Parallel algorithms for parameter identification in ODEs

Citation for published version (APA):

Document status and date:
Published: 01/01/1996

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 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

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.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 16. May. 2019
Parallel algorithms for parameter identification in odes

by

R.M.M. Mattheij and S.J. Wright
PARALLEL ALGORITHMS FOR PARAMETER IDENTIFICATION IN ODES

R. M. M. MATTHEIJ* AND S. J. WRIGHT†

Abstract. We discuss the solution of parametrized linear difference equations subject to (possibly overdetermined) side conditions. By viewing the problem as a structured quadratic program and using techniques from matrix perturbation theory, we discuss well conditioning of the problem and its consequences for the distribution of the side conditions and parameter coefficients. An efficient, stable, and parallel algorithm is described, together with computational results on current multiprocessor computer architectures.

Key words. ordinary differential equations, matrix perturbations, parallel computing

AMS(MOS) subject classifications. 65L10, 65L20, 65W05

1. Introduction. The linear parameter identification problem in ordinary differential equations consists of finding a smooth function \( x(\cdot) : [0,1] \rightarrow \mathbb{R}^n \) and a parameter vector \( \lambda \in \mathbb{R}^m \) such that the sum of squares

\[
\sum_{i=1}^{p} M_i x(t_i) + Z \lambda - d \quad 0 \leq t_1 < \cdots < t_p \leq 1,
\]

is minimized. The function \( x(\cdot) \) is constrained to satisfy the linear ODE

\[
\dot{x}(t) = A(t)x(t) + F(t)\lambda + f(t), \quad t \in [0,1].
\]

Examples of such problems are discussed by Bock [1], for example. The problem (1.1), (1.2) has a corresponding difference equation form, which we express as

\[
\min_{x_1,\ldots,x_{N+1},\lambda} \left\| \sum_{i=1}^{N+1} M_i x_i + Z \lambda - d \right\|_2^2
\]

subject to

\[
A_i x_i + B_i x_{i+1} + F_i \lambda = f_i, \quad i = 1,\ldots,N,
\]

where \( A_i \) and \( C_i \) are \( n \times n \) and \( F_i \) is \( m \times n \). The discretization can be obtained by applying algorithms such as multiple shooting, finite differencing, and collocation to the continuous form (1.1), (1.2). Note that (1.3), (1.4) is simply an equality-constrained linear least squares problem with a highly structured constraint matrix.

In this paper, we discuss the conditioning of the problem (1.3), (1.4); that is, the sensitivity of its solution \( (x_1, x_2, \ldots, x_{N+1}, \lambda) \) to perturbations in the data. Sufficient conditions for well-conditioning are described in Section 2, and consequences of well-conditioning for the coefficient matrices are discussed in Section 3. The tools that we use in our analysis arise from numerical linear algebra and matrix perturbation theory. Similar analytical tools have been used previously in this context; see, for example, Mattheij [6]. Another approach, based on the structure of the fundamental solution spaces has received much exposure in the literature (de Hoog and Mattheij [3], Mattheij [7]). The latter approach is appealing because it is more closely tied to the difference-equation nature of the problem. However, we concentrate on the linear-algebra-based analysis in

* Department of Mathematics and Computing Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, the Netherlands.
† Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.
this paper, since part of our aim here is to demonstrate that this approach is quite powerful. Naturally, the two approaches are related, as we note in the text.

We describe our parallel algorithm in Section 4 and present computational results in Section 5. The algorithm uses the constraints (1.4) to eliminate subsets of the variables, by using the QR factorization with column pivoting. The algorithm is similar to the cyclic-reduction strategy for two-point boundary value problems described in Wright [11], though its extra features allow it to handle the complications that arise in this more general problem.

The method in this paper is more robust than the one in Mattheij and Wright [8], since it uses globally identifiable quantities to adjust the elimination procedure. Also, we describe the potentially non-dichotomic structure of the solution space, which induces the adjustment process. (do we?)

Throughout the paper we use $\| \cdot \|$ to denote the Euclidean norm of a vector or matrix. The Frobenius norm of a matrix $A \in \mathbb{R}^{t \times s}$ is defined as

$$
\|A\|_F = \left( \sum_{i,j} A_{ij}^2 \right)^{1/2}.
$$

### 2. Well-Conditioning of the Problem.

Here we discuss the conditioning of equality-constrained linear least squares problems; that is, the sensitivity of their solutions to perturbations in the data. For the purposes of this section, we ignore the structure in the formulation (1.3), (1.4) and deal instead with the general structure

$$(2.1) \quad \min_{x} \frac{1}{2} \| M x - f \|_2^2 \quad \text{subject to} \quad W^T x = b,$$

where $W \in \mathbb{R}^{t \times s}$, $M \in \mathbb{R}^{m \times t}$, and so on. We assume that the data is scaled so that none of the elements are too large. To be specific, we assume that $\| M f \|$ and $\| W b \|$ are both 1.

We start by defining a little terminology and notation. The singular value decomposition (SVD) of a matrix $A \in \mathbb{R}^{t \times s}$ with $t \geq s$ is given by

$$
A = U \begin{bmatrix} S & 0 \\ 0 & \end{bmatrix} V^T = \sum_{i=1}^{s} \sigma_i u_i v_i^T.
$$

Here, $U = [ u_1 \ u_2 \ \ldots \ u_t ]$ and $V = [ v_1 \ v_2 \ \ldots \ v_s ]$ are orthogonal with dimensions $t$ and $s$, respectively, and $S = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s)$, where the $\sigma_i$ are the singular values of $A$ arranged in descending order; that is, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_s \geq 0$. Note that $\| A \| = \sigma_1$ and the condition number of $A$ is $\kappa(A) = \sigma_1 / \sigma_s$. We say that the matrix $A$ is well-conditioned if $\kappa(A)$ is not too large.

The condition for $x$ to be a solution of (2.1) is that there exists a Lagrange multiplier vector $z \in \mathbb{R}^t$ such that

$$(2.2) \quad \begin{bmatrix} M^T M & W \\ W^T & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} M^T f \\ b \end{bmatrix}.$$

