A false transient solution method for equations occurring in solving two point boundary value problems by multiple shooting
Kramer, M.E.; Mattheij, R.M.M.

Published: 01/01/1990

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):
RANA 90-18
December 1990

A FALSE TRANSIENT SOLUTION
METHOD FOR EQUATIONS OCCURRING
IN SOLVING TWO POINT BOUNDARY
VALUE PROBLEMS BY MULTIPLE SHOOTING

by

M.E. Kramer
R.M.M. Mattheij
A false transient solution method for equations occurring in solving two point boundary value problems by multiple shooting.

by

M.E.Kramer
R.M.M. Mattheij

Eindhoven University of Technology
P.O.Box 513
5600 MB Eindhoven
The Netherlands
The solution space of well conditioned two point boundary value problems may have both rapid growing and rapid decaying modes. Due to this the nonlinear equations arising in multiple shooting can be sensitive with respect to changes in certain components of the shooting vector. Hence the convergence domain of Newton's method for these equations may be small. An alternative is presented where the nonlinear system after suitable preconditioning acts as a vector field of an ODE of which the sought solution is a stable rest point. We present a preconditioner and prove its suitability for exponential boundary value problems with separated boundary conditions.
§ 1 Introduction

Consider the two-point BVP

\[
\begin{align*}
\dot{y} &= h(x,y(x)) \quad a < x < b \\
g(y(a),y(b)) &= 0.
\end{align*}
\]

After discretization (which might be based on any method, like collocation, finite differences or multiple shooting) we obtain a set of nonlinear equations for the vector \( s \) of unknowns at a number of nodes

\[
f(s) = 0.
\]

It is well known that even well-conditioned BVP can have rapidly growing modes. Depending on the method the nonlinear system can be more or less sensitive to smaller changes in the components of \( s \). Such a sensitivity can be quite large, in particular, for multiple shooting; indeed there integration per subinterval is done either forward or backward, despite the fact that the underlying BVP may have modes that grow quite rapidly either way. This sensitivity of \( f(s) \) may cause divergence of Newton’s method outside of quite small domains already, even though the BVP is well conditioned. In this paper we like to address this problem by suggesting an alternative for Newton’s method. Although this may be employed in combination with other discretization methods as well we shall assume throughout that multiple shooting is used.

The inspiration for our method arises from steady state computations for dissipative systems; we describe it here in an ODE context (i.e. 1-D elliptic and parabolic equations). Consider the equation

\[
\begin{align*}
\frac{d^2 u}{dx^2} &= F(x,u,\frac{du}{dx}) \quad a \leq x \leq b \\
g(u(a),\frac{du}{dx}(a),u(b),\frac{du}{dx}(b)) &= 0.
\end{align*}
\]

Suppose this describes a steady state of the (parabolic type) PDE

\[
\begin{align*}
\frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} - F(x,u,\frac{du}{dx}) \quad , \quad t > 0 \, , \, a \leq x \leq b \\
g(u(t,a),\frac{\partial u}{\partial x}(t,a),u(t,b),\frac{\partial u}{\partial x}(t,b)) &= 0 \\
u(0,x) &= u_0(x)
\end{align*}
\]

Such systems often occur in chemical engineering (cf.[10]) and have lead to methods that just append a time derivative to (1.3) to compute a so called false transient, of which only the limit as \( t \to \infty \) is of interest. This method, which is also sometimes referred to as time stepping, has proven to be extremely helpful in solving difficult steady state of flow-combustion problems (cf.[16]). One should note that such parabolic problems often involve elliptic operators which are responsible for the stable curve tracking. For more general BVP (and the resulting discretizations) the operator does not necessarily have such a nice property; roughly speaking we may not assume that the Jacobian matrices have eigenvalues with negative real parts. This is why we suggest to precondition our (discretized) problem (1.2) by some matrix \( M(s) \) and try to solve the IVP
\[
\frac{ds}{dt} = M(s)f(s) \quad t > 0
\]

\[
s(0) = s_0.
\]

In [9] local convergence was proven, if the IVP is locally contractive, i.e.

\[
\exists \alpha > 0 \exists \beta > 0 \ \forall \xi_{1}, \xi_{2}, \gamma_{1} < \gamma_{2} \quad \forall \xi_{k} \in \mathbb{R} : < M(s)\xi_{1}, \xi_{2} > < -\alpha |\xi_{1}|^{\beta}.
\]

In this paper we shall derive a preconditioner \(M(s)\) satisfying (1.6) for exponentially dichotomic BVP's with separated boundary conditions. These problems inherently have some negative eigenvalues (relating to decreasing modes) and some positive eigenvalues (relating to growing modes). By requiring these somewhat strong properties we are able to show the effect of this analytically. Nevertheless, the method also works for larger classes (including ones with moderately growing modes) as some examples show. The preconditioner \(M(s)\) introduced in this paper is based on the idea, that after decoupling the growing and decaying modes, the increments of the former are monitored in the opposite direction (i.e. from right to left), thus creating a contracting ODE (1.5). A remarkable feature of this method is that it benefits from larger subintervals, since that gives the decaying and growing character of the modes more 'time' to develop. This is just the opposite to what one has to do, if Newton's method is not able to solve for the equations (1.2), because smaller subintervals decrease the sensitivity of \(f(s)\) with respect to the directions of the growing modes!

Our aim in solving (1.2) by this false transient method is to enlarge the convergence domain. The numerical tests we performed clearly show that convergence on coarser grids and for less good initial values is obtained. However, if Newton's method works it is cheaper, both in iterations and in the amount of evaluations of the function \(h(x,y)\), defining the BVP (1.1).

