7.1 Introduction

We’d like to extend the initial-value or shooting procedures to vector systems of \( m \) first-order ODEs. However, before studying nonlinear problems, let’s focus on the linear system

\[
y' = Ay + b \tag{7.1.1a}
\]

with separated boundary conditions.

\[
Ly(a) = l, \tag{7.1.1b}
\]

\[
Ry(b) = r \tag{7.1.1c}
\]

We’ll suppose that \( A \) is \( m \times m \), \( L \) is \( l \times m \), \( R \) is \( r \times m \), \( b \) is an \( m \)-vector, \( l \) is an \( l \)-vector, and \( r \) is an \( r \)-vector. Of course, \( l + r = m \).

It’s simple to develop shooting procedures be realizing that the solution of the linear BVP (7.1.1) can be written in the form \([3, 4]\)

\[
y(x) = Y(x)c + v(x). \tag{7.1.2a}
\]

The function \( v(x) \) is a particular solution satisfying, e.g.,

\[
v' = Av + b, \quad v(a) = \alpha, \tag{7.1.2b}
\]
for some convenient choice of the vector $\alpha$ (e.g., $\alpha = 0$). The $m \times m$ matrix $Y(x)$ is a set of fundamental solutions satisfying

$$Y' = AY, \quad Y(a) = I,$$  

(7.1.2c)

where $I$ is the $m \times m$ identity matrix. The constant vector $c$ is determined by satisfying the initial and terminal conditions (7.1.1b,c) give

$$\begin{bmatrix} LY(a) \\ Ry(b) \end{bmatrix} = Qc + \begin{bmatrix} L \v a(a) \\ R \v a(b) \end{bmatrix} = \begin{bmatrix} 1 \\ r \end{bmatrix}.$$  

(7.1.3a)

The solution exists since

$$Q = \begin{bmatrix} LY(a) \\ Ry(b) \end{bmatrix} = \begin{bmatrix} L \\ Ry(b) \end{bmatrix}$$  

(7.1.3b)

has full rank $m$ for a well-posed problem. Once $c$ has been determined, the solution of the BVP is determined by (7.1.2a). The procedure requires the solution of the $m + 1$ vector IVPs (7.1.2b,c).

Although this technique seems straight forward, we recall that the stability of BVPs is quite different from that for IVPs. An IVP for (7.1.1a) is stable when the eigenvalues of $A$ have non-positive real parts. This is generally not the case for BVPs where the eigenvalues of $A$ for stable problems can have positive and negative real parts. Let’s explore a simple example.

**Example 7.1.1** ([1], Section 4.2). Consider the second-order linear BVP

$$y'' - R^2 y = 1, \quad y(0) = y(1) = 0,$$

and write this as a first-order system by letting

$$y_1 = y, \quad y_2 = \frac{y'}{R}.$$  

This leads to the symmetric system

$$y' = \begin{bmatrix} 0 & R \\ R & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ 1/R \end{bmatrix}$$

with the initial and terminal conditions

$$[1 \ 0]y(0) = 0, \quad [1 \ 0]y(1) = 0.$$  

2
We’ll let the particular solution satisfy
\[ \mathbf{v}' = \begin{bmatrix} 0 & R \\ R & 0 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 0 \\ 1/R \end{bmatrix}, \quad \mathbf{v}(0) = \begin{bmatrix} -1/R^2 \\ 0 \end{bmatrix}. \]

Thus,
\[ \mathbf{v}(x) = \begin{bmatrix} -1/R^2 \\ 0 \end{bmatrix}. \]

The set of fundamental solutions satisfies
\[ \mathbf{Y}' = \begin{bmatrix} 0 & R \\ R & 0 \end{bmatrix} \mathbf{Y}, \quad \mathbf{Y}(0) = \mathbf{I}. \]

The matrix
\[ \mathbf{A} = \begin{bmatrix} 0 & R \\ R & 0 \end{bmatrix} \]
has the eigenvalues \( \lambda = \pm R \); thus, an IVP would be unstable. We’ll have to investigate the stability of the BVP.