We say that (2.1) is a well-conditioned problem if its solution $x$ is not too sensitive to perturbations in the data $M$, $W$, $f$, and $b$. We can make this condition explicit by assuming that $W$ has full rank and defining an orthonormal matrix $Y \in \mathbb{R}^{t \times (t-s)}$ that spans the null space of $W^T$; that is, $W^T Y = 0$ and $Y^T Y = I$. By decomposing $x$ as

$$
x = x_w + w_y = W \tilde{x}_w + Y \tilde{x}_y,$$

we can rewrite (2.2) as

$$(2.3) \quad \begin{bmatrix} Y^T M^T M Y & Y^T M^T M W \\ W^T M^T M Y & W^T M^T M W \end{bmatrix} \begin{bmatrix} \tilde{x}_y \\ \tilde{x}_w \\ z \end{bmatrix} = \begin{bmatrix} Y^T M^T f \\ W^T M^T f \end{bmatrix}.$$
By denoting the SVDs of \( W \) and \( MY \) by

\[
W = \sum_{i=1}^{s} \sigma_i(W) u_i v_i^T, \quad MY = \sum_{i=1}^{r-s} \sigma_i(MY) u_i v_i^T,
\]

we have from (2.3) that

\[
(2.4) \quad x_w = W \tilde{x}_w = W(W^T W)^{-1} b = \sum_{i=1}^{s} \frac{v_i^T b}{\sigma_i(W)} u_i
\]

and

\[
(2.5) \quad \tilde{x}_y = \sum_{i=1}^{r-s} \frac{v_i^T (f - M x_w)}{\sigma_i(MY)} u_i.
\]

It follows from (2.4) and (2.5) that \( x \) is not too sensitive to perturbations in \( f, b, M, \) and \( w \) provided that the following assumption is true:

**Assumption 1.** The matrices \( W \) and \( MY \) are well conditioned.

This assumption is assumed to hold for the rest of the paper. Since we have assumed that \( W \) and \( M \) (and hence \( MY \)) are scaled so that their largest elements are around 1, we have that \( \sigma_1(W) \) and \( \sigma_1(MY) \) are also similar to 1. Hence, Assumption 1 implies that the smallest singular values \( \sigma_s(W) \) and \( \sigma_{r-s}(MY) \) are not too small.

As an aside, we note how this notion of well conditioning naturally extends existing theory for the case of two-point boundary value problems and ODEs with multipoint side conditions. Consider the following discretized multipoint problem:

\[
(2.6) \quad \sum_{i=1}^{N+1} M_i x_i = d \quad A_i x_i + B_i x_{i+1} = f_i, \quad i = 1, 2, \ldots, N,
\]

which can be expressed in matrix form as

\[
(2.7) \quad \begin{bmatrix}
  A_1 & B_1 \\
  A_2 & B_2 \\
  \vdots & \vdots \\
  A_N & B_N \\
  M_1 & M_2 & \cdots & M_N & M_{N+1}
\end{bmatrix}
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N \\
  x_{N+1}
\end{bmatrix} = \begin{bmatrix}
  f_1 \\
  f_2 \\
  \vdots \\
  f_N \\
  d
\end{bmatrix},
\]

where all blocks \( A_i, B_i, \) and \( M_i \) are \( n \times n \). The problem is usually said to be well conditioned if the coefficient matrix in (2.7) is well conditioned. As part of their solution procedure, some algorithms find a fundamental solution \( \{\Phi_i\}_{i=1}^{N+1} \) and a particular solution \( \{\tilde{x}_i\}_{i=1}^{N+1} \) satisfying

\[
\Phi_i \in \mathbb{R}^{n \times n}, \quad A_i \Phi_i + B_i \Phi_{i+1} = 0, \quad i = 1, 2, \ldots, N,
\]

\[
\tilde{x}_i \in \mathbb{R}^n, \quad A_i \tilde{x}_i + B_i \tilde{x}_{i+1} = f_i, \quad i = 1, 2, \ldots, N.
\]

(Both solutions are usually found by some stable procedure that involves decoupling of the increasing and decreasing fundamental modes.) The actual solution can then be expressed as \( x_i = \Phi_i z + \tilde{x}_i \), where the vector \( z \in \mathbb{R}^n \) can be found by applying the side conditions and solving the system

\[
(2.8) \quad \left( \sum_{i=1}^{N+1} M_i \Phi_i \right) z = d - \sum_{i=1}^{N+1} M_i \tilde{x}_i.
\]
It is a known consequence of well-conditioning of (2.7) and stable calculation of $\Phi_i$ that the $n \times n$ coefficient matrix in (2.8) is well conditioned. We can see how this relates to our standing assumption by making the identifications

$$W^T = \begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \\ \vdots & \vdots \\ A_N & B_N \end{bmatrix}, \quad M = \begin{bmatrix} M_1 & M_2 & \cdots & M_N & M_{N+1} \end{bmatrix}, \quad Y = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_{N+1} \end{bmatrix}.$$  

(The $Y$ defined here is not orthonormal in general, but the stable way in which it is calculated usually ensures that it spans the null space of $W^T$ and is well conditioned.) Note that the well-conditioned coefficient matrix in (2.8) is simply $MY$, while well conditioning of the coefficient matrix in (2.7) certainly implies well conditioning of its row submatrix $W$, as we show in the next section. Hence, we have shown that Assumption 1 is consistent with the standard notion of well conditioning for the multipoint problem (2.6).

In our problem (1.3), (1.4), we have

$$M = \begin{bmatrix} M_1 & M_2 & \cdots & M_{N+1} & Z \end{bmatrix}, \quad W = \begin{bmatrix} A_1 & B_1 & F_1 \\ A_2 & B_2 & F_2 \\ \vdots & \vdots & \vdots \\ A_N & B_N & F_N \end{bmatrix}.$$  

For the remainder of the paper, we use $M$ and $W$ to refer to these two specific matrices.

3. Some Consequences of Well-Conditioning. The well-conditioning assumption has some interesting consequences for the properties of certain submatrices of $M$ and $W$. Since we are concerned as much with computational experience in this paper as with theoretical issues, we prefer to state and prove our results in terms of numerical rank rather than exact rank. We start by distinguishing these two notions of rank.

Matrices are rank deficient when some of their singular values are zero. Numerically rank deficient matrices are not as easy to define. We could take the numerical rank to be the number of singular values that are larger than some small prescribed constant $\epsilon$, where $\epsilon$ is of the order of unit roundoff $u$ or its square root, possibly scaled by the largest singular value of the matrix. This definition may not always be satisfying, however. In the case of a continuum of singular values between $u$ and 1, for instance, there is no obvious place to draw the line between the "morally zero" values and the "significantly nonzero" values. For our purposes, we assume all the matrices we deal with show a clear separation between the two classes of singular values. This assumption is reasonable since matrices that arise in this and other applications are often "structurally" rank deficient; that is, tiny nonzero singular values arise only from roundoff errors incurred in forming the matrix or in storing its elements on a finite precision computer.

Given the small quantity that distinguishes large singular values from small ones, we write $\sigma_i = \delta_i$ if the ratio $\sigma_i/\epsilon$ is bounded by a moderate constant and $\sigma_i \ll 1$. We also occasionally use $\delta_i$ to denote a matrix or vector whose norm is of the order of $\epsilon$. Using this notation, we say that a $t \times s$ matrix $C$ has numerical rank $r$ if

$$\sigma_1(C) \geq \sigma_2(C) \geq \cdots \geq \sigma_r(C) \gg \epsilon$$  

while

$$\sigma_i(C) = \delta_i, \quad i = r + 1, \ldots, \min(t, s).$$  

In the remainder of the paper, we use $\text{nr}(\cdot)$ to denote numerical rank.

