A sequential shooting technique for singular perturbation problems
Staarink, G.W.M.; Mattheij, R.M.M.

Published: 01/01/1985

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):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 20. Jan. 2019
report no. WD 85-05

A Sequential Shooting Technique for
Singular Perturbation Problems

G.W.M. Staarink, R.M.M. Mattheij

december 1985

project 830902
Implementing a Sequential Shooting Technique for Singular Perturbation Problems

G.W.M. Staarink †

R.M.M. Mattheij ‡

† Wiskundige Dienstverlening
Katholieke Universiteit Nijmegen
Toernooiveld
6525 ED Nijmegen

‡ Mathematisch Instituut
Katholieke Universiteit Nijmegen
Toernooiveld
6525 ED Nijmegen
§ 1. Introduction

Consider the ODE

\[
\frac{dx}{dt} = Lx + r, \quad \alpha \leq t \leq \beta,
\]

where \( L(t) \) is an \( n \times n \) matrix function and \( r(t) \) an \( n \) vector function. Let a boundary condition \((BC)\) be given by:

\[
M_\alpha x(\alpha) + M_\beta x(\beta) = b.
\]

We assume that the homogeneous part of (1.1) has solutions, belonging to various groups of activity, at least there should exist one subspace of solutions that have kinematic eigenvalues absolutely large compared to the ratio \( \|\dot{x}(t)\|/\|x(t)\| \) of the solution outside special regions, the so called layers. It is most helpful to think in terms of singular perturbed ODE:

\[
\begin{align*}
\frac{dx_1}{dt} &= \frac{1}{\varepsilon} \tilde{L}^{11} x_1 + \frac{1}{\varepsilon} \tilde{L}^{12} x_2 + r_1, \quad \tilde{L}^{11} k \times k \\
\frac{dx_2}{dt} &= \tilde{L}^{21} x_1 + \tilde{L}^{22} x_2 + r_2.
\end{align*}
\]

Here \( \varepsilon \) is supposed to be a small parameter. Apparently the system in (1.3) has a Lipschitz constant of the order \( \max\left\{ \frac{1}{\varepsilon} \|\tilde{L}^{11}\|, \|\tilde{L}^{22}\| \right\} \).

If \( \|\tilde{L}^{ij}\| \) is small compared to \( \frac{1}{\varepsilon} \) and \( \tilde{L}^{11} \) is nonsingular, we may say that (1.3) has solutions of two time scales: "fast" solutions with \( \|\dot{x}(t)\|/\|x(t)\| = O\left(\|\tilde{L}^{11}\|\right) \) and "slow" solutions with \( \|\dot{x}(t)\|/\|x(t)\| = O\left(\|\tilde{L}^{22}\|\right) \). Of course both a "fast" and a "slow" mode might be either increasing or decreasing (a "slow" mode even none of the two). Now if we assume that \( \|\varepsilon r_1\| \) is not large compared to \( \|\tilde{L}^{11}\| \) we therefore should expect a solution to have layers of thickness \( O(\varepsilon) \) at \( t=\alpha \) and \( t=\beta \), where \( \|x_1\| = O\left(\frac{1}{\varepsilon}\right) \) (see fig. (1.1)) and a smooth behaviour in between. Of course the general ODE (1.1) allows more complicated situations.
In general the presence of fast (increasing) modes makes numerical initial value integration over (even smaller) intervals unattractive. Nevertheless this type of integration lies at the heart of the method we like to use: repeated superposition or multiple shooting. The reasons we like to use this method are threefold. First, forward integrations allows for simple stepcontrol (though this will not work satisfactory if we do not take appropriate measures). Second, we can fully employ the linear structure by using the simple but effective superposition technique as was developed in [6]. Third, and probably most important, the "marching" strategy allows for an option to track the layers and their thickness, a fact that is of course much related to the adaptivity of the integration. In [5] it was considered how to perform an adequate integration. In short the problems are twofold. In the first place as we already remarked an integrator is tempted to approximate the fast mode component in a random solution to the required accuracy and this is of little value in the smooth region. In the second place, even if we would have a control that would give stepsizes commensurating with the smoothness of $x$ we may run into the problem that the character of the incremental values does not correspond to that of the actual (continuous) modes; that is we need an in(de)creasing mode to be approximated by an in(de)creasing discrete mode (cf. [1]). Only if we use a so called di-stable method this property (essential to preserve stability and well-conditioning) is present.

