varepsilon_M \tilde{\Phi}_{i_j}[\tilde{\Phi}_{i_{j-1}}]^{-1} d_j$, where $\|d_j\| = O(1)$.

For simplicity we only investigate the global error caused by the latter perturbations. Therefore we introduce discrete Green's functions $Z_j(s)$ defined by

$$\text{(4.7)} \qquad \begin{aligned} &\text{(a)}\ Z_j(s) = V_j Z_{j-1}(s) + \Delta_{js}, \\ &\text{(b)}\ M_\alpha Q_0 Z_0(s) + M_\beta Q_N Z_N(s) = 0, \end{aligned}$$

and where

$$\text{(4.8)} \qquad \Delta_{js} = \begin{cases} \tilde{\Phi}_{i_j}[\tilde{\Phi}_{i_{j-1}}]^{-1} & \text{if } j = s, \\ 0 & \text{if } j \neq s. \end{cases}$$

Similarly to [13, (3.5)(a)] we get

$$\text{(4.9)} \qquad Z_j(s) = -\tilde{\Phi}_{i_j}(R^{-1} M_\beta Q_N \tilde{\Phi}_N)\tilde{\Phi}_{i_s}^{-1}, \qquad j \leqq s,$$

the only distinction with the analogous formulae in [13] being that the Green's functions here are a "factor" $\tilde{\Phi}_{i_j}[\tilde{\Phi}_{i_{j-1}}]^{-1}$ bigger. Using these $Z_j(s)$ we obtain for the global error due to the above indicated perturbations

$$\text{(4.10)} \qquad e_j = \varepsilon_M \sum_s Z_j(s) d_s,$$

which is analogous to [13, Prop. 4.5].

Property 4.11. Assume that the solution space satisfies similar splitting properties as in [13], viz. there exist positive ν and μ, such that the increasing solutions grow at least like $\exp[(\mu(t_{i_j} - t_{i_{j-1}})]$ and the decreasing solutions at most like $\exp[-\nu(t_{i_j} - t_{i_{j-1}})]$. Then the global error (4.10) is estimated by

$$\|e_j\| \leqq \bar{\chi}(\mathscr{C}\mathscr{N} + 1)\left(\frac{\bar{\gamma}_1}{1 - e^{-\nu h}} + \frac{\bar{\gamma}_2}{1 - e^{-\mu h}}\right)\varepsilon_M \max_s \|d_s\|.$$

(N.B. $h = \min_s (t_{i_s} - t_{i_{s-1}})$, for $\mathscr{C}\mathscr{N}$ see (4.1); $\bar{\gamma}_1$ and $\bar{\gamma}_2$ are just constants appearing in estimating the basis solutions.)

Remark 4.12. Using the notation $\Omega_{p,q}$ (see (4.6)) we can write

$$\tilde{\Phi}_i = \begin{pmatrix} I & \Omega_{i,N} \\ \varnothing & I \end{pmatrix} \begin{pmatrix} \prod_{j=0}^{i-1} B_j & \varnothing \\ \varnothing & \prod_{j=0}^{i-1} E_j \end{pmatrix}.$$

Hence, by definition of T_i in [13], the so-called "direction matrix", we see that

$$T_i = \begin{pmatrix} I & \Omega_{i,N} K_i \\ \varnothing & K_i \end{pmatrix}$$

for some K_i with $\|K_i^{-1}\| \leqq 1$. Hence we see that

$$\bar{\chi} = \max_i \|T_i^{-1}\| \leqq \max_i \left\| \begin{pmatrix} I & \Omega_{i,N} \\ \varnothing & D_i^{-1} \end{pmatrix} \right\| \leqq 1 + \max_i \|\Omega_{i,N}\|.$$

Therefore, if the solutions are directionally well separated, implying that $\|\Omega_{i,N}\|$ is not large (cf. § 4.2), we have $\bar{\chi} = O(1)$.