We start with some technical results from numerical linear algebra and matrix perturbation theory. The first result describes the effect of a rank-one update on the eigenvalues of a symmetric matrix.
Theorem 3.1. Let $G$ and $H$ be $m \times m$ symmetric matrices with $H = G + \tau cc^T$, where $c$ is a vector with $\|c\|_2 = 1$. Then

\begin{equation}
\lambda_i(H) = \lambda_i(G) + \mu_i \tau, \quad i = 1, 2, \ldots, m,
\end{equation}

where the $\mu_i$ satisfy

\begin{equation}
\sum_{i=1}^m \mu_i = 1, \quad \mu_i \geq 0, \quad i = 1, 2, \ldots, m.
\end{equation}

If $\tau > 0$, the following interlacing property holds:

\begin{equation}
\lambda_1(H) \geq \lambda_1(G) \geq \lambda_2(H) \geq \cdots \geq \lambda_m(H) \geq \lambda_m(G).
\end{equation}

These results are well known; Wilkinson [10] contains proofs of both. See [10, pp. 94–97] for a proof of (3.1) and [10, pp. 103–104] for a proof of (3.2). A slightly different (and more general) interlacing result also proves to be useful.

Theorem 3.2. Let $\bar{C} \in \mathbb{R}^{m \times s}$ be a column submatrix of $C \in \mathbb{R}^{m \times s}$ with $0 < \bar{s} < s$. Then the singular values $\sigma_i$ of $\bar{C}$ are related to the singular values $\sigma_i$ of $C$ by

\begin{equation}
\sigma_i \geq \sigma_{i+1}(s-\bar{s}), \quad i = 1, 2, \ldots, \bar{s}.
\end{equation}

Proof. Without loss of generality, suppose that $\bar{C}$ consists of the first $\bar{s}$ columns of $C$, so that we can partition $C$ as

\[ C = \begin{bmatrix} \bar{C} & \hat{C} \end{bmatrix}. \]

Note that $\bar{C}^T \bar{C}$ is the leading principal $\bar{s} \times \bar{s}$ submatrix of $C^T C$. Hence, by Corollary IV.4.3 of Stewart and Sun [9], we have

\[ \lambda_i(C^T C) \geq \lambda_i(\bar{C}^T \bar{C}) \geq \lambda_i(s-\bar{s})(\bar{C}^T \bar{C}) \quad \text{for any matrix } M. \]

The result follows immediately from the fact that $\sigma_i(M) = \lambda_i(M^T M)^{1/2}$ for any matrix $M$.

A third result relates specifically to numerical rank; its trivial proof is omitted.

Lemma 3.3. Let $\bar{C}$ and $\hat{C}$ be two matrices with the same number of rows. Then

\[ \text{urank}\left[ \begin{bmatrix} \bar{C} & \hat{C} \end{bmatrix} \right] \leq \text{urank}(\bar{C}) + \text{urank}(\hat{C}). \]

Our final technical result is simple to state, but its proof is somewhat complicated and is relegated to the appendix.

Theorem 3.4. Let $A$ and $\bar{A}$ be two $t \times s$ matrices with $0 < s \leq t$ and full rank $s$, and denote their ordered singular values by $\sigma_1, \sigma_2, \ldots, \sigma_s$ and $\bar{\sigma}_1, \bar{\sigma}_2, \ldots, \bar{\sigma}_s$, respectively. Let $\bar{Y}$ be any $t \times (t-s)$ orthonormal matrix whose columns span the null space of $\bar{A}^T$. Then we can define an orthonormal matrix $Y$ whose columns span the null space of $A^T$ in such a way that

\[ \|Y - \bar{Y}\|_F \leq \frac{2}{\sigma_s} \|A - \bar{A}\|_F. \]
Numerical rank deficiency in the leading square principal submatrices of $W$ indicates the presence of decreasing fundamental modes. If the problem is well conditioned, it is known that such modes must be regulated by initial conditions. The amount of rank deficiency is directly related to the number of decreasing modes which, in turn, is related to the amount of regulation we expect from the leading part of the coefficient matrix $M$. The following result makes these observations more explicit.

**Theorem 3.5.** Suppose that some leading $In \times In$ submatrix of $W$ $(1 \leq I \leq N)$ is numerically rank deficient, with numerical rank $In - r$ for some $r$ with $1 \leq r < n$. Then the numerical rank of the partial coefficient matrix $\tilde{M}_I$ defined by

$$\tilde{M}_I = \begin{bmatrix} M_1 & M_2 & \cdots & M_I \end{bmatrix}$$

is at least $r$.

**Proof.** We start by defining some notation. Let

$$W = \begin{bmatrix} W_{I1} & W_{I2} \\ 0 & W_{I3} \end{bmatrix},$$

where

$$W_{I1} = \begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \\ \ddots & \ddots \\ A_{I-1} & B_{I-1} \\ A_I & B_I \end{bmatrix},$$

with $W_{I2}$ and $W_{I3}$ defined accordingly. Also, define

$$\tilde{M}_I = \begin{bmatrix} M_{I+1} & M_{I+2} & \cdots & M_{N+1} & Z \end{bmatrix},$$

so that the total coefficient matrix $M$ is partitioned as $M = [\tilde{M}_I | \tilde{M}_I]$. By our assumptions on $W_{I1}$ we have

$$\sigma_i(W_{I1}) > \epsilon, \quad i = 1, 2, \ldots, In - r,$$

$$\sigma_i(W_{I1}) = \delta, \quad i = In - r + 1, \ldots, In,$$

and we can write the SVD of $W_{I1}$ as

$$W_{I1} = \sum_{i=1}^{In} \sigma_i(W_{I1}) u_i v_i^T,$$

where $U = [u_1 \mid u_2 \mid \ldots \mid u_{In}]$ and $V = [v_1 \mid v_2 \mid \ldots \mid v_{In}]$ are both orthogonal matrices in $R^{In \times In}$. Now, define the purified matrix $\tilde{W}_{I1}$ by

$$\tilde{W}_{I1} = \sum_{i=1}^{In-r} \sigma_i(W_{I1}) u_i v_i^T.$$

Accordingly, define $\tilde{W}$ to be a perturbation of (3.4) in which $\tilde{W}_{I1}$ takes the place of $W_{I1}$; that is,

$$\tilde{W} = \begin{bmatrix} \tilde{W}_{I1} & W_{I2} \\ 0 & W_{I3} \end{bmatrix}.$$

Since $W$ is well conditioned and

$$||W - \tilde{W}||_F = \left\| \sum_{i=1}^{n} \sigma_i(W_{I1}) u_i v_i^T \right\|_F = \sum_{i=In-r+1}^{In} \sigma_i(W_{I1}) = \delta,$$
then $\tilde{W}$ is also well conditioned.