In our discussion we shall use information about the eigenvalues of $L(t)$. Although it is known (cf. [7]) that in general these eigenvalues may be quite meaningless to derive information about the growth of various modes and hence of the dichotomy of the system, they are indicative for the fast modes, if their rotation is not fast. We shall denote the positive real valued ones by $\mu_1, \ldots, \mu_k$ ($Re(\mu_1) \geq \ldots \geq Re(\mu_k)$) and the negative ones by $\lambda_1, \ldots, \lambda_l$ ($Re(\lambda_1) \leq \ldots \leq Re(\lambda_l)$). Our aim is to attain the solution $x$ within an
accuracy TOL say.

The paper is built up as follows: First we review multiple shooting and a special
decoupling algorithm, that is an essential part of our method (§ 2). Next we treat a di-
stable integrator, SYMIRK, in § 3. The problem of how to obtain an initial value of a
smooth solution is dealt with in § 4. The actual global strategy then is considered in §
5. Finally we give a simple but instructive example in § 6.

§ 2. Multiple shooting and upper triangular recursions.

In order to understand the importance of the subject dealt with in later sections,
we first outline the basic idea of multiple shooting and the special variant that is em-
ployed here.

For simplicity, let us assume that a set of shooting points \( \{ t_i \}_{i=1}^{N+1} \) is given, which
split the interval \([\alpha, \beta]\) into \(N\) subintervals \([t_i, t_{i+1}]\), \(i=1, \ldots, N\). On each subinterval
\([t_i, t_{i+1}]\) we compute a particular solution \(p_i(t)\) and a fundamental solution \(F_i(t)\) (we
disregard the fact that we actually compute these solutions on a certain grid only and
moreover only "approximately", see later). These fundamental solutions are chosen
such that

\[
F_i(t_{i+1}) = F_{i+1}(t_{i+1})U_{i+1} = Q_{i+1}U_{i+1},
\]

where \(Q_{i+1}\) is an orthogonal matrix and \(U_{i+1}\) an upper triangular matrix (this can e.g.
be done by using Householder's method). From the linearity it then follows that there
exists vectors \(a_i\) such that

\[
x(t) = F_i(t)a_i + p_i(t) \quad , t \in [t_i, t_{i+1}].
\]

By continuity of \(x\) we can find a recursive relation between these \(a_i\), via

\[
x(t_{i+1}) = F_i(t_{i+1})a_i + p_i(t_{i+1})
\]

\[
= F_{i+1}(t_{i+1})U_{i+1}a_i + p_i(t_{i+1})
\]

\[
= F_{i+1}(t_{i+1})a_{i+1} + p_{i+1}(t_{i+1}) ,
\]
so, using (2.1)

\[(2.4a) \quad a_{i+1} = U_{i+1}a_i + d_{i+1},\]

where

\[(2.4b) \quad d_{i+1} = Q_i^{-1}[p_i(t_{i+1}) - p_{i+1}(t_{i+1})].\]

If the problem is \textit{well-conditioned}, then the underlying ODE (1.1) is \textit{dichotomic}, i.e. there is a splitting in non decreasing and non increasing solutions (cf. [3]), the dimension of the subspace of former solutions being \(k\) say. If the initial value \(Q_1 = F_1(t_1) = F_1(\alpha)\) is chosen appropriately (a very generally occurring fact, see [6]), and the discretization preserves growth properties (\textit{di-stability}, cf [1]), then the decoupling in (2.4a) leads to a natural partitioning:

\[(2.5a) \quad a_{i+1}^1 = B_{i+1}a_i^1 + C_{i+1}a_i^2 + d_{i+1}^1,\]

\[(2.5b) \quad a_{i+1}^2 = E_{i+1}a_i^2 + d_{i+1}^2.\]

Here \(B_{i+1}\) is a \(k^{th}\) order square matrix. Because of the dichotomy, (2.5b) is stable for increasing index and (2.5a) is stable for decreasing index.