The importance of the result in (4.11) is that it shows that the global effect of assembling is more or less independent of the length of the major shooting interval. Of course, one should realize that $\|d_s\|$ is of order $\sum_{l=i_{s-1}}^{i_s-1} \|A_l\|$ and hence about linear in the number of minor shooting intervals; however, this is not at all the exponential error growth that would occur in assembling coupled recursions! The crucial point is that, though the absolute errors in the computed $\|V_j\|$ may be large, they get smaller the lower the row index, i.e. relative to the increment of a certain mode as is given by a sequence of a certain column of the $\tilde{\Phi}_j$; hence, we have stability (thanks to our special computation by forward and backward recursion).

The problem that remains is: What happens when there is no dichotomy? This is a difficult question and certainly needs a more extensive study than we have made. Although one may be able to show that there exist a solution that does not increase and another that changes behaviour somewhere (like in turning point problems), a suitable linear combination of these may still exhibit dichotomy. In any case, assuming well conditioning, the dichotomy is assured, cf. [14]. Moreover, it is not difficult to give examples where lack of dichotomy leads to severe error growth, see Example 5.3.

In MUTS we have implemented a safety check in order to warn that global (rounding) errors may threaten the required accuracy. The best check would be to determine

$$\max_{p,q} \|\Omega_{p,q}\|, \max_{j,i} \left\| \prod_{l=j}^{i} E_l \right\| \quad \text{and} \quad \max_{j,i} \left\| \left(\prod_{l=j}^{i} B_j \right)^{-1} \right\|$$

and use this to get hold of the expression in (4.5). The next best check (as this is at least practically performable and easy to implement) is to compute

$$\rho = \max_{j,i} \left\| \prod_{l=j}^{i} \operatorname{diag}(E_l) \right\| \cdot \max_{j,i} \left\| \prod_{l=j}^{i} \operatorname{diag}(B_l^{-1}) \right\|.$$

Indeed, the maximal diagonal elements of the matrices $|B_l^{-1}|$ and $|E_l|$ are often a good estimate of their respective norms (cf. [11, see 6]). Hence if ρ is not large we can expect that both $\|\prod E_l\|$ and $\|(\prod B_l)^{-1}\|$ are $O(1)$. Since $\|B_l^{-1} C_l\|$ usually, is $O(1)$, we therefore may also use ρ as an estimate for $\max_{p,q} \|\Omega_{p,q}\|$. This estimate ρ is now used as follows. Suppose the user wants an accuracy tol. If the code detects that $\rho \varepsilon_M >$ tol, a warning error is given indicating that rounding errors may be blurring the result. However, in all test problems we noted that ρ also gave a fairly good estimate for the global discretization error amplification, see Examples 5.3 and 5.4.

5. Examples. In this section we give a number of numerical examples in order to illustrate the remarks and analyses above. All solutions of the following problems have been computed using the double precision code for the inhomogeneous problems,

DMUTSG, as was developed at Nijmegen cf. [19]. The computations were performed on an IBM 4341/MVS computer.

We did not make explicit comparisons with other existing codes, like PASVAR [9], COLSYS [1] or SUPORT [18]. The main reason is that a simple comparison of cpu time is not very meaningful, as the number of examples is too small to draw significant conclusions and/or the way each code is implemented may blur any relative theoretical efficiency predictions. For SUPORT there was the additional argument that this code is designed for separated BC only. As far as memory requirements are concerned it might be obvious that any reasonable multiple shooting code is superior to methods that necessarily need to store information at all grid points. However, if a user wants a lot of output points, the storage requirements of our code and so-called global methods become comparable again. We finally note that the triangularization of the recursion and the solution of resulting BVP has a similar complexity as solving a (sparse) multiple shooting system by some decomposition method (cf. [15]); an LU-decomposition method might, however, require a complicated pivoting strategy and additional memory space, whereas our method only needs $\approx N(n^2+n)$ numbers (N—the number of major shooting intervals) to store and is straightforward.