We now construct an orthonormal matrix $\tilde{Y}$ of dimension $((N + 1)n + p) \times (n + p)$ whose columns span the null space of $\tilde{W}$. From (3.5) and (3.6), we have

$$\tilde{W}_{i1} v_i = 0, \quad i = In - r + 1, \ldots, In.$$  

It follows that

$$\tilde{W} \begin{bmatrix} v_i \\ 0 \end{bmatrix} = \begin{bmatrix} \tilde{W}_{i1} \\ 0 \end{bmatrix} = \begin{bmatrix} v_i \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$  

and we can choose the first $r$ columns of $\tilde{Y}$ to be

$$\begin{bmatrix} v_i \\ 0 \end{bmatrix}, \quad i = In - r + 1, \ldots, In.$$  

The structure of the remaining $(n + p - r)$ columns of $\tilde{Y}$ does not particularly concern us, so we write

$$\tilde{Y} = \begin{bmatrix} \tilde{Y}_{11} & \tilde{Y}_{12} \\ 0 & \tilde{Y}_{22} \end{bmatrix},$$  

where

$$\tilde{Y}_{11} = [v_{In-r+1} | \ldots | v_{In}] \in \mathbb{R}^{In \times r},$$  

and so on. By Theorem 3.4, we can choose a matrix $Y$ whose columns span the nullspace of $W$ in such a way that

$$||Y - \tilde{Y}||_F \leq \frac{2}{\sigma_{In-r}(W_{11})} ||W - \tilde{W}||_F.$$  

Because of (3.7) and since, by our definition of numerical rank, we have $\sigma_{In-r}(W_{11}) \sim 1$, we conclude from (3.8) that

$$||Y - \tilde{Y}||_F = \delta.$$

Therefore $Y$ has the form

$$Y = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} \tilde{Y}_{11} + \delta \varepsilon & \tilde{Y}_{12} + \delta \varepsilon \\ \delta \varepsilon & \tilde{Y}_{22} + \delta \varepsilon \end{bmatrix}.$$  

Now, consider the matrix $\times (n + p)$ matrix $MY$, which is well conditioned by our prevailing assumptions. We have

$$MY = \begin{bmatrix} \tilde{M}_{11} & \tilde{M}_{12} \\ \tilde{M}_{21} & \tilde{M}_{22} \end{bmatrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} = \begin{bmatrix} \tilde{M}_{11} \tilde{Y}_{11} + \delta \varepsilon & \tilde{M}_{11} \tilde{Y}_{12} + \delta \varepsilon \\ \tilde{M}_{21} \tilde{Y}_{21} + \delta \varepsilon & \tilde{M}_{22} \tilde{Y}_{22} + \delta \varepsilon \end{bmatrix}. $$

If $\operatorname{rank}(\tilde{M}_{11}) < r$, then $\operatorname{rank}(\tilde{M}_{11} \tilde{Y}_{11} + \delta \varepsilon) < r$ also, and it follows from Theorem 3.1 that $\operatorname{rank}(MY) < n + p$. This assertion contradicts well-conditioning, so we must have $\operatorname{rank}(\tilde{M}_{11}) \geq r$, and the proof is complete. □

The following corollary is immediate.

**Corollary 3.6.** Suppose that some diagonal block $A_I$ has numerical rank $n - r$ for some $r$ with $1 \leq r \leq n$. Then the numerical rank of the matrix $\tilde{M}_I$ defined in Theorem 3.5 is at least $r$.

A similar result for the above-diagonal blocks $B_I$ rests on a theorem similar to Theorem 3.5 applied to a different part of the matrix $W$.
**Corollary 3.7.** Suppose that some above-diagonal block $B_I$ has numerical rank $n - r$ for some $r$ with $1 \leq r \leq n$. Then the numerical rank of the partial coefficient matrix $\tilde{M}_I$ defined by

$$
\tilde{M}_I = \begin{bmatrix}
M_{I+1} & M_{I+2} & \cdots & M_{N+1}
\end{bmatrix}
$$

is at least $r$.

**Proof.** Consider the following square submatrix of $W$, whose dimension is $(N - I + 1)n$:

$$
W_{I4} = \begin{bmatrix}
B_I & A_{I+1} & B_{I+1} & \cdots & \cdots & A_{N+1} & B_{N+1}
\end{bmatrix}
$$

Because $\text{null}(B_I) = n - r$, we have that $W_{I4}$ is numerically rank deficient with $\text{null}(W_{I4}) \leq (N - I + 1)n - r$. Since there are no other nonzeros in the columns of $W$ occupied by $W_{I4}$, we can apply the reasoning of Theorem 3.5 to deduce that the corresponding part of $M$—the matrix $M_I$, in fact—has numerical rank at least $r$. $\Box$

We can also state a result about the parameter components $F_{ij}$ of the matrix in (1.4).

**Theorem 3.8.** Suppose there are indices $I$ and $J$ with $1 \leq I \leq J \leq N$ such that $\text{null}(A_I) = n - r_I$ and $\text{null}(B_J) = n - r_J$ with $r_I + r_J > n$. Then the numerical rank of the partial parameter coefficient matrix defined by

$$
F_{IJ} = \begin{bmatrix}
F_I \\
\vdots \\
F_J
\end{bmatrix}
$$

is at least $r_I + r_J - n$.

**Proof.** Consider the following row submatrix of $W$, whose dimension is $(J - I + 1)n \times (J - I + 2)n + p$:

$$
W_{IJ} = \begin{bmatrix}
A_I & B_I & F_I \\
\vdots & \ddots & \vdots \\
A_J & B_J & F_J
\end{bmatrix}
$$

(The zero columns have been omitted since they do not enter into the argument.) Suppose for contradiction that $\text{null}(F_{IJ}) < r_I + r_J - n$. From Lemma 3.3, we have

$$
\text{null}(W_{IJ}) \leq \text{null}\left(\begin{bmatrix}A_I \\ 0\end{bmatrix}\right) + \text{null}\left(\begin{bmatrix}0 \\ B_J\end{bmatrix}\right) + \text{null}\left(\begin{bmatrix}B_I & A_{I+1} & B_{I+1} & \cdots \\ A_J & \cdots & \cdots & \cdots
\end{bmatrix}\right) + \text{null}(F_{IJ})
$$

$$
< (n - r_I) + (n - r_J) + (J - I)n + (r_I + r_J - n) = (J - I + 1)n.
$$

Hence $W_{IJ}$ is numerically rank deficient and so, by Lemma 3.3, $W$ is also numerically rank deficient. But this assertion contradicts well conditioning of $W$, so we must have $\text{null}(F_{IJ}) \geq r_I + r_J - n$. $\Box$

Finally, we state a simple result about the numerical rank of the column submatrix of $W$ that our algorithm actually operates on.