We write the fundamental solution as
\[ \mathbf{Y}(x) = \begin{bmatrix} \cosh Rx & \sinh Rx \\ \sinh Rx & \cosh Rx \end{bmatrix}. \]

From (7.1.2a), the solution of this BVP has the form
\[ \mathbf{y}(x) = \begin{bmatrix} \cosh Rx & \sinh Rx \\ \sinh Rx & \cosh Rx \end{bmatrix} \mathbf{c} + \begin{bmatrix} -1/R^2 \\ 0 \end{bmatrix}. \]

The initial and terminal conditions (7.1.3) are
\[ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} -1/R^2 \\ -1/R^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \]

Thus,
\[ \mathbf{c} = \frac{1}{R^2} \begin{bmatrix} 1 \\ \frac{1-\cosh R}{\sinh R} \end{bmatrix}, \]
and the solution of the BVP is
\[ \mathbf{y}(x) = \frac{1}{R^2} \begin{bmatrix} \cosh Rx & \sinh Rx \\ \sinh Rx & \cosh Rx \end{bmatrix} \begin{bmatrix} 1 \\ \frac{1-\cosh R}{\sinh R} \end{bmatrix} + \begin{bmatrix} -1/R^2 \\ 0 \end{bmatrix}. \]

Difficulties arise when \( R \gg 1 \). In this case, the solution \( y_1(x) \) is asymptotically given by
\[ y_1 \approx \frac{1}{R^2} [e^{-Rx} + e^{-R(1-x)} - 1]. \]
Thus, it is approximately \(-1/R^2\) away from the boundaries at \(x = 0\) and \(1\) with narrow boundary layers near both ends.

When \(Rx\) is large, we would not be able to distinguish between \(\sinh Rx\) and \(\cosh Rx\) and, instead of the exact result, we would compute

\[
y(x) \approx \frac{1}{2} \begin{bmatrix} e^{Rx} & e^{Rx} \\ e^{Rx} & e^{Rx} \end{bmatrix} \mathbf{c} + \begin{bmatrix} -1/R^2 \\ 0 \end{bmatrix}, \quad x > 0.
\]

The matrix \(Y(x)\) would be singular in this case. With small discernable differences between \(\sinh Rx\) and \(\cosh Rx\), \(Y\) would be ill-conditioned. Assuming this to be so, let’s write the solution as

\[
y(x) \approx \frac{1}{2} \begin{bmatrix} e^{Rx}(1 + \delta) & e^{Rx}(1 - \delta) \\ e^{Rx}(1 - \delta) & e^{Rx}(1 + \delta) \end{bmatrix} \mathbf{c} + \begin{bmatrix} -1/R^2 \\ 0 \end{bmatrix}, \quad x > 0,
\]

where \(\delta \ll 1\). (If \(\delta = e^{-2R}\) then the above relation is exact at \(x = 1\).) We would then determine \(\mathbf{c}\) as

\[
\mathbf{c} \approx \frac{1}{R^2} \begin{bmatrix} 1 \\ -1 \end{bmatrix}
\]

and the solution as

\[
y(x) \approx \frac{\delta e^{Rx}}{2R^2} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad x > 0.
\]

With \(R\) large, there is the possibility of having catastrophic growth in the solution even when \(\delta\) is small. In order to demonstrate this, we solved this problem using the MATLAB Runge-Kutta procedure \texttt{ode45}. While an explicit Runge-Kutta code is not the most efficient solution procedure for this problem, efficiency was not our main concern. The maximum errors in the solution component \(y_1\) with \(R = 1, 10\) are reported in Table 7.1.1.

The error has grown by six decades for a ten-fold increase in \(R\). Indeed, the procedure failed to find a solution of the BVP with \(R = 100\).

<table>
<thead>
<tr>
<th>(R)</th>
<th>(|\varepsilon_1|<em>{\infty}/|y_1|</em>{\infty})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(2.445 \times 10^{-8})</td>
</tr>
<tr>
<td>10</td>
<td>(2.940 \times 10^{-2})</td>
</tr>
</tbody>
</table>

Table 7.1.1: Maximum errors in \(y_1\) for Example 7.1.1

Let’s pursue this difficulty with a bit more formality. The exact solution of the linear
BVP (7.1.1) can be written as (Problem 1)

\[ y(x) = Y(x)Q^{-1}\begin{bmatrix} 1 \\ x \end{bmatrix} + \int_a^b G(x, \xi)b(\xi)d\xi \quad (7.1.4a) \]

where \( Q \) was given by (7.1.3b) and \( G(x, \xi) \) is the Green's function

\[
G(x, \xi) = \begin{cases} 
Y(x)Q^{-1}\begin{bmatrix} L \\ 0 \end{bmatrix}Y(a)Y^{-1}(\xi), & \text{if } \xi \leq x \\
-Y(x)Q^{-1}\begin{bmatrix} 0 \\ R \end{bmatrix}Y(b)Y^{-1}(\xi), & \text{if } \xi > x 
\end{cases} \quad (7.1.4b)
\]

The Green's function behaves like the inverse of the differential operator. We may use it to define the stability of BVP.