The paper is built up as follows. In the next two sections we give a brief survey of the theoretical concepts used in this paper. In §4 we derive the formula of the preconditioner \(M(s)\) for the nonlinear equations arising in shooting methods. Finally some numerical results are presented in section 5.
§2 BVP's and well-conditioning

An important concept when solving a BVP numerically is well-conditioning, cf.[8]. We first recall the definition of well-conditioning for the linear problem

\[(2.1a) \quad \dot{y} = A(x)y + q(x) \quad a \leq x \leq b \]
\[(2.1b) \quad B_x y(a) + B_{x'} y(b) = \beta \]

2.2 Definition

Let \( Y(x) \) denote a fundamental solution of the homogeneous part of (2.1) and \( G(x,t) \) the Green's function of (2.1). Define \( Q = B_x Y(a) + B_{x'} Y(b) \).

The BVP (2.1) is well-conditioned, if

\[(2.2a) \quad \| Y(x)Q^{-1} \|_{L_2} \leq \kappa_1 \quad \text{and} \quad \sup_t \left[ \int_a^b \| G(t,s) \|^2_{L_2} ds \right]^{1/2} \leq \kappa_2 \]

with \( \kappa_1 \) and \( \kappa_2 \) of moderate size.

Well-conditioning of BVP's is closely related to the concept of dichotomy, i.e. the solution space can be split into a subspace of growing and a subspace of decaying modes and the angle between both subspaces is bounded away from zero, see [8,13].

2.3 Definition

Let \( Y(x) \) denote a fundamental solution of the ODE (2.1a). The function \( Y(x) \) is dichotomic if there is an orthogonal projection \( P \) and constants \( (K,\lambda,\nu) \), with \( K \) of moderate size and \( \lambda,\nu \geq 0 \), such that

\[(2.3a) \quad \forall x \geq t : \| Y(x)PY^{-1}(t) \| \leq Ke^{-\lambda(x-t)}
\forall x \leq t : \| Y(x)(I-P)Y^{-1}(t) \| \leq Ke^{-\nu(t-x)} \]

The fundamental solution is called exponentially dichotomic if both \( \lambda \) and \( \nu \) are positive.

In [8] it is shown that the dichotomy constant \( K \) in (2.3a) is of the same order of magnitude as \( \max(\kappa_1,\kappa_2) \) and that vice versa for a dichotomic BVP, \( \kappa_2 \) is of order \( K\kappa_1 \).

The well-conditioning or stability concepts for initial value problems in fact state that there should be no (fast) growing modes, since they cannot be controlled by the initial value condition. From the above theorem it is clear that for boundary value problems growing modes are allowed, but that they have to be controlled by the end point conditions and that vice versa the decaying modes have to be controlled at the begin point. In [1] the following theorem is derived that gives some information on the relation between the dichotomy splitting and the boundary conditions.
2.4 Theorem ([1] Th.3.107)
Suppose BVP (2.1) is well-conditioned and has separated boundary conditions and let the fundamental solution $Y(x)$ be dichotomic with projection $P$. Then

(i) \[ \text{rank}(B_a) = \text{rank}(P) \]
(ii) \[ \text{null}(B_a) \cap R(Y(a)P) = \{0\} \]
(iii) \[ \text{null}(B_a) \cap R(Y(b)(I-P)) = \{0\} \]

\[ \blacklozenge \]

For nonlinear BVP's we can generalize this concept of well conditioning using a standard linearization argument. In order to make this meaningful one should require some uniform (dichotomy) bound for (linearized) problems in a suitable neighbourhood of $y(x)$. This induces the following definition.

2.5 Definition
The nonlinear BVP (1.1) is well-conditioned at an isolated solution $y^*(x)$, if

\[ (2.5a) \exists \varepsilon > 0 \exists \kappa(\varepsilon) \forall y \in C^2([a,b]-\{\bullet\}) : \max_{x \in [a,b]} \|y(x)-y^*(x)\| < \varepsilon \Rightarrow \text{the conditioning constants } \kappa_1 \text{ and } \kappa_2 \text{ of the linearized BVP at } y(x) \text{ are smaller than } \kappa(\varepsilon). \]

\[ \blacklozenge \]

If the linearized BVP at the solution $y^*(x)$ is exponentially dichotomic, then this is also true for the linearization at neighbouring functions $y(x)$ (see [13]). For BVP's with fast growing or decaying modes the conditioning number $\kappa$ can vary strongly with $\varepsilon$. 

4
§3 Shooting methods and the use of false transients

As remarked in §1 we shall consider multiple shooting as a means to solve the BVP (1.1). To recall, for this method one chooses \( N+1 \), say, shooting points \( x_i \in [a,b] \),

\[
a = x_1 < x_2 < \ldots < x_{N+1} = b.
\]

On each subinterval one has the following initial value problem

\[
y' = h(x,y(x)) \quad x_i < x < x_{i+1}, \quad 1 \leq i \leq N
\]

\[
y(x_i) = s_i
\]

The solution of the IVP on the \( i \)th interval is denoted by \( y_i(x;s_i) \). The concatenation of these solutions is the solution to the original BVP (2.1) by requiring that they form a continuous function satisfying the boundary conditions. So the initial vectors \( s_i \) have to be the solution of the following set of nonlinear equations

\[
f(s) = 0, \quad f \in C^1(\mathbb{R}^{(N+1)} \rightarrow \mathbb{R}^{(N+1)})
\]

and

\[
f(s) := \begin{pmatrix}
y_1(x_2;s_1) - s_2 \\
y_2(x_3;s_2) - s_3 \\
\vdots \\
y_N(x_{N+1};s_N) - s_{N+1} \\
g(s_1,s_{N+1})
\end{pmatrix}
\]

If the BVP is nonlinear, the function \( f(s) \) is nonlinear too and (3.2) has to be solved by some iterative scheme. Generally Newton’s method is used because of its favourable convergence rate. However, its convergence domain might be small.

As we mentioned in the previous section, the ODE part of (1.1) and thus its linearization may allow for both growing and decaying modes. In case of an exponential dichotomy there are exponentially growing solution modes and the initial value problems on the subintervals will not be well-conditioned (stable). Consequently the function \( f(s) \) is very sensitive to changes in \( s \) in some directions. This problem can be alleviated by choosing smaller subintervals, though the price for this is a system of higher dimension.