We now have a stable way to compute the \(x(t_i)\). First we remark that the \(BC\) (1.2) induce a \(BC\) for \(\{a_i\}_{i=1}^{N+1}\). Next we realize that we can safely use superposition for the discrete system, i.e. we set

\[(2.6) \quad a_i = \Phi_i c + z_i,\]

where \(\{\Phi_i\}\) is a fundamental solution of (2.4a), \(\{z_i\}\) a particular solution of (2.4a) and \(c\) some fixed vector. For stability reasons we let \(\{\Phi_i\}\) satisfy the separated \(BC:\)

\[(2.7) \quad \Phi_i^1 = (\emptyset | I) ; \Phi_{N+1}^1 = (I | \emptyset)\]

and \(\{z_i\}\)
Given \( \{ \Phi_i \} \) and \( \{ z_i \} \) we can find \( c \) from the relation, cf (1.2)

\[
(2.9) \quad [M_\alpha Q_1 \Phi_1 + M_\beta Q_{N+1} \Phi_{N+1}]c = b - M_\alpha p_\alpha(\alpha) - M_\beta p_{N+1}(\beta) - M_\alpha Q_1 z_1 - M_\beta Q_{N+1} z_{N+1}.
\]

As was shown in [3] the condition numbers of the problem, viz. indicating the sensitivity of the solution with respect to the \( BC \) and a bound for the Green's function respectively, are easily estimated by quantities in this algorithm. The first one by the norm of \([M_\alpha Q_1 \Phi_1 + M_\beta Q_{N+1} \Phi_{N+1}]^{-1}\) and the second one by suitable monitoring of diagonal elements occurring in \( B_i \) and \( E_i \) (see [6]).

§ 3. **A special distable integrator.**

As remarked before it is essential for stability that the integrator be dichotomically stable, cf [1] (although we realize that a more general notion is in fact necessary when dealing with non autonomous problems). The integrator we use is described in [2] and is the fourth order member of a two stage off step derivative family. It turns out to be identical with some long known Runge-Kutta methods. Typically the two-stage scheme, with \( x_{j+1} \) as approximation to \( x(t_{j+1}) \), reads:

\[
(3.1a) \quad x_{j+\nu_2} = \frac{1}{2}(x_{j+1} + x_j) + \frac{1}{8} h_j(\dot{x}_{j+1} - \dot{x}_j)
\]

\[
(3.1b) \quad x_{j+1} = x_j + h_f(\frac{1}{6} \dot{x}_j + \frac{2}{3} \dot{x}_{j+\nu_2} + \frac{1}{6} \dot{x}_{j+1}),
\]

where \( h_f \) is the stepsize, i.e. \( t_{j+1} - t_j \), and \( \dot{x}_j \) denotes the righthandside of (1.1). We shall not go into details of how this method is used in a variable step code, called SYMIRK, but only remark that the implementation require some special and novel aspects, cf [2]; in particular since one has to design the stepsize selection mechanism such that it takes larger steps for \( |h_\lambda| \) large for it is tuned for model problems \( \dot{x} = \lambda x \), \( \lambda \) scalar. It is easily seen that the increment function for (3.1) looks like
(3.2) \[ \text{Incr}(z) = \frac{1 + \frac{1}{2}z + \frac{1}{12}z^2}{1 - \frac{1}{2}z + \frac{1}{12}z^2} , \]

where \( z = h\lambda \in \mathbb{C} \) and \( \text{Incr}(z) \) is defined by

(3.3) \[ x_{j+1} = \text{Incr}(h\lambda)x_j , \]

for \( h \) fixed and \( \lambda = \lambda x \). A graph for \( h\lambda \in \mathbb{R} \) is given in \textit{fig.3.1}. 

\textbf{fig.3.1 Incr(x)}
On can see, for real $\lambda$ also from fig. 3.1, that $\text{Incr}(z) \to 1$ for $|z| \to \infty$. This then implies that the estimator in a predictor-corrector setting should also approximate stiff modes, i.e. for $|h\lambda|$ large (whether $\lambda$ being positive or negative), by a quantity "close" to 1. Once a large step has been found it is heuristically clear that the stepsize does not rapidly becomes small for some large number of integrations.

§ 4. Getting through the smooth region.

The problem in singularly perturbed ODE is not finding one's way through the layer(s), which is an accuracy problem, so smaller steps commensurable with the stiffness parameters, but rather to integrate efficiently in between regions of high activity; that means we have to circumvent the restrictions set by the stability. With SYMIRK (see § 3) we are in good shape if only we have a smooth solution (i.e. without significant fast components) to give to the integrator for determining steps.