**Definition 7.1.1.** The BVP (7.1.1) is stable if there exists a constant \( \kappa \) such that

\[
\|y(\cdot)\|_{\infty} \leq \kappa[|I|_{\infty} + |R|_{\infty} + \int_a^b |b(\xi)|_{\infty}d\xi] \quad (7.1.5a)
\]

where

\[ \|y(\cdot)\|_{\infty} = \max_{a \leq x \leq b} |y(x)|_{\infty}, \quad |b|_{\infty} = \max_{1 \leq i \leq m} |b_i|. \quad (7.1.5b) \]

From (7.1.4a), we see that

\[ \kappa = \max(\|Y(\cdot)Q^{-1}\|_{\infty}, \|G(\cdot, \cdot)\|_{\infty}) \quad (7.1.5c) \]

assuming that all quantities are bounded. Thus, the solution of a stable BVP is bounded by its data. Following the techniques used for IVPs, we may show that this also applies to a perturbation of the data associated with the BVP (7.1.1) ([2], Chapter 6); hence, the reason for the name stability.

**Definition 7.1.2.** The BVP (7.1.1) has exponential dichotomy if there are positive constants \( K, \alpha, \) and \( \beta \) such that

\[
\|G(\cdot, \cdot)\|_{\infty} \leq K \begin{cases} 
 e^{\alpha(x - \xi)}, & \text{if } \xi \leq x \\
 e^{\beta(x - \xi)}, & \text{if } \xi > x 
\end{cases} \quad (7.1.6)
\]

The BVP has dichotomy if (7.1.6) holds with \( \alpha = \beta = 0. \)
Remark 1. The notions of dichotomy and exponential dichotomy correspond to stability and asymptotic stability, respectively, for an IVP.

It is relatively easy to show that the stability constant $\kappa$ for Example 7.1.1 has a modest size and that this problem has exponential dichotomy (Problem 2). The problem, once again, is that the IVP is unstable and this renders $Q$ ill conditioned.

There is an alternative shooting procedure for linear systems that is slightly more efficient than (7.1.2 - 7.1.3) when $m$ is large. We’ll present it, although it will not solve our difficulties.

1. Obtain a particular solution $\mathbf{v}$ that satisfies

$$\mathbf{v}' = A\mathbf{v} + \mathbf{b}, \quad L\mathbf{v}(a) = \mathbf{l}. \quad (7.1.7)$$

We’ll have to describe a means of computing $\mathbf{v}(a)$ so that it satisfies the initial condition (7.1.1b), but let’s postpone this.

2. Obtain linearly-independent solutions $\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \ldots, \mathbf{u}^{(r)}$, to the IVPs

$$(\mathbf{u}^{(i)})' = A\mathbf{u}^{(i)}, \quad L\mathbf{u}^{(i)}(a) = \mathbf{0}, \quad i = 1, 2, \ldots, r. \quad (7.1.8)$$

Again, we’ll assume that a means of finding $\mathbf{u}^{(i)}, i = 1, 2, \ldots, r$, to satisfy the initial condition (7.1.1b) can be found. With this, we observe that the work of solving this system is about half that of the previous procedure (7.1.2c) when the number of terminal conditions $r$ is approximately $m/2$.

3. Write the general solution of the linear BVP (7.1.1) as

$$\mathbf{y}(x) = c_1\mathbf{u}^{(1)}(x) + c_2\mathbf{u}^{(2)}(x) + \ldots + c_r\mathbf{u}^{(r)}(x) + \mathbf{v}(x)$$

or

$$\mathbf{y}(x) = \mathbf{U}(x)c + \mathbf{v}(x) \quad (7.1.9a)$$

where

$$\mathbf{U}(x) = [\mathbf{u}^{(1)}(x), \mathbf{u}^{(2)}(x), \ldots, \mathbf{u}^{(r)}(x)], \quad c = [c_1, c_2, \ldots, c_r]^T. \quad (7.1.9b)$$
4. By construction, this representation satisfies the ODE (7.1.1a) and the initial conditions (7.1.1b). It remains to satisfy the terminal conditions (7.1.1c) and this can be done by determining \( c \) so that

\[
\mathbf{R} \mathbf{y}(b) = \mathbf{R}[\mathbf{U}(b) c + \mathbf{v}(b)] = \mathbf{r}
\]

or

\[
\mathbf{R} \mathbf{U}(b) c = \mathbf{r} - \mathbf{R} \mathbf{v}(b). \tag{7.1.10}
\]

As noted, the main savings of this procedure relative to (7.1.2-7.1.3) is the reduced number of ODEs that have to be solved. This is offset by (possible) difficulties associated with selecting a basis \( \mathbf{U}, \mathbf{v} \) to satisfy the initial conditions (7.1.1b).