In order to solve (3.2) for a larger range of starting values we look at the false transient method

\[
\frac{ds}{dt} = M(s)f(s) \quad t > 0 \\
s(0) = s_0
\]

Of course the preconditioner has to be such that the solution \( s^* \) of (3.2) is a stable steady state of (3.4). In [9] it was found that local convergence occurs, if \( M(s) \) satisfies

\[
\exists_{\varepsilon > 0} \exists_{\alpha > 0} \forall_{x, y, r \in C} \forall_{\xi \in \mathbb{R}} : < M(s)\xi, \xi > < -\alpha |\xi|^2
\]
i.e. the linearized problem is a contraction. The quantity

\[(3.6) \quad \mu_2[A] := \max_{\xi \neq 0} \frac{\langle A\xi, \xi \rangle}{\langle \xi, \xi \rangle}\]

is called the *logarithmic norm*. In the next section we give an algorithm for forming a suitable preconditioner $M(s)$ for the nonlinear function (3.3), if the boundary value conditions are separated and the BVP (1.1) is exponentially dichotomic. In practice the same algorithm also works for BVP's with ordinary dichotomy, although the contractivity of (3.4) for this case is not immediate. A similar remark can be made for BVP with general two point conditions.

In the [9] an implicit integration method, called mixed Euler, for (3.4) was introduced. The iterates $\{x_i\}$ satisfy the relation

\[(3.7) \quad x_{i+1} = x_i + h_i M(x_i)f(x_{i+1})\]

Like Euler forward, this method uses only one evaluation of $M(s)$ and $J(s)$ per step, but its stability properties are comparable to those of implicit Euler. Furthermore, the method can be implemented in such a way that it converges to Newton's method if the preconditioner is $-J^{-1}(x)$. 

6
§4 Forming a Preconditioner

The function \( f(s) \) resulting from multiple shooting can be considerably more sensitive to changes in some directions, because both the growing and decaying modes are integrated forward. A first step towards balancing \( f \) is finding a splitting between those two kinds of modes. A suitable decoupling strategy is outlined in [1] Ch.4, 6. There the decoupling is used to solve the linear system \( J \xi = f \) in a stable way. However, the same ideas can also be exploited to balance \( f(s) \).

Consider a BVP with separated boundary conditions

\[
\begin{align*}
\frac{dy}{dx} &= h(x,y(x)) , \quad a < x < b \quad \text{and} \quad y \in C^1([a,b] \rightarrow \mathbb{R}^n) \\
g_1(y(b)) &= 0 \quad \text{and} \quad g_2(y(a)) = 0 , \\
\text{with} \quad g_1 : \mathbb{R}^n \rightarrow \mathbb{R}^r \quad \text{and} \quad g_2 : \mathbb{R}^n \rightarrow \mathbb{R}^s , \quad 1 \leq p \leq n
\end{align*}
\]  

that satisfies the following assumption.

4.2 Assumption
The BVP (4.1) is well-conditioned at its solution \( y^*(x) \) and the linearization at \( y^*(x) \) is exponentially dichotomic.

Let \( y(x) \) be an approximation of the solution of (4.1). If \( y(x) \) is a sufficiently good approximation, the linearized problem at \( y(x) \) will also be exponentially dichotomic (see [3] Ch.4), say with constants \((K,\lambda,v)\).

The linearization of (4.1) at \( y(x) \) will be denoted by

\[
\begin{align*}
\dot{z} &= A(x) , \quad a < x < b \\
B_a z(a) + B_b z(b) &= 0 \\
\text{with} \quad B_a &= \begin{pmatrix} 0 \\ B_{a2} \end{pmatrix} , \quad B_b = \begin{pmatrix} B_{b1} \\ 0 \end{pmatrix}^{1n-p}
\end{align*}
\]

The Jacobian \( J(s) \) of the nonlinear equation (3.2) is

\[
\begin{pmatrix}
\Psi_1(x_2) & \cdots & -I \\
\Psi_2(x_3) & \cdots & -I \\
\Psi_n(x_{N+1}) & \cdots & -I \\
B_o & \cdots & B_v
\end{pmatrix}
\]

where \( \Psi_i(x) \) is the fundamental solution of (4.3) with \( \Psi_i(x_0) = I \).

Let \( Y(x) \) be the fundamental solution of (4.3) with \( B_a Y(a) + B_b Y(b) = I \), then \( Y(x) \) is dichotomic with
projection $P = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$; i.e. the first $p$ columns of $Y(x)$ span the growing modes. Generally speaking it is expensive to obtain $Y(x)$. However, if $Z(x)$ is a fundamental solution with $B_{a2}Z(a) = (0 \mid \ast)$, then the matrix $H$, such that $Z(x) = Y(x)H$, has a zero left lower $(n-p)xp$ block; i.e. the first $p$ columns of $Z(x)$ and $Y(x)$ span the same space of increasing modes. So we choose the fundamental solution $\Phi_i(x)$ of (4.3) on the subintervals such that $\Phi_i(x_j) = Q_i$, with $Q_i$ is an orthogonal matrix satisfying

$$B_{a2}Q_i = \begin{pmatrix} 0 & B_{a2}^{(n)} \end{pmatrix}$$

and $Q_i$ the orthogonal matrix resulting from the QU-decomposition $\Phi_i(x_j) = Q_iU_{i-1}$. Then the first $p$ columns of $\Phi_i(x_j)$ span the increasing modes integrated up to $x = x_j$ and so do the first columns of $Q_j$. By an induction argument this holds for all $Q_i$. The fundamental matrices $\Phi_i$ can be introduced by differentiating $f(s)$ with respect to another argument. Define

$$Q := \text{diag} (Q_1, Q_2, \ldots, Q_{N-1}) \quad \text{and} \quad \dot{Q} := \text{diag} (Q_2, Q_3, \ldots, Q_{N-1}, I)$$

then

$$\frac{\partial f(s)}{\partial Q(s)} = \begin{pmatrix} Q_2U_1 & -Q_2 \\ Q_3U_2 & -Q_3 \\ \vdots & \vdots & \ddots & \vdots \\ B_{a2}Q_1 & -Q_{N-1}U_N & \ldots & \ldots \\ B_{a2}Q_1 & -Q_{N-1}U_N & \ldots & \ldots \\ B_{a2}Q_1 & -Q_{N-1}U_N & \ldots & \ldots \\ B_{a2}Q_1 & -Q_{N-1}U_N & \ldots & \ldots \\ \end{pmatrix} = \dot{Q} \cdot \begin{pmatrix} U_1 & -I \\ U_2 & -I \\ \vdots & \vdots & \ddots & \vdots \\ U_N & -I \\ \end{pmatrix}$$