Example 5.1. Consider the ODE

$$\frac{dx}{dt}=\begin{pmatrix} 1-19\cos 2t & 0 & 1+19\sin 2t \\ 0 & 19 & 0 \\ -1+19\sin 2t & 0 & 1+19\cos 2t \end{pmatrix}x+f(t), \tag{5.1}$$

where $f(t)=e^t(-1+19(\cos 2t-\sin 2t), -18, 1-19(\cos 2t+\sin 2t))^T$ and the BC

$$x(0)+x(\pi)=(1+e^{\pi}, 1+e^{\pi}, 1+e^{\pi})^T. \tag{5.2}$$

The exact solution to this problem is $x=(e^t, e^t, e^t)^T$. The homogeneous part has solutions growing like $\sim e^{20t}$, $\sim e^{19t}$, $\sim e^{-18t}$, cf. [12, Example 6.2]. As requirements to have our solution approximated, we asked for an absolute tolerance of $1.0-6$ $(=10^{-6})$ and a maximal increment of homogeneous solutions $M=1.0+3$. (N.B. the code considers values between $.5M$ and $2.0M$ to be acceptable.) In Table 5.1 we give the result (up to two decimals). Note that the last major shooting interval is smaller than the rest. In Table 5.2 we give the result with tolerance $1.0-7$ (as before) but now with a maximal increment $M=1.0+30$. Note that this increment means that the *recursion* is solved by single shooting! The fact that the errors are so much smaller than was asked for is a typical vice of multiple shooting; indeed, as we let the integrator determine a fairly general solution (which most likely contains a component of increasing modes) within the required tolerance, the resulting grid usually gives a significantly smaller error for a smooth solution. Finally we remark that the cpu time for both cases was almost the same, as we noted in [15].

Example 5.2. Consider the same BVP as in Example 5.1. We used DMUTSG, now asking for output on 15 equally spaced points (hence the increment M per interval equals ≈ 100). Moreover we deliberately skipped that part where an optimal Q_0 is to be determined, so Q_0 was taken to be I. Since a fundamental solution of (5.1) is given by (cf. [12, Ex. 6.2])

$$F_0(t)=\begin{bmatrix} \sin t & 0 & -\cos t \\ 0 & 1 & 0 \\ \cos t & 0 & \sin t \end{bmatrix} \operatorname{diag}(e^{20t}, e^{19t}, e^{-18t}), \tag{5.3}$$

TABLE 5.1

i	t	U_{i-1}	$\|\text{error}\|_\infty$
0	0.00		1.9 −9
1	0.34	1.1 +3 0 2.5 −3 0 8.2 +2 0 0 0 1.7 −3	1.2 −9
2	0.70	7.0 +2 0 8.5 −4 0 5.0 +2 0 0 0 2.8 −3	1.8 −9
3	1.05	5.7 +2 0 5.0 −4 0 4.2 +2 0 0 0 3.2 −3	1.7 −9
⋮	⋮	⋮ ⋮ ⋮	⋮
9	2.83	1.7 +3 0 6.2 −5 0 1.2 +3 0 0 0 1.2 −3	1.9 −9
10	3.14	1.6 +1 0 3.8 −7 0 1.4 +1 0 0 0 8.1 −2	1.9 −9

TABLE 5.2

i	t	U_{i-1}	$\|\text{error}\|_\infty$
0	0		1.9 −9
1	3.14	1.9 +27 0 4.2 +21 0 8.4 +25 0 0 0 2.8 −25	1.9 −9