**Theorem 3.9.** Consider the following column submatrix of $W$, whose dimensions are $Nn \times (N - 1)n$:

$$
W = \begin{bmatrix}
B_1 \\
A_2 & B_2 \\
\vdots & \ddots & \ddots \\
A_{N-1} & B_{N-1} & A_N
\end{bmatrix} \in \mathbb{R}^{Nn \times (N - 1)n}.
$$
Then the numerical rank of $\bar{W}$ is at least $(N - 2)n - p$.

Proof. Trivial consequence of Lemma 3.3. \(\square\)

4. The Algorithm. To describe the algorithm we consider a slightly generalized form of the problem (1.3),(1.4) in which we add the auxiliary constraints

$$
C_i x_i + D_i x_{i+1} + E_i \lambda = g_i, \quad i = 1, \ldots, N,
$$

where $C_i$, $D_i$, $E_i$, and $g_i$ all have $n_i$ rows. These constraints have the same form as the constraints (1.4) arising from the ODE, except that the number of rows may vary with the stage. They do not usually arise in the problem as originally posed, but they may appear as a by-product of the recursive reduction process used by the algorithm. Essentially, the algorithm proceeds by reducing the problem (1.3),(1.4),(4.1) to a smaller problem by eliminating approximately half the stages $x_i$, $i = 1, \ldots, N + 1$. The reduced problem has exactly the same form (1.3),(1.4),(4.1) but about half as many stages. This process continues recursively until a problem with just two stages (that is, $N = 1$) is obtained. This problem is solved by using standard linear least-squares techniques, and the reduction process is reversed to recover the stages that were eliminated earlier in the process.

The algorithm can be viewed as a generalization of the cyclic-reduction variant of the structured-QR approach described in Wright [11] for two-point boundary value problems. The enhancements we describe here allow the algorithm to deal with possible rank-deficiency in the columns of the constraint matrix of (1.4). The theory of Section 2 places a reasonable upper bound on the amount of rank deficiency we can expect to encounter during the elimination process, so proper handling of this feature adds to the robustness of the algorithm without greatly increasing its running time.

4.1. The Simple Case of No Rank Deficiency. The basic step in the algorithm is the elimination of all or part of a stage vector $x_{i+1}$ by expressing it in terms of its neighboring stage vectors and the parameter vector $\lambda$. We start by describing this process for the case in which the problem has no auxiliary constraints (4.1) and near-rank-deficiency does not exist in the columns of the constraint matrix in (1.4) that correspond to $x_{i+1}$. In this case, all components of $x_{i+1}$ can be eliminated. By extracting the two equations that involve $x_{i+1}$ from the constraint system (1.4), we obtain the subsystem

$$
\begin{bmatrix}
A_i & B_i & 0 & F_i \\
0 & A_{i+1} & B_{i+1} & F_{i+1}
\end{bmatrix}
\begin{bmatrix}
x_i \\
x_{i+1}
\end{bmatrix}
= 
\begin{bmatrix}
f_i \\
f_{i+1}
\end{bmatrix}.
$$

By applying a QR factorization, using either Householder transformations or Givens rotations, we obtain

$$
Q_i \begin{bmatrix} B_i \\ A_{i+1} \end{bmatrix} = \begin{bmatrix} R_i \\ 0 \end{bmatrix},
$$

where $Q_i$ is $2n \times 2n$ orthogonal and $R_i$ is $n \times n$ upper triangular. Applying the transformation $Q_i$ to all columns of (4.2), we obtain the equivalent system

$$
\begin{bmatrix}
G_i & R_i & J_i & H_i \\
A_i & 0 & B_i & F_i
\end{bmatrix}
\begin{bmatrix}
x_i \\
x_{i+1}
\end{bmatrix}
= 
\begin{bmatrix}
h_i \\
f_i
\end{bmatrix}.
$$

By our assumption that $R_i$ has full rank (and is, we hope, well conditioned), we can eliminate $x_{i+1}$ by using the first block row in (4.4) and write

$$
x_{i+1} = R_i^{-1} [h_i - G_i x_i - J_i x_{i+2} - H_i \lambda].
$$

A reduced constraint system can be obtained by extracting the second block row from (4.4). When this reduced system is solved for $x_i$, $x_{i+2}$, and $\lambda$, the eliminated stage $x_{i+1}$ can be recovered by a back-substitution
with the matrix $R_i$, according to the formula (4.5). We can also use (4.5) to eliminate $x_{i+1}$ from the objective function (1.3). The expression

$$\cdots + M_i x_i + M_{i+1} x_{i+1} + M_{i+2} x_{i+2} + \cdots + Z \lambda - d$$

is transformed to

$$\cdots + \tilde{M}_i x_i + \tilde{M}_{i+2} x_{i+2} + \cdots + \tilde{Z} \lambda - \tilde{d},$$

where

$$\begin{align*}
(4.8a) & & \tilde{M}_i & = M_i - M_{i+1} R_i^{-1} G_i \\
(4.8b) & & \tilde{M}_{i+2} & = M_{i+2} - M_{i+1} R_i^{-1} J_i \\
(4.8c) & & \tilde{Z} & = Z - M_{i+1} R_i^{-1} H_i \\
(4.8d) & & \tilde{d} & = d - M_{i+1} R_i^{-1} h_i.
\end{align*}$$

The process above can also be viewed as a squeezing of the two block rows in (4.2) into the single row formed by the second equation in (4.4). It can be carried out simultaneously for many stages in the original system (1.4), provided that no two stages that we are trying to eliminate are adjacent to each other. The update formulae (4.8) will need to be changed when, for example, $x_{i-1}, x_{i+1},$ and $x_{i+2}$ are all being eliminated simultaneously. If we define $R_{i-2}, R_{i+2},$ etc, by replacing $i$ by $i - 2$ and $i + 2$ in (4.4) and (4.5), we obtain

$$\begin{align*}
(4.9a) & & \tilde{M}_i & = M_i - M_{i+1} R_i^{-1} G_i - M_{i-1} R_i^{-1} J_{i-2}, \\
(4.9b) & & \tilde{M}_{i+2} & = M_{i+2} - M_{i+1} R_i^{-1} J_i - M_{i+3} R_i^{-1} G_{i+2}, \\
(4.9c) & & \tilde{Z} & = Z - M_{i+1} R_i^{-1} H_i - M_{i-1} R_i^{-1} H_{i-2} - M_{i+3} R_i^{-1} H_{i+2}, \\
(4.9d) & & \tilde{d} & = d - M_{i+1} R_i^{-1} h_i - M_{i-1} R_i^{-1} h_{i-2} - M_{i+3} R_i^{-1} h_{i+2}.
\end{align*}$$