This smooth solution will be called the "pathfinder" as the grid obtained for this solution will be used to "approximate" fundamental system components. Note that we use quotes, for we have by no means an accurate approximation of the fast modes outside the layers; again we only want to limit their influence on the numerical stability and from this point of view an increment per step close to 1 is quite ideal in a multiple shooting context. The basic part of the "pathfinder technique" is an explicit integrator with only a small absolute stability region. Since we aim at fourth order accuracy we use the four step Adams-Bashforth

\begin{equation}
(4.1) \quad x_{j+1} = x_j + \frac{h}{24}(55\dot{x}_j - 59\dot{x}_{j-1} + 37\dot{x}_{j-2} - 9\dot{x}_{j-3}).
\end{equation}

To get an idea, the intersection of the stability region of this method with real axis is $[-.3,0]$. Now suppose we want to have a "smooth initial value" at $t=t_i$, which is a shooting point in a "low activity" region. First of all we can easily compute the projection of such a component onto the orthogonal complement of the unstable solution space, in other words: if $y$ is to be such a pathfinder, then we may compute $y_i^2$, defined by

\begin{equation}
(4.2) \quad \tilde{y}_i := Q_i^{-1}y_i ,
\end{equation}
where we identify (for simplicity of argument) $y_i$ with $y(t_i)$ and the partitioning is induced by the dichotomy as in (2.4). Choosing

\[(4.3)\quad \dot{y}_i^2 = 0 ,\]

we can easily find $\ddot{y}_i^2$ from (2.4b). Next we define

\[(4.4)\quad z(t_i) = Q_i \begin{bmatrix} 0 \\ \dot{y}_i^2 \end{bmatrix} ,\]

\[(4.5)\quad G(t_i) = Q_i\]

and first use the trapezoidal rule to determine approximate values at points $\tau_{-1}, \tau_{-2}$, where

\[(4.6)\quad \tau_j := t_i + jh ,\]

for some $h$ to be determined later on. Note that we have both at $\tau_{-1}$ and $\tau_{-2}$ $O(h^3)$ errors only. Second, we use the three step Adams-Bashforth

\[(4.7)\quad y_{j+1} = y_j + \frac{h}{12} (23\dot{y}_j - 16\dot{y}_{j-1} + 5\dot{y}_{j-2}) ,\]

to compute approximate $z(\tau_1)$ and $G(\tau_1)$. Note that the error at $\tau_1$ is $O(h^4)$. Now we have completed the start-up procedure for using (4.1) to compute $z(\tau_j)$ and $G(\tau_j)$, $j \geq 2$, giving fourth order (global) errors for $z(\tau_j)$. Schematically:

![fig. 4.1](image-url)
Because of the small stability region all fast modes will blow up (in fact we expect $|\text{Incr}(h\lambda)| \sim |h\lambda|$). The algorithm can check this effect by performing $QU$ factorizations as outlined in § 2. We stop when fast modes, in particular if the unstable fast modes, have grown by a factor larger than $TOL^{-1}$, where $TOL$ is the overall desired accuracy. Then we use a forward-backward recursion employing the decoupled parts (with a partitioning now that is usually different from the dichotomy in § 2) for the approximate solution $z(t)$ at the points $\tau_0, \tau_1, \ldots, \tau_M$ say. We thus have defining,

\begin{align}
(4.8) & \quad \tilde{y}_j = Q_j^{-1}z(\tau_j) , \\
(4.9) & \quad "\tilde{y}_M^1" = 0
\end{align}

(where now the superscript refers to the dichotomy induced by the Adams method!) and we start with "$\tilde{y}_0^2$" equal to that part of $\tilde{y}_0^2$ which still corresponds to the stable mode part of the recursion induced by this Adams method (roughly speaking $-3 \leq h\lambda \leq 0$ if $\lambda$ is real). This way we finally obtain an approximate $\tilde{y}_0$, so a "smooth initial value" $y(t_i)$ via

\begin{align}
(4.10) & \quad y(t_i) = Q_i\tilde{y}_0 .
\end{align}

Smooth in this sense means with only $O(TOL)$ components of fast solutions. Note that this pathfinder technique is aiming at damping out fast increasing modes only. For "intermediately" growing solutions we may want to live with the well-known vice of multiple shooting, viz. that the stepsize is determined by the fastest relevant (or significant) mode; in other words, it may be less expensive to have a pathfinder with still unstable components of some slow modes than to try to eliminate them completely.