**Example 7.1.2.** Conte [5] constructed the constant-coefficient fourth-order BVP

\[
y^{iv} - (1 + R)y'' + Ry = -1 + \frac{Rx^2}{2}, \quad 0 < x < 1,
\]

\[
y(0) = y'(0) = 1, \quad y(1) = \frac{3}{2} + \sinh 1, \quad y'(1) = 1 + \cosh 1.
\]

To try to explain difficulties that researchers were having when solving certain flow-instability problems. We’ll rewrite the problem as the first-order system

\[
\mathbf{A} = \begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
-R & 0 & 1 + R & 0
\end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix}
y_1 \\
y_2 \\
y_3 \\
y_4
\end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix}
0 \\
0 \\
0 \\
-1 + Rx^2/2
\end{bmatrix}.
\]

\[
\mathbf{L} = \mathbf{R} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}, \quad \mathbf{l} = \begin{bmatrix}
1 \\
1
\end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix}
\frac{3}{2} + \sinh 1 \\
1 + \cosh 1
\end{bmatrix}.
\]

Let’s go through the steps in solving this problem by the shooting procedure that we just described. We note that

\[
\mathbf{v}(0) = [1 \ 1 \ 0 \ 0]^T
\]

satisfies the initial conditions and, thus, the particular-solution problem (7.1.7) has been specified. Similarly,

\[
\mathbf{u}^{(1)} = \begin{bmatrix}
0 \\
0 \\
1 \\
0
\end{bmatrix}, \quad \mathbf{u}^{(2)} = \begin{bmatrix}
0 \\
0 \\
0 \\
1
\end{bmatrix}.
\]
satisfy trivial versions of the initial conditions; hence, the homogeneous problems (7.1.9)
are specified.

Conte [5] solved the IVPs (7.1.7, 7.1.9) using a fixed-step Runge-Kutta method. (It
was done in the 1960s.) His results for the error $\|e_1\|_\infty$ in the first component $y_1(x)$ of
the solution are reported in Table 7.1.2 for $R = 400, 3600$. The exact solution of this
problem is
\[ y_1(x) = 1 + \frac{x^2}{2} + \sinh x. \]

<table>
<thead>
<tr>
<th>$R$</th>
<th>$|e_1|_\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>400</td>
<td>0.00173</td>
</tr>
<tr>
<td>3600</td>
<td>400</td>
</tr>
</tbody>
</table>

Table 7.1.2: Maximum errors of Example 7.1.2 using Conte’s [5] linear shooting procedure.

The difficulties with this problem are similar to those of Example 7.1.1. The fundamental
solutions of the ODE are

\[ \sinh x, \quad \cosh x, \quad \sinh \sqrt{Rx}, \quad \cosh \sqrt{Rx}. \]

Although the exact solution doesn’t depend on the rapidly growing components ($\sinh \sqrt{Rx}
and $\cosh \sqrt{Rx}$), the IVPs (7.1.7) and (7.1.9) do. Thus, the matrix $RU(b)$ appearing in
(7.1.10) will be ill-conditioned when the parameter $R$ is large. The two components of
the fundamental solution $\sinh \sqrt{Rx}$ and $\cosh \sqrt{Rx}$ are numerically linearly dependent for
large values of $\sqrt{Rx}$. Although the exact solution of the BVP is independent of these
rapidly growing components, small round off errors introduce them and they eventually
dominate the exact solution.

**Problems**

1. Show that the solution representation (7.1.4 satisfies the linear BVP (7.1.1).

2. Find the Green’s function for the BVP of Example 7.1.1. Find the stability con-
   stant. Show that this problem is stable and has exponential dichotomy.
7.2 Multiple Shooting

As seen in Section 7.1, errors may accumulate as the solution is generated from the initial to the terminal point. Since round-off errors increase in proportion to the number of operations, they can be reduced by integrating over shorter distances. With this procedure, called parallel or multiple shooting, we create IVPs that start at several points on \([a, b]\). Let us illustrate the procedure with a linear \(m\)-dimensional BVP of the form

\[
y' = Ay + b, \quad a < x < b, \quad (7.2.1a)
\]

\[
Ly(a) = l, \quad (7.2.1b)
\]

\[
Ry(b) = r, \quad (7.2.1c)
\]

with \(L\) of dimension \(l \times m\) and \(R\) of dimension \(r \times m\). Let us further divide \([a, b]\) into \(N\) subintervals as shown in Figure 7.2.1. Multiple shooting procedures that are based on the strategies \((7.1.2 - 7.1.3)\) and \((7.1.7 - 7.1.10)\) have been developed. The first approach is the simpler of the two, so let’s begin there.

1. On each subinterval \(x_{j-1}, x_j\), \(j = 1, 2, \ldots, N\), solve the IVPs

\[
v'_j = Av_j + b, \quad v_j(a) = 0, \quad (7.2.2a)
\]
\[ Y_j' = AY_j, \quad Y_j(a) = I, \quad x_{j-1} < x \leq x_j, \quad j = 1, 2, \ldots, N, \quad (7.2.2b) \]

where \( \alpha_j, j = 1, 2, \ldots, N, \) are chosen and \( I \) is the \( m \times m \) identity matrix.