Our approach is to eliminate all the even-index stages $x_2, x_4, x_6,$ and to obtain a reduced system in the odd-index variables. If $N$ is even, it is easy to see that the reduced system formed by (4.7) and the second block row of (4.4) has exactly the form of (1.3),(1.4) where the unknowns are $x_1, x_3, x_5, \ldots, x_{N+1}$ and $\lambda$. Formally, if we relabel the components for the reduced system as

$$\begin{align*}
\hat{A}_i & \overset{\text{def}}{=} \hat{A}_{2i-1}, & \hat{B}_i & \overset{\text{def}}{=} \hat{B}_{2i-1}, & \hat{F}_i & \overset{\text{def}}{=} \hat{F}_{2i-1}, \\
\hat{f}_i & \overset{\text{def}}{=} \hat{f}_{2i-1}, & \hat{M}_i & \overset{\text{def}}{=} \hat{M}_{2i-1}, & \hat{x}_i & \overset{\text{def}}{=} x_{2i-1}, \\
\hat{Z} & \overset{\text{def}}{=} \hat{Z}, & \hat{d} & \overset{\text{def}}{=} \hat{d}, & \hat{N} & = N/2,
\end{align*}$$

we obtain the following reduced problem:

$$\begin{align*}
(4.10) & & \min_{\hat{x}_1, \ldots, \hat{x}_{\hat{N}+1}, \hat{\lambda}} & \left\| \sum_{i=1}^{\hat{N}+1} \hat{M}_i \hat{x}_i + \hat{Z} \hat{\lambda} - \hat{d} \right\|_2^2, \\
\text{subject to} & & \hat{A}_i \hat{x}_i + \hat{B}_i \hat{x}_{i+1} + \hat{F}_i \hat{\lambda} = \hat{f}_i, & i = 1, \ldots, \hat{N},
\end{align*}$$

where $\hat{A}_i$ and $\hat{B}_i$ are $n \times n$ and $\hat{F}_i$ is $m \times n$.

The elimination procedure can now be applied in a recursive fashion to the reduced system (4.10), (4.11) to obtain an even more compact system. Eventually, we obtain a system with just two stages ($N = 1$), which
we write as

\begin{align}
(4.12a) \quad & \min_{x_1, x_2, \lambda} \left\| \begin{bmatrix} M_1 & M_2 & Z \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda \end{bmatrix} - d \right\|_2^2, \\
(4.12b) \quad & \begin{bmatrix} A_1 & B_1 & F_1 \\ A_2 & B_2 & F_2 \\ \vdots & \vdots & \vdots \\ A_N & B_N & F_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \lambda \end{bmatrix} = f_1.
\end{align}

This system can be solved directly by using standard null-space algorithms for equality-constrained linear least squares problems. When \( N \) is a power of 2, our algorithm can be expressed informally as the following recursive routine.

\begin{algorithm}
compress(N, A_i, B_i, F_i, f_i, M_i, Z, d, x_i, \lambda)
\begin{algorithmic}
\If {N = 1}
\State solve root problem (4.12) for \((x_1, x_2, \lambda)\);
\State return \(x_1, x_2, \lambda\);
\Else
\State eliminate \(x_2, x_4, \ldots, x_N\) to construct the reduced system with data \(\tilde{A}_i, \tilde{B}_i, \tilde{M}_i\), etc., and unknowns \(\tilde{x}_i\) and \(\lambda\);
\State \(\text{compress}(N/2, \tilde{A}_i, \tilde{B}_i, \tilde{F}_i, \tilde{f}_i, \tilde{M}_i, \tilde{Z}, \tilde{d}, \tilde{x}_i, \lambda)\);
\State set \(x_{2i-1} = \tilde{x}_i, i = 1, \ldots, N/2 + 1\); recover \(x_2, x_4, \ldots, x_N\) from (4.5);
\State return \(x_1, x_2, \ldots, x_{N+1}, \lambda\);
\EndIf
\end{algorithmic}
\end{algorithm}

4.2. The General Case. Note again that the validity of the algorithm above rests on the invertibility of the matrices \(R_i\) that arise whenever we try to eliminate a stage. If the matrix formed by the constraints (1.4) had full column rank, then the submatrix in (4.3) also has full column rank, so \(R_i\) is nonsingular. In general, however, the constraint system (1.4), (4.1) may not have a coefficient matrix with full column rank. Hence, we use the term “full rank” to refer to “full row rank” in this context, as we now state formally.

**Definition 1.** The system of constraints (1.4), (4.1) is said to have full rank if their coefficient matrix, namely

\[
\begin{bmatrix}
A_1 & B_1 & F_1 \\
A_2 & B_2 & F_2 \\
\vdots & \vdots & \vdots \\
A_N & B_N & F_N \\
C_1 & D_1 & E_1 \\
C_2 & D_2 & E_2 \\
\vdots & \vdots & \vdots \\
C_N & D_N & E_N
\end{bmatrix}
\]

has full row rank.

The basic modification we make to the algorithm to handle the more general case is to introduce column pivoting and rank detection into the QR factorization step (4.3). We assume that Householder QR factorization is used, and that before stage \(i\) of the process (in which all the subdiagonal elements in column \(i\) are eliminated), the column with largest remaining 2-norm among columns \(i\) through \(n\) is swapped into position \(i\). The process is described in detail by Golub and Van Loan [4, Section 5.4.1]. The result of this strategy is
that the upper triangular factor $R_i$ has the following properties:

$$
|(R_i)_{jj}| \geq |(R_i)_{j+1,j+1}|, \quad j = 1, \ldots, n - 1,
$$

$$
|(R_i)_{jj}| \geq |(R_i)_{jk}|, \quad k = j + 1, \ldots, n, \quad j = 1, \ldots, n - 1.
$$

If $R_i$ is rank deficient, its last few diagonals will be zero. We say that $R_i$ has nullity $n_i$ if its first $n - n_i$ diagonals are nonzero and the last $n_i$ are zero. Because of this property, the $n_i \times n_i$ submatrix in the lower right corner of $R_i$ is zero.

We now work through the elimination of components of $x_{i+1}$ for the case of $R_i$ rank deficient. Suppose that we have a problem defined by (1.3), (1.4), and the auxiliary constraints (4.1). We introduce column pivoting into the QR factorization process (4.3) to yield an orthogonal $Q_i$, and upper triangular $R_i$, and an $n \times n$ permutation matrix $\Pi_i$ such that

$$
Q_i \left[ \begin{array}{c|c} B_i & A_{i+1} \end{array} \right] \Pi_i = \left[ \begin{array}{c|c} R_{i1} & R_{i2} \\ \hline 0 & 0 \end{array} \right],
$$

where $R_{i1}$ has dimension $(n-n_i) \times (n-n_i)$ and $R_{i2}$ has dimension $n_i \times n_i$. We choose $n_i$ so that $(n-n_i)$ is the approximate column rank of the matrix in (4.14). In exact arithmetic, $n_i$ would be the largest nonnegative integer that makes $R_{i2}$ a zero matrix. We do not require $R_{i2}$ to be exactly zero, however, since it is desirable for reasons of stability to treat nearly rank-deficient matrices (for which $R_{i2}$ tends to be "small") in the same way as exactly-rank-deficient matrices. When the orthogonal matrix $Q_i$ is applied to the whole subsystem (4.2), we obtain the following transformed system.

$$
\begin{bmatrix}
G_i^1 & R_{i1}^1 & R_{i2}^1 & J_i^1 & H_i^1 \\
G_i^2 & 0 & R_{i2}^2 & J_i^2 & H_i^2 \\
A_i & 0 & 0 & B_i & F_i \\
\end{bmatrix}
\begin{bmatrix}
x_i \\
(\Pi_i^T x_{i+1})^1 \\
(\Pi_i^T x_{i+1})^2 \\
\end{bmatrix} =
\begin{bmatrix}
h_i^1 \\
h_i^2 \\
\lambda \\
\end{bmatrix}.
$$

Note that $G_i$, $J_i$, $H_i$, and $h_i$ have been row-partitioned according to the value of $n_i$ detected during the factorization (4.14).