the 1st column of Q_0 generates a decreasing mode, if the computations were exact. In particular we can no longer hope that the B_i represent the increments of the increasing modes. However, due to discretization errors the *discrete* fundamental solution will most likely deviate from "the" exact solution; specifically, we can expect a discrete dominant solution to exist which has as initial value $(\varepsilon_1, \varepsilon_2, 1)^T$, where $\varepsilon_1, \varepsilon_2$ are of the order of the discretization error ε. Therefore, in practice this first column of Q_0 still contains a small component of the most unstable mode. After some time this mode has grown by a factor $1/\varepsilon$ and will become dominant; this can be seen from the U_i which will then have left upper blocks representing the increments of the dominant modes again. The consequence of this temporary "disorder" is that for smaller l both $\|[\prod_{j=0}^{l} B_j]^{-1}\|$ and $\|\prod_{j=0}^{l} E_j\|$ may be $O(1/\varepsilon)$ thus making the $\|\Omega_{0,q}\|$ (cf. (4.6)) larger. It can be shown that $\max_{p,q} \|\Omega_{p,q}\| = O(1/\varepsilon)$ is achievable (cf. [11, Ex. 5.2]). From the error analysis in § 4.2, cf. (4.5), it follows that we therefore may expect global errors of the order $\varepsilon_M/\varepsilon$. This is a funny result, since it means that for

ε smaller than $(\varepsilon_M)^{1/2}$, a smaller tolerance will give a larger global error (being a result of rounding errors!). Since we are working with a $\varepsilon_M \approx 1.0-16$ we expect this to happen if $\varepsilon \approx 1.0-8$. (Bearing in mind the "over-killing" effect, noted in Example 5.1, the threshold value of tol is $\approx 1.0-6$.) This is nicely demonstrated in Table 5.3. It is also instructive to see how the "errors" restore the proper ordering in the diagonal of the U_i after a few steps and how this has its impact on the error (see Table 5.4), where we only give the $(1,1)$, $(1,3)$ and $(3,3)$ element of the U_i, for a tolerance tol $= 1.0-10$.

TABLE 5.3

tol	max $\|\text{error}\|_\infty$	min $\|\text{error}\|_\infty$	"ε"
1.0 −4	2.2 −7	1.5 −7	1.0 −6
1.0 −6	5.0 −9	1.5 −9	1.0 −8
1.0 −8	6.6 −7	1.5 −11	1.0 −10
1.0 −10	1.1 −4	2.0 −13	1.0 −12

TABLE 5.4

i	U_{i-1}	$\|\text{error}\|_\infty$
0		1.1 −4
1	1.5 −2 1.9 −6 0 1.1 +2	2.3 −6
2	1.5 −2 1.4 −2 0 1.1 +2	3.9 −8
3	2.1 −2 1.2 +2 0 7.7 +1	7.0 −10
4	1.1 +2 6.5 +1 0 1.6 −2	9.5 −12
5	1.1 +2 1.1 −6 0 1.5 −2	2.3 −13
⋮	⋮ ⋮	⋮

Example 5.3. Consider the following ODE

$$\frac{dx}{dt} = \begin{pmatrix} \psi(t) & 0 \\ 2\psi(t) & -\psi(t) \end{pmatrix} x + \begin{pmatrix} (1-\psi(t))\, e^t \\ 2\, e^t \end{pmatrix} \tag{5.4}$$

where $\psi(t) = 20 \sin t + 20t \cos t$. Let the BC be given by

$$x(0) + x(T) = \begin{pmatrix} 1+e^T \\ 2(1+e^T) \end{pmatrix}, \qquad T > 0. \tag{5.5}$$

As one may check, a fundamental solution for (5.7) is given by

$$F_0(t) = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \operatorname{diag}(e^{\phi(t)}, e^{-\phi(t)}), \tag{5.6}$$

where $\phi(t) = 20t \sin t$. Apparently $x = \binom{1}{2} e^t$ satisfies (5.5) and (5.4). For larger values of T it can be seen that there does not exist a dichotomic solution space. In fact we have a kind of turning point problem. In Table 5.5 we show the dramatic effect of this lack of dichotomy on the global error by giving some results for $T = 2$, 2.5 and 3. We gave a tolerance of 1.0 −6 and asked for output at equally spaced points with distance 0.1. The quantity κ is an estimate for $\mathscr{CN}$ (cf. [12], [13]); ρ is defined in § 4.3, being an estimate for the skewness Ω. In order to understand this result, one should realize