So in fact at every subinterval-endpoint $x_{i+1}$ the fundamental solution is split into an orthogonal matrix $Q_{i+1}$ that contains information on the evolution of the directions of the various modes and an upper triangular matrix $U_{i}$ that contains information on the growth behaviour of those modes. This growth behaviour now is described by the dichotomy of the problem and so we can relate the magnitude of the elements of $U_{i}$ to the dichotomy constants. Let the upper triangular matrix $U_{i}$ be split into four blocks as in

$$U_{i} = \begin{pmatrix} B_{i} & C_{i} \\ 0 & E_{i} \end{pmatrix} \quad \text{with} \quad B_{i} \in \mathbb{R}^{(n-p)x(n-p)} \quad \text{and} \quad E_{i} \in \mathbb{R}^{xp}$$

For these matrices the following property is useful

4.8 Lemma ([1] Th. 6.15)

$$\forall i : \|B_{i}\| \leq K^{1/2} + 1 \cdot e^{-\Delta s_i}, \quad \text{and} \quad \|B_{i}^{-1}\| \leq K^{1/2} + 1 \cdot e^{-\Delta s_i},$$

with $\Delta s_i = x_{i+1} - x_i$
This shows that, as expected, both $\| E_i \|$ and $\| B_i^{-1} \|$ become small as we integrate over larger intervals. The part $C_i$ would be zero if the increasing and decreasing modes were orthogonal to each other. This would be an ideal case as it would give a complete decoupling between the two modes. However, by a local coordinate system transformation this situation can be reached. For this purpose we use a Ricatti-transformation. Note that this process is started from the end-point $x = b$.

From equation (4.6) one can see that, due to the zero structure of $B_bQ_i$ and $B_bQ_{N+1}$, non singularity of the Jacobian $J(s)$ implies non singularity of the matrices $B_i$, $E_i$ and the left upper block $B^{(i)}_b$ of $B_bQ_{N+1}$:

$$B_bQ_{N+1} = \begin{pmatrix} B_b^{(1)} & B_b^{(2)} \\ 0 & 0 \end{pmatrix}, \quad B_b^{(i)} \in \mathbb{R}^{(n-p) \times (n-p)}$$

In other words the endpoint conditions control the space spanned by the first columns of $Q_{N+1}$, i.e. almost the space of growing modes.

Now the Ricatti-matrices $R_i \in \mathbb{R}^{(n-p) \times p}$ are fully determined by

$$R_{N+1} = \left( B_b^{(i)} \right)^{-1} B_b^{(2)}$$

$$B_iR_i - C_i - R_{i+1}E_i = 0 \quad \text{for } i=N \text{ downto } 1$$

Consequently the endpoint boundary conditions are 'concentrated' in its the upper $(n-p) \times (n-p)$ block, viz.

$$B_bQ_{N+1} \begin{pmatrix} I & -R_{N+1} \\ 0 & I \end{pmatrix} = \begin{pmatrix} B_b^{(i)} & 0 \\ 0 & 0 \end{pmatrix}$$

The Ricatti transformation results in a decoupling of the growing and decaying modes, viz.

$$\begin{pmatrix} B_i & C_i \\ 0 & E_i \end{pmatrix} \begin{pmatrix} I & -R_i \\ 0 & I \end{pmatrix} = \begin{pmatrix} B_i - R_{i+1}E_i \\ 0 & E_i \end{pmatrix} = \begin{pmatrix} I & -R_{i+1} \\ 0 & I \end{pmatrix} \begin{pmatrix} B_i & 0 \\ 0 & E_i \end{pmatrix}$$

Again this can be interpreted as a change of the variable to which $f(s)$ is differentiated. Define the matrices

$$S_i = \begin{pmatrix} I & R_i \\ 0 & I \end{pmatrix}, \quad S_i \in \mathbb{R}^{n \times n}$$

and

$$S := \text{diag}(S_1, S_2, \ldots, S_{N+1}) \quad \text{and} \quad \hat{S} := \text{diag}(S_2, S_3, \ldots, S_{N+1}, I)$$

Then
We have set out to find a preconditioner \( M(s) \) for the Jacobian \( J(s) \) of \( f(s) \) which yields a negative logarithmic norm for \( M(s)J(s) \). The form of the Jacobian after the transformations we just performed suggests the use of the following theorem presented in [17].

4.16 Theorem

Let \( A \in \mathbb{R}^{m \times m} \) be partitioned into \( m^2 \) \( n \times n \)-blocks \( A_{ik} \) and let \( \tilde{A} \) be the \( m \times m \) matrix defined by

\[
\tilde{A}_{ik} = \begin{cases} \mu_2[A_{ik}] & i = k \\ |A_{ik}| & i \neq k \end{cases}
\]

Then

\[
\max_k \mu_2[A_{ik}] \leq \mu_2[A] \leq \mu_2[\tilde{A}]
\]

The fact that \( \mu_2[-I] = -1 \) can give us a clue to create a matrix with a negative logarithmic norm from the Jacobian. For this the norm of the non-identity blocks should be small. The \( E_i \) blocks have a small norm as they represent the growth of the decaying modes. However, the \( B_i \) blocks have a norm larger than 1. So we have to rescale the \( B_i \) and boundary condition blocks out of this matrix and perform a permutation to move the \(-I\) blocks to the diagonal.