Finally we address the question how to obtain a reasonable stepsize. Within the limits set by the global strategy, see § 5, we can try to monitor the activity of "the" smooth solution at $t_i$; by this we mean that we should try to find sufficiently high derivatives of a quantity that has the same smoothness properties as a solution with only $O(TOL)$ fast components. There are good reasons to assume that the forcing term $r(t)$ exhibits this activity. On the other hand decaying, but slow, components will more or less determine the magnitude of that solution. Hence we suggest to approximate $\|y^{(5)}(t_i)\|$ by
The stepsize $h$ should then accordingly be chosen such that

$$h^4EST_i \leq TOL.$$  

The derivative in (4.11) is found numerically, by using an interpolation/differentiation technique with a mesh $0(e_M^{1/2})$, where $e_M$ is the machine constant.

§ 5. Global strategy.

We now have indicated all basic ingredients. We repeat them here

(i) A dichotomic variable step integrator, capable both to track fast solutions in the layer and also to integrate smooth solutions (of the stiff problem) with a relaxed stepsize.

(ii) A pathfinder technique to obtain initial values of a smooth solution at shooting points.

(iii) A decoupling technique, which is used to make the marching procedure efficient and stable, but also is a part of the technique in (ii).

In addition we assume:

(iv) A method to determine the (real part of the) eigenvalues of $L(t)$ (e.g. the QR-algorithm).

The actual algorithm proceeds as follows:

(a) Split the interval $[\alpha, \beta]$ into two subintervals, $[\alpha, \frac{\alpha + \beta}{2}]$ and $[\frac{\alpha + \beta}{2}, \beta]$ and perform a marching technique on both, starting from $\alpha$ and $\beta$ respectively.

(b) Compute the eigenvalues of $L(\alpha)$. By assumption we have $Re(\lambda_i) \leq 0$, $Re(\mu_j) \geq 0$, $Re(\lambda_1) \leq \ldots \leq Re(\lambda_l)$, $Re(\mu_1) \geq \ldots \geq Re(\mu_k)$. Determine the width $w_1$ of the first layer.
\[ w_1 := \ln(TOL)/|Re(\lambda_1)| \]

(c) Choose as initial estimate for the stepsize

\[ h_1 := TOL^{\frac{1}{\mu}}/|Re(\lambda_1)| \]

and define

\[ h_{\min,1} = \frac{1}{4} h_1. \]

Choose initial values \( p_1(\alpha) = 0, F_1(\alpha) = I \) (cf. §2) and use SYMIRK to integrate with a stepsize \( \geq h_{\min,1} \)

(c1) either till the increments endanger stability (i.e. \( \|F_1(t)\| > \frac{TOL}{\varepsilon_M} \)), in which case we do a QU step, so choose a shooting point as in §2,

(c2) or till \( t = \alpha + w_1 \), so choose \( t \) as next shooting point.

(d) At \( t = \alpha + w_1 =: t_1 \) check whether the fastest component in the fundamental solution has decayed indeed by a factor \( TOL \) (by monitoring the diagonal of \( U_2 \), see §2). If this is not the case, continue till \( t_1 \) is acceptable as a layer endpoint.

(e) At \( t = t_1 \) use the pathfinder technique (somewhat adapted) to remove potentially growing components from \( p_1(t_1) \). Since the stepsize for this should commensurate with the activity of the next layer solution, use as a stepsize

\[ h_2 := \min\left( \frac{\|Q^{-1}Z_2^1 p_2(t_1)\|^2}{\|r(t_1)\|} \frac{\|r^{(5)}(t_1)\|}{|Re(\lambda_2)|} \right) \frac{TOL^{\frac{1}{\mu}}}{|Re(\lambda_2)|} \]

Of course this "removal" only makes sense for those modes that correspond to \( \mu_i \) with \( Re(\mu_i) \geq |Re(\lambda_1)| \). This can be monitored using the eigenvalues of \( L(t_1) \); hence perform such a QU step at each shooting point. By defining

\[ h_{\min,2} = \frac{1}{4} h_2, \]

we can use SYMIRK again, now for an obviously updated pathfinding solution \( p_2(t) \).