2. As with simple shooting, consider the solution of the BVP in the form (7.1.2a), which now becomes

\[ y(x) = Y_j(x)c_j + v_j(x), \quad x_{j-1} < x \leq x_j, \quad j = 1, 2, \ldots, N. \quad (7.2.3) \]

The solution must be continuous at the interior shooting points \( x_j, j = 1, 2, \ldots, N - 1; \) thus, with the initial conditions specified by (7.2.2)

\[ Y_j(x_j)c_j + v_j(x_j) = c_{j+1}, \quad j = 1, 2, \ldots, N - 1. \]

The boundary conditions (7.2.1b,c) must also be satisfied and, with (7.2.3) and (7.2.2), this implies

\[ Ly(a) = L Y_1 c_1 + L v(a) = Lc_1 = l, \quad Ry(b) = R Y_N(b)c_N + R v_N(b) = r. \]

Writing this system in matrix form

\[ Ac = g \quad (7.2.4a) \]

where

\[ A = \begin{bmatrix}
L & I \\
-Y_1(x_1) & I \\
-Y_2(x_2) & I \\
& \ddots & \ddots \\
& & -Y_{N-1}(x_{N-1}) & I \\
& & & R Y_N(b)
\end{bmatrix} \quad (7.2.4b) \]

\[ c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}, \quad g = \begin{bmatrix} l \\ v_1(x_1) \\ \vdots \\ v_{N-1}(x_{N-1}) \\ r - R v_N(b) \end{bmatrix}. \quad (7.2.4c) \]

We will encounter algebraic systems like this in this and the following chapters. We’ll describe procedures for solving them in Chapter 8 that only require \( O(N) \) operations.
3. Once $c$ has been determined, the BVP solution may be reconstructed from (7.2.3).  

Remark 1. All of the IVPs (7.2.2) can be solved in parallel on a computer with $N$ processors. This is why multiple shooting is also called parallel shooting.

Conte [5] described a method for stabilizing the procedure (7.1.7 - 7.1.10) which was later automated and improved by Scott and Watts [11]. The technique is less parallel than the one that we just described, but it does offer some advantages. To begin, we calculate solutions of the IVPs

$$
U'_1 = AU_1, \quad x_0 < x < x_1, \quad LU_1(a) = 0, \\
v'_1 = Av_1 + b, \quad x_0 < x \leq x_1, \quad Lv_1(a) = l,
$$

and write the solution of the BVP as

$$
y(x) = U_1(x)c_1 + v_1(x), \quad x_0 < x < x_1.
$$

As in Section 7.1,

$$
U_1 = [u_1^{(1)}, u_1^{(2)}, \ldots, u_1^{(r)}].
$$

The integration proceeds to a point $x_1$ where, due to the accumulation of round off errors, the columns of $U_1$ no longer form a good linearly independent basis for the solution. The point $x_1$ is determined as part of the solution process but, for the present, let’s assume that it is known.

New initial conditions $U_2(x_1)$ are determined by orthogonalizing the columns of $U_1(x_1)$. We’ll subsequently show that this is equivalent to finding an upper triangular matrix $P_1$ such that

$$
U_2(x_1)P_1 = U_1(x_1).
$$

We also select new initial conditions $v_2(x_1)$ for the particular solution that are in the orthogonal complement of $U_2(x_1)$. Again, we’ll postpone the details and note that this is equivalent to determining a vector $w_1$ such that

$$
v_2(x_1) = v_1(x_1) - U_2(x_1)w_1.
$$

The time integration can now continue by solving

$$
U'_2 = AU_2, \quad x_1 < x < x_2,
$$

11
\[ v'_1 = Av_1 + b, \quad x_1 < x < x_2. \]

The quantities \( U_2(x_1) \) and \( v_2(x_1) \) are used as initial conditions. The process continues to \( x_2 \) where the columns of \( U_2(x_2) \) fail to satisfy a linear independence test.

The general procedure is summarized as follows.