Define

\[
\hat{B} = \text{diag}(-B_1, -B_2, \ldots, -B_N, B_b^{(1)}, -B_s^{(2)})
\]

and
This finally leads us to the desired form

\[
\frac{\partial f(s)}{\partial \log s} = \hat{Q} \cdot \hat{S}^{-1} \cdot \hat{B} \cdot \hat{P} \cdot \hat{J}
\]

where

\[
\hat{J} = \begin{pmatrix}
-1 & 0 & B_i^{-1} & 0 & \cdots & 0 & B_N^{-1} \\
-1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\
-1 & 0 & B_2^{-1} & 0 & \cdots & 0 & B_N^{-1} \\
E_1 & 0 & -1 & \cdots & \cdots & \cdots & \cdots \\
0 & -1 & 0 & B_3^{-1} & \cdots & \cdots & \cdots \\
E_2 & 0 & -1 & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots \\
-1 & 0 & B_N^{-1} & \cdots & \cdots & \cdots & \cdots \\
E_{N-1} & 0 & -1 & \cdots & \cdots & \cdots & \cdots \\
0 & -1 & 0 & \cdots & \cdots & \cdots & \cdots \\
E_N & 0 & -1 & \cdots & \cdots & \cdots & \cdots 
\end{pmatrix}
\]

4.21 Theorem

Let \(0 < \varepsilon < 1\) and suppose that the interval length \(x_{i+1} - x_i\) is sufficiently long so that

\[
\forall_i : \| E_i \|_2 < 1 - \varepsilon \quad \text{and} \quad \| B_i^{-1} \|_2 < 1 - \varepsilon
\]
And let \((K,\lambda,\nu)\) be the dichotomy constants of the linearized BVP at \(y(x)\). Then with

\[
(4.21b) \quad M = QS^TP^TB^{-1}SQ^T
\]

\[
(4.21c) \quad \mu_2[MJ] < \frac{-\varepsilon}{(K^2 + K + 1)}.
\]

\[\blacktriangle\]

**Proof**

The relationship \(\mu_2[\hat{J}] = \max \{ \lambda \mid \lambda \in \sigma((A^*A)/2) \}\) combined with Gershgorin’s circle theorem and theorem 4.16 gives

\[
\mu_2[\hat{J}] \leq \max_{1 \leq i \leq N+1} \left( \max\left( -1 + \frac{b_{i-1} + b_i}{2}, -1 + \frac{e_{i-1} + e_i}{2} \right) \right)
\]

with \(b_i := \begin{cases} |B^{-1}| & \text{if } 1 \leq i \leq N \\ 0 & \text{if } i=0 \text{ or } i=N+1 \end{cases}\)

and \(e_i := \begin{cases} |E_i| & \text{if } 1 \leq i \leq N \\ 0 & \text{if } i=0 \text{ or } i=N+1 \end{cases}\)

From this we obtain \(\mu_2[\hat{J}] < -\varepsilon\). Now with \(V := SQ^T\) and \(J_y := \frac{\partial f(s)}{\partial V}\)

\[-\varepsilon > \mu_2[\hat{J}]
\]

\[= \mu_2[V^{-T}MJ_y]
\]

\[= \frac{1}{2} \max \left\{ \frac{\langle (V^{-T}MJ_y + J_y^TM^{-T}V^{-1})x, x \rangle}{\langle x, x \rangle} \mid x \neq 0 \right\}
\]

\[= \frac{1}{2} \max \left\{ \frac{\langle [MJ_yV + V^TJ_yM]V^{-1}x, V^{-1}x \rangle}{\langle V^{-1}x, V^{-1}x \rangle} \cdot \frac{\langle V^{-1}x, V^{-1}x \rangle}{\langle x, x \rangle} \mid x \neq 0 \right\}
\]

\[\geq \mu_2[MJ_yV] \cdot |V^{-1}|^2
\]

\[= \mu_2[MJ] \cdot |V^{-1}|^2
\]

One can easily prove that \(\|R_j\|_2 < K\) and thus \(\|V^{-1}\|_2^2 < (K^2 + K + 1)\), yielding

\[\mu_2[MJ] < \frac{-\varepsilon}{(K^2 + K + 1)}
\]

\[\blacktriangle\]

We see from Lemma 4.8 that the contractivity of the IVP

\[
(4.22) \quad \frac{ds}{dt} = M(s)f(s) \quad t > 0
\]

\[s(0) = s_0
\]

12
increases if the shooting intervals are enlarged. However, the IVP-integrator generally still bounds the length of the subintervals. Also, since we take $\| M(s) \|_2$ as an estimate for the error in $s$, it is important that $\| M^{-1} \|_2$ is not too large, for the main term therein is $\| \hat{B} \|_2 \geq \frac{e^{\nu_{\max}}}{K\sqrt{K^2+1}}$. 

If a BVP has both exponentially fast growing and decaying modes, but does not have separated boundary conditions, the algorithm derived in this section cannot be applied directly. By adding a few variables the system can be transformed to one with separated boundary conditions (see [2]). However, the additional variables introduce constant modes, so that the theory does not hold. Nevertheless application of the algorithm may still be worthwhile, because one can not expect "redundant" BC to have an impact on properly controlling modes (which is essential in our decoupling analysis).