The process (a) - (d) can hence be repeated till we are through all layers. Once we are
in the smooth region we continue as outlined in § 4. Application of the decouling argument of § 2 then gives us a particular solution \( \{ z_i \} \) and a fundamental solution \( \{ \Phi_i \} \) for \( i=1, \ldots, m_1 \) say. Our derivation enables us to filter the thus obtained approximants as follows:

(f) If \( \exp[\min_\tau \Re(\mu_i(t)) \frac{\beta - \alpha}{2}] > TOL^{-1} \), set the \( i^{th} \) column of \( \Phi_i \) equal to zero.

We thus have within accuracy \( TOL \)

\[
x(t_i) = Q_i[\Phi_i c + z_i] + p_i(t_i) \quad i=1, \ldots, m_1.
\]

Similarly, we can perform the same procedure on \( [\beta, \frac{\alpha + \beta}{2}] \), thus giving us, say

\[
x(s_i) = R_i[\Psi_i d + u_i] + q_i(s_i) \quad i=1, \ldots, m_2,
\]

where the \( s_i \) are the shooting points with \( s_1 = \beta \) and \( s_{m_2} = \frac{\alpha + \beta}{2} \); moreover \( R_i \) is orthogonal and \( \{ \Psi_i \}, \{ u_i \} \) are the fundamental and particular solution of the upper triangular recursion, with those columns of \( \Psi_i \) equal to zero that correspond to modes with \( \exp[\min_\tau \Re(\lambda_i(t)) \frac{\alpha - \beta}{2}] > TOL^{-1} \) (cf. (f)).

The last step now is

(g) Match the two parts of the solution at \( \frac{\alpha + \beta}{2} \) and use the BC in order to determine \( c \) and \( d \):

\[
\begin{bmatrix}
Q_{m_1} \Phi_{m_1} - R_{m_2} \Psi_{m_2} \\
M_\alpha Q_1 \Phi_1 \
M_\beta R_1 \Psi_1
\end{bmatrix}
\begin{bmatrix}
c \\
d
\end{bmatrix}
= \begin{bmatrix}
R_{m_2} U_{m_2} + q_{m_1}(s_{m_2}) - Q_{m_1} z_{m_1} - p_{m_1}(t_{m_1}) \\
0 - M_\alpha Q_1 z_1 - M_\beta R_1 u_1
\end{bmatrix}.
\]

Note that we thus obtain an approximate solution at the (possible user requested) shooting points and at the layer endpoints as well.
§ 6. An example.

We have implemented the previously outlined algorithm in a FORTRAN code MUTSSYM, which uses some features of BOUNDPAK (cf. [8]), but for the rest employs the special strategy of § 5, including SYMIRK as an adaptive integrator. Consider the following test problem:

\[
\frac{dx}{dt} = \begin{bmatrix} 0 & \lambda \\ \lambda & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1 - \lambda^2}{\lambda} e^t \end{bmatrix}
\]

with the boundary condition

\[
x(0) + x(2) = (1 + e^2, (1 + e^2) / \lambda)^T.
\]

The solution of this problem is \(x = (e^t, e^t / \lambda)^T\).

In tables 6.3, 6.4 and 6.5 the results of the computations with MUTSYM are given for different \(\lambda\) and different absolute tolerances. We allowed the integrator to take at most 20 steps per minor shooting interval (see [6]) and we asked for output at \(t=0\), \(t=1\), \(t=2\) and at the layer points. The maximum error occurs at \(t=0\) and \(t=2\). Note that the number of steps taken in the layers is independent of \(\lambda\). Furthermore one can see that for \(\lambda=1000\), in the smooth region \(|h\lambda|\) is not large enough to eliminate the influence of the fast modes in SYMIRK. This results in a smaller stepsize and also in a smaller error in the solution (see table 6.4 and 6.5). This also explains why the number of steps taken in the smooth region is larger for \(\lambda=1000\) than for \(\lambda=10,000\), \(\lambda=100,000\) and \(\lambda=1,000,000\). \(\lambda=1000\) and \(\lambda=10000\), for several tolerances.