1. Determine \( U_1(a) \) and \( v_1(a) \) as solutions of

\[
LU_1(a) = 0, \quad (7.2.5a)
\]

\[
Lv_1(a) = l. \quad (7.2.5b)
\]

2. Beginning with \( j = 1 \), solve the IVPs

\[
U'_j = AU, \quad x_{j-1} < x < x_j, \quad (7.2.6a)
\]

\[
v'_j = Av + b, \quad x_{j-1} < x < x_j. \quad (7.2.6b)
\]

Let the solution of the BVP have the form

\[
y(x) = U_j(x)c_j + v_j(x), \quad x_{j-1} < x < x_j. \quad (7.2.6c)
\]

3. At a failure of a linear independence test, compute new initial conditions \( U_{j+1}(x_j) \) and \( v_{j+1}(x_j) \) by solving

\[
U_{j+1}(x_j)P_j = U_j(x_j) \quad (7.2.7a)
\]

\[
v_{j+1}(x_j) = v_j(x_j) + U_{j+1}(x_j)w_j. \quad (7.2.7b)
\]

4. Repeat the previous two steps until reaching the terminal point \( x_N = b \). The BVP solution at \( b \) is

\[
y(b) = U_N(b)c_N + v_N(b), \quad (7.2.8)
\]

5. Using the terminal condition

\[
Ry(b) = RU_N(b)c_N + Rv_N(b) = r,
\]

determine \( c_N \) as the solution of

\[
RU_N(b)c_N = Rv_N(b) - r \quad (7.2.8)
\]
6. Once \( c_N \) has been determined, the remaining \( c_j, j = 1, 2, \ldots, N-1 \), are determined by enforcing solution continuity at the orthogonalization points, i.e.,

\[
U_j(x_j)c_j + v_j(x_j) = U_{j+1}(x_{j+1})c_{j+1} + v_{j+1}(x_j).
\]

This relation can be simplified by the use of (7.2.7) to

\[
U_{j+1}(x_j)P_j c_j + v_j(x_j) = U_{j+1}(x_j)c_{j+1} + v_j(x_j) + U_{j+1}(x_j)w_j
\]

or

\[
U_{j+1}(x_j)\left[P_j c_j - c_{j+1} - w_j\right] = 0.
\]

Since \( P_j \) is non-singular, we may take the solution of this system as

\[
P_j c_j = c_{j+1} + w_j, \quad j = N - 1, N - 2, \ldots, 1. \tag{7.2.9}
\]

Several details still remain, including

1. the satisfaction of the initial conditions (7.2.5a,b),

2. the orthogonalization procedure (7.2.7a,b), and

3. the test for a linearly independent basis.

Consider the solution of the under-determined system (7.2.5a,b) first. The \( r \) columns of \( U_1(a) \) are calculated so that they are mutually orthogonal and span the null space of \( L \). A procedure follows.

1. Reduce \( L^T \) to upper triangular form using the QR algorithm ([6], Chapter 3). This involves finding an \( m \times m \) orthogonal matrix \( Q \) such that

\[
QL^T = \begin{bmatrix} T \\ 0 \end{bmatrix}. \tag{7.2.10a}
\]

Recall that an orthogonal matrix is one where

\[
Q^T Q = I. \tag{7.2.10b}
\]

The matrix \( T \) is an \( l \times l \) upper triangular matrix and the zero matrix is \( r \times l \). Pivoting can frequently be ignored with orthogonal transformations and we’ll assume this to be the case here.
2. Choose
\[ U_1(a) = Q^T \begin{bmatrix} 0 \\ I \end{bmatrix} \]  
(7.2.11)

where I is the \(r \times r\) identity matrix and the zero matrix is \(l \times r\). Thus \(U_1\) is the last \(r\) columns of \(Q^T\). Using (7.2.10) and (7.2.11) we verify that
\[ LU_1(a) = LQ^T \begin{bmatrix} 0 \\ I \end{bmatrix} = [T^T 0] \begin{bmatrix} 0 \\ I \end{bmatrix} = 0. \]

3. The vector \(v_1(a)\) is obtained as the least squares solution [9] of (7.2.5b) by solving
\[ T^T s = l \]  
(7.2.12a)

for \(s\) using forward substitution and setting
\[ v_1(a) = Q^T \begin{bmatrix} s \\ 0 \end{bmatrix}. \]  
(7.2.12b)

We readily verify that
\[ L v_1(a) = LQ^T \begin{bmatrix} s \\ 0 \end{bmatrix} = [T^T 0] \begin{bmatrix} s \\ 0 \end{bmatrix} = T^T s = l \]

and, using (7.2.12b) and (7.2.11),
\[ v_1^T(a)U_1(a) = [s^T 0]QQ^T \begin{bmatrix} 0 \\ I \end{bmatrix} = 0. \]

The orthogonalization task (7.2.7) can also be done by the QR procedure.

1. Determine an orthogonal matrix \(Q\) such that
\[ QU_j = \begin{bmatrix} P_j \\ 0 \end{bmatrix} \]  
(7.2.13a)

where \(P_j\) is an \(r \times r\) upper triangular matrix and the zero matrix is \(l \times r\). (Spatial arguments in (7.2.7) at \(x_j\) have been omitted for clarity. Although the symbol \(Q\) has been used in (7.2.10 - 7.2.12), the matrix in (7.2.13a) is generally not the same.)