When applying the mixed Euler method to the IVP (4.22), the calculation of the iterates requires less work if the norm of $M(s)$ is small (see [9]). From (4.21b) it follows that

$$\begin{align*}
\| M(s) \|_2 &\leq \| S^T \|_2 \| \hat{B} \|_2 \| S \|_2 \\
&\leq (K^2+K+1) \cdot \max(1, \| (B_s^{(0)})^{-1} \|_2, \| (B_s^{(0)})^{-1} \|_2) \\
&= O(1)
\end{align*}$$

if the boundary conditions are properly scaled.
§5 Numerical results

The idea of preconditioned time stepping with the preconditioner presented in the previous section is implemented in a code called TS. We compare this code with two other multiple shooting codes, viz. MUSN and RWPM (see [1,12], [6] resp.); both use variants of Newton's method to solve the nonlinear equations. The results, presented below, indicate that the time stepping algorithm can increase the convergence domain, sometimes even on problems that, though well-conditioned, do not satisfy the conditions of Th.4.21.

In the TS-program the required tolerance for the solution is denoted by TOL. The convergence criterion used is $\| M(x_j) f(x_j) \|_2 < TOL$ or $\| M(x_i) f(x_i) \|_2 < TOL$, where $f(x)$ must be evaluated with an accuracy smaller than TOL. The program employs the preconditioner $M(x)$ defined in the previous section and uses the mixed Euler method

\[(5.1) \quad x_{i+1} = x_i + h_i M(x_i) f(x_i)\]

(see [9]) to proceed through time. The discretization error hereof is bounded by the user prescribed tolerances ATOL and RTOL for the absolute and relative error respectively. Based on these tolerances the TS-program determines the stepsize $h_i$. The iterate $x_{i+1}$ is obtained by modified Newton's method using the Jacobian at $x_i$ only and not at any intermediate point. If this process does not yield an approximation to $x_{i+1}$ with an error less than $\min(1d-2, \text{ATOL} + \text{RTOL} \| x_i \|)$ within three iterations, the stepsize $h_i$ is halved until an adequate approximation of $x_{i+1}$ is found. For more details see [9].

The local IVP's on the subintervals are integrated using RKF45 as implemented in MUSN. Two parameters ER and DAMP are used to control this process. During the RKF45 integration we require the discretization error to be less than $\text{ER}(1+\| x_i \|)$; i.e. ER is a combined absolute and relative tolerance. Of course this tolerance has to be less than the required tolerance TOL for the solution of the BVP at the end of the time stepping process. However, if the vector $s$ is still far from the solution of (3.2) a small value of ER will require more work without increasing the convergence speed considerably. Hence the user has to give an initial value for ER and during the process ER is taken as the minimum of its previous value and the norm of the residual $M(x_i) f(x_i)$ divided by a factor DAMP.

A good indication for the computational costs of BVP-solving algorithms is the number of evaluations of the function defining the field of directions of the BVP ($h(x,y)$ in (4.1)). In the tables in this section this quantity is denoted by $\# f^{\text{func}}$-calls (N.B. this is not equal to the number of times $f(x)$ is computed).

5.2 Example

Consider the problem attributed to Troesch [18]

\[(5.3) \quad \ddot{z}(x) = \lambda \sinh(\lambda z) \quad 0 < x < 1\]
\[z(0) = 0\]
\[z(1) = 1\]

This has been used as a test problem by many authors (e.g. [4,15]). The linearisation of this problem at its exact solution is exponentially dichotomic with the growth factors $\lambda e^{\lambda}$ and $-\lambda e^{\lambda}$. Due to this, forward integration becomes inaccurate over longer subintervals and the nonlinear function
$f(s)$ is very sensitive to small changes of the starting vector $s_i$ in the direction of the growing mode; in fact the local IVP's are ill-posed.

We look at the effect of choosing too large initial values $s_i$ and uniform (i.e. non-optimal) sub-intervals for rather small values of $\lambda$ ($\lambda \leq 5$). For the parameters we choose $\text{ATOL} = \text{RTOL} = 10^{-1}$, $\text{ER} = 10^{3}$, $\text{DAMP} = 10^{1}$ and set the required tolerance $\text{TOL} = 10^{-6}$. The initial guess to the solution is chosen to be

$$
\begin{align*}
    z(x) &= x \\
    \dot{z}(x) &= 1
\end{align*}
$$

The results (see table 5.1) clearly show that if the Newton's method works it requires less iterations and function calls than time stepping, as has to be expected. However, the time stepping algorithm can solve the problem for coarser grids, i.e. for more difficult cases.

For all choices of $\lambda$ the upper triangular matrices $U_i$ (see (4.7)) satisfy the condition that $\|B_i^{-1}\| < 1$ and $\|E_i\| < 1$, and coarser grids gave smaller values, i.e. the IVP (3.5) is stronger attractive. This does not appear from the number of required iterations, because for coarser grids the initial value of $\|M(x)f(x)\|$ is larger and the stepsize $h_i$ increases slower, since the Newton process to solve (5.1) requires a somewhat more careful treatment.

However, it should be clear that once the time stepping method has reached a reasonably small residual, one should switch to full Newton in practice; this would make the complexity for the combined method lower (on top of its, more important, better convergence behaviour).

### Table 5.1

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>subint</th>
<th>iter</th>
<th>result</th>
<th>steps</th>
<th>#fincalls</th>
<th>result</th>
<th>#fincalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>fail</td>
<td>3,350</td>
<td>21</td>
<td>conv</td>
<td>3,228</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>3</td>
<td>conv</td>
<td>2,923</td>
<td>18</td>
<td>conv</td>
<td>5,027</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>fail</td>
<td>16,955</td>
<td>23</td>
<td>conv</td>
<td>10,314</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>4</td>
<td>conv</td>
<td>6,710</td>
<td>22</td>
<td>conv</td>
<td>13,684</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>11</td>
<td>fail</td>
<td>38,625</td>
<td>32</td>
<td>conv</td>
<td>35,482</td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td>6</td>
<td>conv</td>
<td>12,978</td>
<td>30</td>
<td>conv</td>
<td>36,831</td>
</tr>
<tr>
<td>5</td>
<td>15(17)'</td>
<td>1</td>
<td>exp.overflow</td>
<td>52</td>
<td>conv</td>
<td>73,002</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>20(22)'</td>
<td>11</td>
<td>fail</td>
<td>80,787</td>
<td>50</td>
<td>conv</td>
<td>83,000</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>11</td>
<td>conv</td>
<td>55,875</td>
<td>51</td>
<td>conv</td>
<td>93,050</td>
</tr>
</tbody>
</table>