TABLE 5.5

T	max $\|\text{error}\|_\infty$	min $\|\text{error}\|_\infty$	ρ	κ
2	4.2 −8	4.0 −10	1.5	2.4
2.5	2.4 −3	5.2 −10	4.0 +5	2.4
3	5.0 +4	1.8 −2	2.8 +11	3.9 +9

that for $t = 0$ and $t \approx 2.029$ we have a turning point, at which the increasing mode in F_0 becomes decreasing and the decreasing one increasing. Hence, even though there may be a globally dominant mode, we can no longer expect the diagonal of the upper triangular matrices U_i to be ordered for all i; the effect of this disordering is shown by the large ρ for $T = 2.5$ and the very large ρ for $T = 3$. For $T = 2.5$ this instability has a limited effect on the accuracy of the solutions (although we lose 3 digits compared to $T = 2$); also the conditioning of the system (2.15) (cf. also (4.1)) is still reasonable for $T = 2.5$. However, for $T = 3$, we may expect error amplification of the order of 1.0 +11 which nicely agrees with the result in the second column (note that we may expect local errors of the order of 1.0 −8 in this example, cf. the result for $T = 2$). It was interesting to see that direct use of a Crout routine to solve the multiple shooting system either gave slightly worse results or no result at all (e.g. for $T = 3$ the system was found to be numerically singular).

Example 5.4. Consider the ODE

$$\frac{dx}{dt} = \begin{pmatrix} t(1-\cos 2t) & 1 + t \sin 2t \\ -1 + t \sin 2t & t(1+\cos 2t) \end{pmatrix} x + f(t) \tag{5.7}$$

and a BC where $M_\alpha = M_\beta = I$. The function $f(t)$ and the vector b (in (1.2)) are chosen such that

$$x = \begin{pmatrix} 1 + \cos t \\ 1 - \sin t \end{pmatrix}.$$

A fundamental solution of (5.7) is given by

$$F_0(t) = \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix} \operatorname{diag}(1, e^{t^2}). \tag{5.8}$$

Hence, we see that the second basis solution changes at $t = 0$ from a decreasing solution to an increasing solution. Therefore we have a kind of dichotomy (i.e. a splitting

between nondecreasing and nonincreasing solutions) if either $[\alpha, \beta] \subset [0, \infty]$ or $[\alpha, \beta] \subset [-\infty, 0]$. In Table 5.6 below we have given errors and condition numbers for computations on the intervals $[0, 4]$, $[-2, 2]$ and $[-4, 4]$ respectively. The tolerance was set to $1.0 - 8$ and we asked for output on equally spaced points (distance $= 0.4$); ρ and κ are defined as in Example 5.3. Once again we see that ρ quite accurately predicts the loss

TABLE 5.6

$[\alpha, \beta]$	max $\|\text{error}\|_\infty$	min $\|\text{error}\|_\infty$	ρ	κ
$[0, 4]$	5.8 −9	2.0 −9	1.1 +1	4.1
$[-2, 2]$	3.9 −7	2.5 −7	2.2 +2	1.6
$[-4, 4]$	4.2 −2	2.2 −2	2.5 +7	1.1

of significant digits. Like in the previous example, linear algebraic methods to compute solutions of the multiple shooting system did not perform any better.

Example 5.5. As the last of a series of "turning point" problems consider the scalar problem

$$\frac{d^2\xi}{dt^2} + 40t\frac{d\xi}{dt} = (1+40t)\, e^t, \tag{5.9}$$

with BC $\xi(\alpha) = e^\alpha$, $\xi(\beta) = e^\beta$. Hence $\xi(t) = e^t$. Written in vector form this corresponds to the BVP

$$\text{(a)}\quad \frac{dx}{dt} = \begin{pmatrix} 0 & 1 \\ 0 & -40t \end{pmatrix} x + \begin{pmatrix} 0 \\ (1+40t)\, e^t \end{pmatrix}, \qquad x = \begin{pmatrix} \xi \\ \dfrac{d\xi}{dt} \end{pmatrix},$$

$$\text{(b)}\quad \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} x(\alpha) + \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} x(\beta) = \begin{pmatrix} e^\alpha \\ e^\beta \end{pmatrix}. \tag{5.10}$$