2. Select
\[ U_{j+1} = Q^T \begin{bmatrix} I \\ 0 \end{bmatrix}. \]  
(7.2.13b)
Using (7.2.13a)

\[ U_j = Q^T \begin{bmatrix} P_j \\ 0 \end{bmatrix}. \]

According to (7.2.13b)

\[ Q^T = [U_{j+1} \ V_{j+1}] \]

Thus,

\[ U_j = [U_{j+1}, \ V_{j+1}] \begin{bmatrix} P_j \\ 0 \end{bmatrix} = U_{j+1}P_j. \]

3. In order to satisfy

\[ v_{j+1}^T U_{j+1} = 0, \]

choose

\[ w_j^T = -v_j^T U_{j+1}. \hspace{1cm} (7.2.14) \]

To verify this, consider (7.2.7b), (7.2.14), and

\[ v_{j+1}^T U_{j+1} = (v_j^T + w_j^T U_{j+1})U_{j+1} = v_j^T U_{j+1} + w_j^T = 0. \]

Conte [5] suggested examining the “angle” between pairs of columns of \( U_j(x_j) \) and re-orthogonalizing whenever this angle became too small. Again, we’ll omit the spatial argument \( x_j \) and compute

\[ \cos \alpha_{ik} = \frac{(u_j^{(i)}, u_j^{(k)})}{|u_j^{(i)}||u_j^{(k)}|}, \quad i, k = 1, 2, \ldots, r - l, \]

where

\[ (u, w) = v^T w, \quad |u|_2 = \sqrt{(u, u)}. \hspace{1cm} (7.2.15a) \]

The matrix \( U_j \) is re-orthogonalized whenever

\[ \min_{1 \leq i, k \leq r - l} |\alpha_{ik}| < \epsilon \hspace{1cm} (7.2.15c) \]

**Example 7.2.1.** Solve Example 7.1.2 with \( R = 3600 \) using the multiple shooting procedure with orthogonalization described above with \( \epsilon = 10^6 \). The result for the maximum error in the component \( y_1 \) of the solution is \( 2.0 \times 10^{-7} \). Four orthogonalization points were used during the integration.
Let us conclude this section with a brief discussion of the QR algorithm to reduce
an \( m \times n \) \((m \geq n)\) matrix \( A \) to upper triangular form using orthogonal transformations,
i.e., for finding an \( m \times m \) orthogonal matrix \( Q \) such that
\[
QA = \begin{bmatrix} T \\ 0 \end{bmatrix},
\]
where \( T \) is \( n \times n \).

The procedure can be done in many ways, but we’ll focus on the use of plane reflections
or Householder transformations
\[
H(\omega) = I - 2\frac{\omega\omega^T}{\omega^T\omega}.
\]

Householder transformations satisfy:

1. For all real values of \( \alpha \neq 0 \),

\[
H(\alpha\omega) = H(\omega).
\]

2. \( H \) is symmetric.

3. \( H \) is orthogonal as we easily check

\[
H^T H = H^2 = (I - 2\frac{\omega\omega^T}{\omega^T\omega})(I - 2\frac{\omega\omega^T}{\omega^T\omega})
\]

or

\[
H^T H = I - 4\frac{\omega\omega^T}{\omega^T\omega} + 4\frac{\omega\omega^T\omega\omega^T}{(\omega^T\omega)^2} = I.
\]

4. \( H(\omega)v = v \), if and only if \( v^T\omega = 0 \).

5. \( H(\omega)\omega = -\omega \).

6. Let \( u = v + \omega \) and \( v^T\omega = 0 \), then

\[
H(\omega)(v + \omega) = v - \omega.
\]

Thus, the Householder transformation reflects \( \omega + v \) in a plane through the origin
perpendicular to \( \omega \).
In order to perform the QR reduction (7.2.16), let $a_1$ be the first column of $A$ and choose 

$$\omega = a_1 \pm \lambda_1 e_1$$

where $e_1$ is the first column of the identity matrix and

$$\lambda_1 = \sqrt{a_1^T a_1}.$$

Letting $H^{(1)} = H(\omega)$, we have

$$H^{(1)}A = \begin{bmatrix} \pm \lambda_1 & \times & \times \\ 0 & \times & \times \\ \vdots & \ddots & \times \\ 0 & \times & \times \end{bmatrix} = \begin{bmatrix} \pm \lambda_1 & (v^{(1)})^T \\ 0 & A^{(1)} \end{bmatrix} = A^{(1)}.$$