These results were computed on an Olivetti PC M24 computer.
### Table (6.3)

**TOL = 1.0 · 3**

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>( 1.0 +3 )</th>
<th>( 1.0 +4 )</th>
<th>( 1.0 +5 )</th>
<th>( 1.0 +6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>layer at 0</td>
<td>7.6191 ·3</td>
<td>7.6191 ·4</td>
<td>7.6191 ·5</td>
<td>7.6191 ·6</td>
</tr>
<tr>
<td>layer at 2</td>
<td>1.99238</td>
<td>1.999238</td>
<td>1.9999238</td>
<td>1.99999238</td>
</tr>
<tr>
<td># steps</td>
<td>77</td>
<td>76</td>
<td>76</td>
<td>76</td>
</tr>
<tr>
<td># steps</td>
<td>101</td>
<td>102</td>
<td>102</td>
<td>102</td>
</tr>
<tr>
<td># steps</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>smooth region</td>
<td>8.16 ·2</td>
<td>8.33 ·2</td>
<td>8.00 ·2</td>
<td>8.16 ·2</td>
</tr>
<tr>
<td>min stepsize</td>
<td>1.48 -2</td>
<td>4.36 -2</td>
<td>4.74 -2</td>
<td>4.33 -2</td>
</tr>
<tr>
<td>max error</td>
<td>3.70 -4</td>
<td>1.18 -3</td>
<td>1.32 -3</td>
<td>1.33 -3</td>
</tr>
<tr>
<td>total # steps</td>
<td>199</td>
<td>199</td>
<td>200</td>
<td>200</td>
</tr>
</tbody>
</table>

### Table (6.4)

**TOL = 1.0 · 4**

<table>
<thead>
<tr>
<th>( \lambda )</th>
<th>( 1.0 +3 )</th>
<th>( 1.0 +4 )</th>
<th>( 1.0 +5 )</th>
<th>( 1.0 +6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>layer at 0</td>
<td>9.6103 ·3</td>
<td>9.6103 ·4</td>
<td>9.6103 ·5</td>
<td>9.6103 ·6</td>
</tr>
<tr>
<td>layer at 2</td>
<td>1.99039</td>
<td>1.999039</td>
<td>1.9999039</td>
<td>1.99999039</td>
</tr>
<tr>
<td># steps</td>
<td>148</td>
<td>148</td>
<td>148</td>
<td>148</td>
</tr>
<tr>
<td># steps</td>
<td>200</td>
<td>200</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td># steps</td>
<td>45</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>smooth region</td>
<td>1.48 ·2</td>
<td>4.36 ·2</td>
<td>4.74 ·2</td>
<td>4.33 ·2</td>
</tr>
<tr>
<td>min stepsize</td>
<td>5.49 ·5</td>
<td>1.28 ·4</td>
<td>1.77 ·4</td>
<td>1.82 ·4</td>
</tr>
<tr>
<td>max error</td>
<td>388</td>
<td>383</td>
<td>383</td>
<td>383</td>
</tr>
<tr>
<td>total # steps</td>
<td>383</td>
<td>383</td>
<td>383</td>
<td>383</td>
</tr>
</tbody>
</table>
Table (6.5)

TOL = 1.0 -5

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>1.0 +3</th>
<th>1.0 +4</th>
<th>1.0 +5</th>
<th>1.0 +6</th>
</tr>
</thead>
<tbody>
<tr>
<td>layer at 0</td>
<td>11.86 -3</td>
<td>11.86 -4</td>
<td>11.86 -5</td>
<td>11.86 -6</td>
</tr>
<tr>
<td>layer at 2</td>
<td>1.9879</td>
<td>1.99879</td>
<td>1.999879</td>
<td>1.9999879</td>
</tr>
<tr>
<td># steps</td>
<td>272</td>
<td>272</td>
<td>272</td>
<td>272</td>
</tr>
<tr>
<td>layer at 0</td>
<td>436</td>
<td>436</td>
<td>436</td>
<td>436</td>
</tr>
<tr>
<td># steps</td>
<td>72</td>
<td>61</td>
<td>61</td>
<td>61</td>
</tr>
<tr>
<td>smooth region</td>
<td>7.58 -3</td>
<td>2.56 -2</td>
<td>2.46 -2</td>
<td>2.41 -2</td>
</tr>
<tr>
<td>min stepsize</td>
<td>6.70 -8</td>
<td>5.93 -6</td>
<td>1.55 -5</td>
<td>1.71 -5</td>
</tr>
<tr>
<td>total # steps</td>
<td>780</td>
<td>769</td>
<td>769</td>
<td>769</td>
</tr>
</tbody>
</table>

References