* The RKF45 integrator added two shooting points near $x = 1$, since the increase over the subintervals exceeded $10^9$
5.5 Example
The next example can be found in [15].

\[ \begin{align*}
\dot{y}_1 &= a \frac{y_1 (y_3 - y_1)}{y_2} \\
\dot{y}_2 &= -a (y_3 - y_1) \\
y_3 &= \frac{1}{y_4} [ b - c (y_3 - y_2) - ay_3 (y_3 - y_1) ] \\
\dot{y}_4 &= a (y_3 - y_1) \\
\dot{y}_5 &= -\frac{c}{d} (y_3 - y_5)
\end{align*} \] (5.6a)

with boundary conditions

\[ \begin{align*}
y(0) &= y_3(0) = y_s(0) = 0 \\
y_4(0) &= -10 \\
y_5(1) &= y_5(1)
\end{align*} \] (5.6b)

and constants

\[ \begin{align*}
a &= 100 ; \\
b &= 0.9 ; \\
c &= 1000 ; \\
d &= 10
\end{align*} \] (5.6c)

This problem does not satisfy the assumptions of exponential dichotomy; since \( y_2 \) and \( y_4 \) are linearly dependent, there is one constant mode. Computation of the eigenvalues of the linearized problem at the solution (as tabulated in [15]) yields one eigenvalue with (large) positive real part, three with (large) negative real part and one equal to zero. Indeed, the matrices \( E_i \) do have 1 or 2 elements larger than 1. Nevertheless the time stepping algorithm converges for the initial guess

\[ \begin{align*}
y_1 &= y_2 = y_3 = y_5 = 1 \\
y_4 &= -10
\end{align*} \] (5.7)

Due to the stiffness of the problem in case of forward integration, the parameter \( ER \) has been chosen small \((10^5)\) to prevent overflow.

Example 5.8
The following problem has been proposed in [7] and describes the flow between two rotating discs

\[ \begin{align*}
\dot{y}_1 &= y_2 \\
\dot{y}_2 &= y_3 \\
\dot{y}_3 &= \frac{(3-n)}{2} y_1 y_3 - ny_2^2 + 1 - y_4^2 + sy_2 \quad 0 < x < \infty \\
\dot{y}_4 &= y_5 \\
\dot{y}_5 &= \frac{(3-n)}{2} y_1 y_5 - (n-1) y_2 y_4 + s (y_4 - 1)
\end{align*} \] (5.9)

with boundary conditions

\[ \begin{align*}
y_1(0) &= y_4(0) = y_6(0) = 0 \quad , \quad y_2(\infty) = 0 \quad , \quad y_4(\infty) = 1
\end{align*} \] (5.10)
In practice a (large) value $L$ is taken as endpoint of the interval. Both in [14] and [4] (5.9) is used as a test problem with the parameter set $n = -0.1$, $s = 0.2$. In [14] $L = 11.3$ is the largest endpoint for which convergence is reached using continuation in $L$. The algorithm proposed in [4] can solve the BVP by continuation in $L$ for $L \leq 15$ with forward shooting and $L \leq 132$ with backward shooting. Forward shooting encounters a matrix with condition number $10^{12}$, because the growing and decaying modes are treated equally. Incidentally, using as initial guess

\begin{align*}
    y_1 &= -x^2 e^{-x} ; \quad y_2 = \dot{y}_1 ; \quad y_3 = \dot{y}_2 \\
    y_4 &= 1 - e^{-x} ; \quad y_5 = \dot{y}_4
\end{align*}

the codes tested here do not encounter this problem. As the solution mainly shows activity near its initial point, we choose a grid which is basically uniform, only the first subinterval is halved, yielding a grid of the form

\[
\text{We use three different codes to solve this problem, viz. MUSN, the TS-code and RWPM and look for the coarsest grid on which a solution was obtained with accuracy $10^{-6}$. For the TS-code the parameters were set to ATOL = RTOL = $10^{-6}$, $h_0 = 10^{-1}$, ER = $10^{-1}$ and DAMP = $10^0$. Although the linearized problem has three eigenvalues with negative real part, the rapid rotation of two decaying modes caused $\| E_j \|$ to exceed 1 on more than half the subintervals. Nevertheless convergence was reached quite easily (in about 20 to 30 steps) even on coarser grids, than either of the two other codes could handle.}

\[
\begin{array}{|c|c|c|c|}
\hline
L & \text{Least number of subintervals required} \\
\hline
& \text{MUSN} & \text{RWPM} & \text{TS} \\
\hline
12 & 13 & 7 & 6 \\
15 & 20 & 9 & 8 \\
20 & 27 & 12 & 10 \\
30 & 39 & 18 & 14 \\
132 & 169 & 73 & 58 \\
\hline
\end{array}
\]

Table 5.2

An even more interesting picture occurs if we plot the amount of BVP-evaluations versus the number of gridpoints for a fixed value of $L$ ($L=15$ in figure 5.1). Even though the TS-code used is not optimal (it does not switch to the Newton method near the convergence point) it performed cheaper than RWPM for coarse grids. This is due to the fact that for those grids over 90% of the iterations in RWPM are damped Newton steps with damping factor between $10^4$ and $10^7$. For finer grids the Newton algorithm in RWPM speeds up considerably, whereas the TS-code does not require essentially less steps, but each step is somewhat more expensive, because the integration interval is longer. This illustrates that the time stepping algorithm does not only serve its purpose of enlarging the convergence domain, but can occasionally even reduce the computational costs.