The same process is repeated on the $(m-1) \times (n-1)$ portion of $A$. Thus, if $a_1^{(1)}$ is the first column of $A^{(1)}$, choose

$$\omega^{(1)} = a_1^{(1)} \pm \lambda_2 e_1^{(1)}$$

where $e_1^{(1)}$ is the first column of the $(m-1) \times (m-1)$ identity matrix. The transformation $H^{(2)}$ is

$$H^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & H(\omega^{(1)}) \end{bmatrix}$$

and the result after an application of $H^{(2)}$ is

$$H^{(2)}A^{(1)} = \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ \vdots & \ddots & \times \end{bmatrix}.$$

After $p = \min(m-1, n)$ steps, we have

$$H^{(p)}H^{(p-1)} \ldots H^{(1)}A = \begin{bmatrix} T \\ 0 \end{bmatrix};$$

thus,

$$Q = H^{(p)}H^{(p-1)} \ldots H^{(1)}$$

and

$$H^{(k)} = \begin{bmatrix} I & 0 \\ 0 & H(\omega^{(k-1)}) \end{bmatrix},$$
where the identity matrix is \((k - 1) \times (k - 1)\).

At each stage of the procedure, the sign of \(\lambda_k\) is chosen so that cancellation is avoided. Thus, if \(a_{11} \geq 0\) choose the positive sign for \(\lambda_1\), and if \(a_{11} < 0\) choose the negative sign. The reduction can be done efficiently in about \(2n^2(m - n/3)\) multiplications plus additions and approximately \(n\) square roots. Techniques to save storage and avoid explicit multiplication to determine \(Q\) are described by Demmel [6].

### 7.3 Multiple Shooting for Nonlinear Problems

We extend the multiple shooting procedures of Section 7.2 to nonlinear problems

\[
y' = f(x, y), \quad a < x < b, \quad g_L(y(a)) = 0, \quad g_R(y(b)) = 0, \tag{7.3.1}
\]

by linearization in function space. Thus, we assume \(\tilde{y}\) to be known and let

\[
y(x) = \tilde{y}(x) + \delta y(x) \tag{7.3.2}
\]

and use a Taylor’s series expansion of \(f\) to obtain

\[
f(x, \tilde{y} + \delta y) = f(x, \tilde{y}) + f_y(x, \tilde{y})\delta y + O(\delta y^2).
\]

Substituting these results into (7.3.1) and neglecting the \(O(\delta y^2)\) terms

\[
y' + \delta y' = f_y(x, \tilde{y})\delta y + f(x, \tilde{y})
\]

or

\[
\delta y' = f_y(x, \tilde{y})\delta y + f(x, \tilde{y}) - y'.
\]

This has the form of a linear system

\[
\delta y' = A(x)\delta y + b(x) \tag{7.3.3a}
\]

with

\[
A(x) = f_y(x, \tilde{y}), \quad b(x) = f(x, \tilde{y}) - y'. \tag{7.3.3b}
\]

Linearizing the left boundary condition yields

\[
g_L(\tilde{y}(a) + \delta y(a)) = g_L(\tilde{y}(a)) + g_L(\tilde{y}(a))\delta y(a) = 0.
\]
This has the form of the linear boundary condition

\[ L \delta y(a) = 1 \]  \hspace{1cm} (7.3.4a)

with

\[ L = g_L(\bar{y}(a)), \quad 1 = -g_L(\bar{y}(a)). \]  \hspace{1cm} (7.3.4b)

At the right end, we have

\[ R \delta y(b) = r \]  \hspace{1cm} (7.3.5a)

with

\[ R = g_R(\bar{y}(b)), \quad r = -g_R(\bar{y}(b)). \]  \hspace{1cm} (7.3.5b)

The linearized system (7.3.3 - 7.3.5) may be solved by iteration using the procedures of Section 7.2; thus,

1. Beginning with an initial guess \( y^{(0)}(x), \ 0 \leq x \leq 1, \)

2. Solve the linear system

\[
\delta y^{(\nu)} = A^{(\nu)}(x) \delta y^{(\nu)} + b^{(\nu)}(x)
\]

\[ L^{(\nu)} \delta y^{(\nu)}(a) = l^{(\nu)}, \quad R^{(\nu)} \delta y^{(\nu)}(b) = r^{(\nu)}, \]

where \( A^{(\nu)}(x) = f_y(x, Y^{(\nu)}), \ etc. \)

3. After each iteration, set

\[ y^{(\nu+1)} = y^{(\nu)} + \delta y^{(\nu)} \]

and repeat the procedure until convergence.

The procedure is awkward since interpolation of \( y^{(\nu)} \) will generally be required to determine \( A^{(\nu)}(x), \ etc. \) at locations that are needed by a variable-step IVP procedure.
Bibliography