A fundamental solution is given by

$$F_0(t) = \begin{pmatrix} \displaystyle\int_\alpha^t e^{-20s^2}\, ds & 1 \\ e^{-20t^2} & 0 \end{pmatrix}. \tag{5.11}$$

For $t \ll 0$ we see that the first column is $\approx \binom{0}{0}$ whereas for $t \gg 0$ this column almost has the same direction as $\binom{1}{0}$. This indicates that some appropriate linear combination of the columns in $F_0(t)$ might give a better conditioned representation (in the sense of directional independence). Indeed computations reveal that we have a basis solution that does not increase and one that does not decrease, at least not significantly, and which are almost orthogonal at each point. We have computed the solutions on $[-1, 1]$ for several tolerances. First we give in Table 5.7 the infinity norms of the first and the second basis solution as they are found from using the decoupled recursion (cf. § 2). It is no surprise that we obtain stable results for the desired solution x. In Table 5.8 we give the maximal error in the computed result on $[-1, 1]$ at equally spaced output points for several tolerances. We remark that in this example the use of incremental

TABLE 5.7

t	$\|\text{first column of } \tilde{\Phi}(t)\|_\infty$	$\|\text{second column of } \tilde{\Phi}(t)\|_\infty$
−1.0	5.2 −9	1.0
− .8	7.0 −6	1.0
− .6	1.9 −3	1.0
− .4	1.0 −1	1.0
− .2	1.1	1.0
0	2.6	9.8 −1
.2	1.4	7.8 −1
.4	1.0	1.0 −1
.6	1.0	1.9 −3
.8	1.0	7.0 −6
1.0	1.0	5.2 −9

TABLE 5.8

tolerance	$\|\text{error}\|_\infty$	ρ	κ
1.0 −4	2.0 −6	1.6 +1	1.0
1.0 −6	2.0 −8	1.6 +1	1.0
1.0 −8	4.7 −10	1.6 +1	1.0

values M (the more "standard implementation") would not give a satisfactory distribution of output points for negative t (both basis solutions have low activity for large negative t!) and in this way one certainly would not "detect the turning point".

Example 5.6. Finally we would like to illustrate our remark in § 3.5 about smooth problems. Consider the ODE

$$\frac{dx}{dt} = \begin{pmatrix} 20 & 0 & 0 \\ 0 & 19 & 0 \\ 0 & 0 & -18 \end{pmatrix} x + \begin{pmatrix} -20 \\ -19 \\ 18 \end{pmatrix} \tag{5.12}$$

and $x(0)+x(\pi)=(2,2,2)^T$. Apparently $x(t)=(1,1,1)^T$. We computed this solution in four ways. First we asked for output at 10 equally spaced nodes, both with the choices $w_i(t_i)=0$ (cf. (3.7)) at each of the minor shooting points and with the choice of $w_i(t_i)$ as in (3.31). Then we did the same computations now with a prescribed amplification $M=1.0+3$. As a tolerance we had $1.0-3$. It is not surprising that the obtained accuracy was essentially on the order of the machine precision. The results

are given in Table 5.9 (the cpu time is in milliseconds). Since we have such a low tolerance, our efficiency strategy could not work optimally (note the doubling of grid points and shooting points in the second experiment as compared to the last one). On the other hand we see that in the last row the number of minor shooting points is equal to the number of major shooting points, indicating that for larger values of M there even may be more room for efficiency. Indeed, if we take $M = 1.0 + 6$ the cpu time drops to 20 milliseconds.