Several other authors [11,15] solved the BVP for $n = 0.2$, $s = 0.2$ and $L = 60$ or 200. For these parameter values the time stepping algorithm also worked well.
Next we want to see the effect of using different values of ER and DAMP. To this end we solve the BVP's (5.9) with \( L = 11.3, n = -0.1, s = 0.2 \) and (5.3) with \( \lambda = 4 \) with the time stepping algorithm for various values of ER and DAMP with 10 equidistant shooting points, required tolerance \( TOL = 10^{-6} \), initial stepsize \( h_0 = 0.1 \), initial guess (5.11), (5.4) resp. and \( ATOL = RTOL = 0.1 \). The results are shown in Table 5.3 and 5.4 respectively.

For both test problems the number of time steps did not vary significantly for different values of ER and DAMP. For larger ER the norm of the residue \( M(x_i)\|f(x_i)\| \) reduces just a little more slowly. More important is the effect that for larger values of DAMP the process speeds up considerably at the last few steps (i.e. \( \| M(x_i)\|f(x_i)\| \) decreases essentially faster). Apparently a better approximation of the update is only of great influence close to the solution.

The value of ER has considerable influence on the amount of \( f_{\text{eval}} \)-calls, since a smaller initial value of ER means a more accurate evaluation of \( f(s) \) and \( J(s) \).

In order to decrease the amount of work, one should try to minimize the number of \( f_{\text{eval}} \)-evaluations and hence choose a large ER. However, this does harbour the danger of divergence of the process, especially for sensitive problems.
Time stepping algorithm for (5.9) with \( n = -0.1 \); \( s = 0.2 \); \( L = 11.3 \)

<table>
<thead>
<tr>
<th>ER</th>
<th>DAMP</th>
<th>steps</th>
<th>( # \text{f} \text{m}-\text{calls} )</th>
<th>( | R_i |_{\text{max}} )</th>
<th>( \max(\text{diag}(B_i)) )</th>
<th>( \max(\text{diag}(E_j)) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^1 )</td>
<td>( 10^1 )</td>
<td>21</td>
<td>26,350</td>
<td>[1.9, 2.9]</td>
<td>[0.6, 0.9]</td>
<td>[1.7, 2.2]</td>
</tr>
<tr>
<td>( 10^3 )</td>
<td>( 10^2 )</td>
<td>21</td>
<td>28,728</td>
<td>[1.9, 2.9]</td>
<td>[0.6, 0.9]</td>
<td>[1.7, 2.2]</td>
</tr>
<tr>
<td>( 10^3 )</td>
<td>( 10^3 )</td>
<td>20</td>
<td>35,457</td>
<td>[1.9, 2.9]</td>
<td>[0.6, 0.9]</td>
<td>[1.7, 2.2]</td>
</tr>
<tr>
<td>( 10^1 )</td>
<td>( 10^3 )</td>
<td>19</td>
<td>26,965</td>
<td>[1.9, 2.9]</td>
<td>[0.6, 0.9]</td>
<td>[1.7, 2.2]</td>
</tr>
<tr>
<td>( 10^3 )</td>
<td>( 10^3 )</td>
<td>19</td>
<td>28,784</td>
<td>[1.9, 2.9]</td>
<td>[0.6, 0.9]</td>
<td>[1.7, 2.2]</td>
</tr>
<tr>
<td>( 10^3 )</td>
<td>( 10^3 )</td>
<td>19</td>
<td>36,527</td>
<td>[1.9, 2.9]</td>
<td>[0.6, 0.9]</td>
<td>[1.7, 2.2]</td>
</tr>
</tbody>
</table>

Table 5.3

Additionally we tabulate the range of the maximum values of the diagonal elements of \( E_j \) and \( B_i^{-1} \) at the various steps, these are indicative for \( E_j \) and \( B_i^{-1} \) resp.. The fifth column shows the development of the maximum of \( R_j \) during the process. This illustrates quite clearly that the decoupling of the growing modes is much better for Holt’s problem than for the Troesch’ problem. However, at the latter the problem the conditions of Th.4.21 are satisfied, whereas that is not the case for Holt’s problem.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Author(s)</th>
<th>Title</th>
</tr>
</thead>
</table>

[16] M.D.Smooke, J.A.Miller, R.J.Kee, Solution of premixed and counterflow diffusion flame
problems by adaptive boundary value methods. In: Numerical boundary value


[18] B.Troesch, A simple approach to a sensitive two-point boundary value problem. J.
<table>
<thead>
<tr>
<th>Number</th>
<th>Author(s)</th>
<th>Title</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>90-13</td>
<td>A. Reusken</td>
<td>Multigrid applied to mixed finite element schemes for current continuity equations</td>
<td>November '90</td>
</tr>
<tr>
<td>90-14</td>
<td>J. de Graaf</td>
<td>Riesz bases of special polynomials in weighted Sobolev spaces of analytic functions</td>
<td>December '90</td>
</tr>
<tr>
<td>90-15</td>
<td>J. Boersma, M.L. Glasser</td>
<td>Asymptotic expansion of a class of Fermi-Dirac integrals</td>
<td>December '90</td>
</tr>
<tr>
<td>90-16</td>
<td>J. Boersma</td>
<td>Note on the singularity exponents for complementary sectors</td>
<td>December '90</td>
</tr>
<tr>
<td>90-17</td>
<td>J. Boersma</td>
<td>Uniform asymptotics of a Bessel-function series occurring in a transmission-line problem</td>
<td>December '90</td>
</tr>
<tr>
<td>90-18</td>
<td>M.E. Kramer, R.M.M. Mattheij</td>
<td>A false transient solution method for equations occurring in solving two point boundary value problems by multiple shooting</td>
<td>December '90</td>
</tr>
<tr>
<td>91-01</td>
<td>P.G. Vroegindeweij</td>
<td>Algebraic preliminaries for field theories in space-time algebra</td>
<td>January '91</td>
</tr>
<tr>
<td>91-02</td>
<td>A. Reusken</td>
<td>Multigrid applied to two-dimensional exponential fitting for drift-diffusion models</td>
<td>January '91</td>
</tr>
</tbody>
</table>