We use the first block of (4.15) to eliminate the $(n-n_i)$-dimensional subvector $(\Pi_i^T x_{i+1})^1$ because, by invertibility of $R_{i1}^1$, we can write

$$
(\Pi_i^T x_{i+1})^1 = (R_{i1}^1)^{-1} \left[ h_i^1 - G_i^1 x_i - R_{i2}^1 (\Pi_i^T x_{i+1})^2 - J_i^1 x_{i+2} - H_i^1 \lambda \right].
$$

To preserve the regular structure of the constraints (1.4), in which each of the coefficient blocks $A_i$ and $B_i$ is square, we transfer the remaining subvector $(\Pi_i^T x_{i+1})^2$ to the parameter vector. That is, we define a new parameter vector $\lambda$ as

$$
\lambda = \begin{bmatrix} \lambda \\ (\Pi_i^T x_{i+1})^2 \end{bmatrix}.
$$

After this elimination and reclassification, the expression (4.6) is transformed to

$$
\cdots + \tilde{M}_i x_i + \tilde{M}_{i+2} x_{i+2} + \cdots + \tilde{Z} \lambda - \tilde{d},
$$

where

$$
\begin{align}
\tilde{M}_i &= M_i - (M_{i+1} \Pi_i)^1 (R_{i1}^1)^{-1} G_i^1 \\
\tilde{M}_{i+2} &= M_{i+2} - (M_{i+1} \Pi_i)^1 (R_{i1}^1)^{-1} J_i^1 \\
\tilde{Z} &= Z - \frac{1}{(M_{i+1} \Pi_i)^1 (R_{i1}^1)^{-1} H_i^1} \left[ (M_{i+1} \Pi_i)^2 - (M_{i+1} \Pi_i)^1 (R_{i1}^1)^{-1} R_{i2}^1 \right] \\
\tilde{d} &= d - (M_{i+1} \Pi_i)^1 (R_{i1}^1)^{-1} h_i^1.
\end{align}
$$
Here, \((M_{i+1} \Pi_i)^1\) refers to the first \(n - n_i\) columns of \(M_{i+1} \Pi_i\), while \((M_{i+1} \Pi_i)^2\) refers to the last \(n_i\) columns.

The second block row of \((4.15)\) becomes part of an auxiliary constraint of the form \((4.1)\). Writing it out explicitly, we obtain

\[
G_i^2 x_i + J_i^2 x_{i+2} + \begin{bmatrix} H_i^2 \\ R_i^{22} \end{bmatrix} \lambda = h_i^2.
\]

Since \(x_i\) and \(x_{i+2}\) are adjacent stages in the reduced system, \((4.20)\) does indeed have the form \((4.1)\). This constraint is merged with the other auxiliary constraints involving \(x_{i+1}\). The constraint

\[
C_i x_i + D_i x_{i+1} + E_i \lambda = g_i
\]

transforms to

\[
\dot{C}_i x_i + \dot{D}_i x_{i+2} + \dot{E}_i \dot{\lambda} = \dot{g}_i,
\]

where

\[
\begin{align*}
\dot{C}_i &= C_i - (D_i \Pi_i)^1 (R_i^{11})^{-1} C_i^1, \\
\dot{D}_i &= - (D_i \Pi_i)^1 (R_i^{11})^{-1} J_i^1, \\
\dot{E}_i &= \begin{bmatrix} E_i - (D_i \Pi_i)^1 (R_i^{11})^{-1} H_i^1 & (D_i \Pi_i)^2 - (D_i \Pi_i)^1 (R_i^{11})^{-1} R_i^{12} \end{bmatrix}, \\
\dot{g}_i &= g_i - (D_i \Pi_i)^1 (R_i^{11})^{-1} h_i^1.
\end{align*}
\]

Similarly, the next auxiliary constraint transforms to

\[
\dot{C}_{i+1} x_{i+1} + \dot{D}_{i+1} x_{i+2} + \dot{E}_{i+1} \dot{\lambda} = \dot{g}_{i+1},
\]

where

\[
\begin{align*}
\dot{C}_{i+1} &= - (C_{i+1} \Pi_i)^1 (R_i^{11})^{-1} C_i^1, \\
\dot{D}_{i+1} &= D_{i+1} - (C_{i+1} \Pi_i)^1 (R_i^{11})^{-1} J_i^1, \\
\dot{E}_{i+1} &= \begin{bmatrix} E_{i+1} - (C_{i+1} \Pi_i)^1 (R_i^{11})^{-1} H_i^1 & (C_{i+1} \Pi_i)^2 - (C_{i+1} \Pi_i)^1 (R_i^{11})^{-1} R_i^{12} \end{bmatrix}, \\
\dot{g}_{i+1} &= g_{i+1} - (C_{i+1} \Pi_i)^1 (R_i^{11})^{-1} h_i^1.
\end{align*}
\]

The auxiliary constraint for the reduced system is obtained by aggregating \((4.20)\), \((4.21)\), and \((4.22)\). The third block row in \((4.15)\) contributes the regular constraint of the form \((1.4)\) to the reduced system, namely

\[
\dot{A}_i x_i + \dot{B}_i x_{i+2} + \begin{bmatrix} \bar{F}_i \\ 0 \end{bmatrix} \lambda = \bar{f}_i.
\]

As in the simple case of Section 4.1, any two non-adjacent stages can be eliminated concurrently, so our strategy is to eliminate the even stages \(x_2, x_4, \text{ and so on}\). (If \(N\) is odd, the final stage—stage \(N + 1\)—is not eliminated, however.) The reduced system obtained from this process has \(\tilde{N} = \lfloor (N + 1)/2 \rfloor\) stages, where \(\lfloor \cdot \rfloor\) denotes the integer part. The unknowns in the reduced system are

\[
\tilde{x}_i \overset{\text{def}}{=} x_{2i-1}, \quad i = 1, \ldots, \tilde{N}, \quad \tilde{x}_{\tilde{N}+1} \overset{\text{def}}{=} x_{N+1}.
\]