TABLE 5.9

OUTPUT required	$w(t_i)$	# grid pts.	# minor s.p.	# major s.p.	cpu time
10 equal pts.	(3.7)	130	30	10	108
10 equal pts.	(3.31)	76	21	10	70
$M = 1.0 + 3$	(3.7)	131	27	8	116
$M = 1.0 + 3$	(3.31)	38	8	8	33

REFERENCES

[1] U. ASCHER, J. CHRISTIANSEN AND R. D. RUSSELL, COLSYS—*a collocation code for boundary value problems*, in Lecture Notes in Computer Science 76, Springer-Verlag, Berlin, 1979, pp. 164–165.

[2] C. DEBOOR AND R. WEISS, SOLVEBLOCK: *A package for solving almost block diagonal linear systems*, ACM Trans. Math. Software, 6 (1980), pp. 80–87.

[3] S. D. CONTE, *The numerical solution of linear boundary value problems*, SIAM Rev., 8 (1966), pp. 309–321.

[4] P. DEUFLHARD, *A modified Newton method for the solution of ill-conditioned systems of nonlinear equations, with application to multiple shooting*, Numer. Math., 22 (1974), pp. 289–315.

[5] ———, *Recent advances in multiple shooting techniques*, in Computational Techniques for Ordinary Differential Equations, Academic Press, New York, London, 1980, Section 10, pp. 217–272.

[6] R. ENGLAND, *A program for the solution of boundary value problems for systems of ordinary differential equations*, Culham Lab., Abingdon, Techn. Rep. CLM-PDN 3/73, 1976.

[7] G. E. FORSYTHE, M. A. MALCOLM AND C. B. MOLER, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[8] H. B. KELLER, *Numerical solution of two point boundary value problems*, CBMS Regional Conference Series in Applied Mathematics 24, Society for Industrial and Applied Mathematics, Philadelphia, 1976.

[9] M. LENTINI AND V. PEREYRA, *An adaptive finite difference solver for nonlinear two-point boundary problems with mild boundary layers*, SIAM J. Numer. Anal., 14 (1977), pp. 91–111.

[10] R. M. M. MATTHEIJ, *Characterization of dominant and dominated solutions of linear recursions*, Numer. Math., 35 (1980), pp. 421–442.

[11] ———, *Stable computation of solutions of unstable linear initial value recursions*, BIT, 22 (1982), pp. 79–93. See also Report 8108, Mathematisch Instituut, Nijmegen, which is somewhat more elaborate.

[12] ———, *The conditioning of linear boundary value problems*, SIAM J. Numer. Anal., 19 (1982), pp. 963–978.

[13] ———, *Estimates for the errors in the solution of linear boundary value problems, due to perturbations*, Computing, 27 (1981), pp. 299–318.

[14] ———, *The stability of LU-decompositions of block tridiagonal matrices*, Rep. Dept. Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY, 12181, 1982. Math. Comp. to appear.

[15] R. M. M. MATTHEIJ AND G. W. M. STAARINK, *On optimal shooting intervals*, Report 8123, Mathematisch Instituut, Katholieke Universiteit, Nijmegen, the Netherlands, 1981.

[16] M. R. OSBORNE, *The stabilized march is stable*, SIAM J. Numer. Anal., 16 (1979), pp. 923–933.
[17] S. M. ROBERTS AND J. S. SHIPMAN, *Two Point Boundary Value Problems: Shooting Methods*, Elsevier, New York, 1972.
[18] M. R. SCOTT AND H. A. WATTS, *Computational solution of linear two point boundary value problems via orthonormalization*, SIAM J. Numer. Anal., 14 (1977), pp. 40–70.
[19] G. W. M. STAARINK AND R. M. M. MATTHEIJ, BOUNDPACK: *A package for solving boundary value problems*, Mathematisch Instituut, Katholieke Universiteit, Toernooiveld, Nijmegen, the Netherlands, 1982.
[20] J. R. STOER AND R. BULIRSCH, *Einführung in die Numerische Mathematik* II, HTB 114, Springer, Berlin, 1973.
[21] J. M. VARAH, *Alternate row and column elimination for solving certain linear systems*, SIAM J. Numer. Anal., 13 (1976), pp. 71–75.