As an example, Table 4.1 illustrates the cyclic reduction process for a system which originally has \(N = 10\). Generalizing the special-case algorithm of the previous section, we state the full algorithm as follows.
algorithm compress\( (N, A_i, B_i, F_i, f_i, M_i, Z, d, C_i, D_i, E_i, g_i, x_i, \lambda) \)

if \( N = 1 \)
then solve root problem for \((x_1, x_2, \lambda)\);
   return \(x_1, x_2, \lambda\);
else \( \hat{N} = \lfloor (N + 1)/2 \rfloor \);
   eliminate \(x_2, x_4, \cdots\) to construct the reduced system with data
   \(\hat{A}_i, \hat{B}_i, \hat{M}_i, \) etc., and unknowns \(\tilde{x}_i\) and \(\tilde{\lambda}\);
   compress\( (\hat{N}, \hat{A}_i, \hat{B}_i, \hat{F}_i, \hat{f}_i, \hat{M}_i, \hat{Z}, \hat{d}, \hat{C}_i, \hat{D}_i, \hat{E}_i, \hat{g}_i, \tilde{x}_i, \lambda)\);
   extract components of \(x_i\) and \(\lambda\) from the reduced solution \((\tilde{x}_i, \tilde{\lambda})\);
   recover the remaining components of \(x_i\) by using (4.16);
   return \(x_1, x_2, \cdots, x_{N+1}, \lambda\);
end if.

The root problem for \((x_1, x_2, \lambda)\) has the form (4.12) with the auxiliary constraint

\[
(4.25) \quad C_1 x_1 + D_1 x_2 + E_1 \lambda = g_1.
\]

Note that the number of "extra" parameters introduced at each level of the recursive process is equal to the sum of the dimensions of the matrices \(R^{(2)}\) created at each level. Therefore, the number of additional global parameters in the root problem equals the total amount of rank deficiency encountered during the recursive reduction process.

How effective is the QR column pivoting strategy (1.14) in determining the column rank of the submatrix

\[
(4.26) \quad \begin{bmatrix} B_i \\ A_{i+1} \end{bmatrix} ?
\]

As noted, we can only expect the submatrix \(R_i^{(2)}\) from (4.14) to be exactly zero if the column rank of (4.26) is \(n - n_i\) and exact arithmetic is performed. In actual computations, we would hope that if (4.26) is only a small perturbation away from a rank-(\(n - n_i\)) matrix, then the elements of \(R_i^{(2)}\) would tend to be much smaller than those of \(R_i^{(1)}\). In particular, because of the properties (4.13), we might expect

\[
(4.27) \quad ||R_i^{(2)}|| \ll ||R_i||.
\]

The strategy used in LAPACK (and in our code) for determining the rank of (4.26) is to maintain estimates of the smallest and largest singular values of the columns of (4.26), which are updated after each stage of the QR factorization. Suppose that the ratio of the smaller value to the larger stays above some threshold \(\epsilon\) for stages 1, 2, \cdots, \(k - 1\) of the factorization but falls below \(\epsilon\) after the \(k\)-th column is processed. The matrix is then deemed to have numerical rank \(k - 1\), that is, \(n_i\) is set to \((n - k + 1)\).

This strategy for rank determination is reliable but not foolproof. In applications that require a high degree of reliability, rank-revealing QR factorizations (see, for example, Chan [2]) can be used. For most purposes, including ours, the strategy described in the previous paragraph is sufficient.
We close this section with a discussion of the total amount of rank deficiency (i.e., the number of additional parameters) that arises during the reduction process. If the root problem contains $O(N)$ extra parameters, then there would be little point in performing the reduction, since the root problem would be similar in size to the original problem while possessing less exploitable structure. As we show here, we expect the number of additional parameters not to exceed $\log_2 N(n + p)$, and even this estimate is pessimistic. Hence, our algorithm is worth using when $N$ is sufficiently large with respect to $n$ and $p$. This bound rests on an assumption that the columns pivoting strategy discussed above effectively reveals the numerical rank.

To start with, we seek an upper bound on the amount of rank deficiency detected during the first stage of the factorization process. Assume for simplicity that $N$ is even. Our algorithm performs QR factorization with column pivoting to alternate block columns of the matrix $W$ of Theorem 3.9. To demonstrate this claim, let us rearrange and partition $W$ as follows:

$$W = \begin{bmatrix} B_1 & A_2 & B_3 & A_4 & \cdots & B_{N-1} & A_N \\ \end{bmatrix} = \begin{bmatrix} G \mid H \end{bmatrix},$$

where $G$ is $Nn \times \frac{1}{2}Nn$ and $H$ has dimension $Nn \times (\frac{1}{2}N - 1)n$. The first stage of the algorithm actually performs QR factorization with column pivoting on the entire submatrix $G$. (The $2n \times n$ blocks along the diagonal of $G$ do not interact numerically during this factorization process, except that the column ordering becomes jumbled.) Since, by Theorem 3.9, the numerical rank of $W$ is at least $(N - 1)n - (n + p)$, and since the numerical rank of $H$ cannot exceed its number of columns $(\frac{1}{2}N - 1)n$, we have from Lemma 3.3 that

$$\text{nrank}(G) \geq \text{nrank}(W) - \text{nrank}(H) \geq (N - 1)n - (n + p) - (\frac{1}{2}N - 1)n = \frac{1}{2}Nn - (n + p).$$

We deduce that at most $n + p$ small diagonals appear during the factorization, so no more than $n + p$ extra parameters are added during the first stage.

We look now at the matrix to be factored during the second stage. After the $Q$ factor for $G$ is applied to the whole matrix $W$ and some row rearrangement is performed, we obtain

$$(4.28) \quad PQ^T \begin{bmatrix} R \mid G \mid H \end{bmatrix} = \begin{bmatrix} R_1 & R_3 & J_1 & J_3 & J_5 \\ R_5 & \cdots & \cdots & \cdots & \cdots \\ \end{bmatrix} = \begin{bmatrix} R \mid J \end{bmatrix},$$

where the $R_i$, $J_i$, $H_i$, etc., blocks are generated during the factorization (see (4.4)). Note that the matrix $W_2$ has exactly the same form as the original $W$, but about half as many blocks. Its dimension is $\frac{1}{2}Nn \times (N - 1)n$. Since the upper partition $\begin{bmatrix} R \mid J \end{bmatrix}$ of (4.28) is $\frac{1}{2}Nn \times (N - 1)n$, we can apply Lemma 3.3 again to deduce that

$$\text{nrank}(W_2) = \text{nrank}(\begin{bmatrix} 0 \mid W_2 \end{bmatrix}) \geq \text{nrank}(W) - \text{nrank}(\begin{bmatrix} R \mid J \end{bmatrix}) \geq (N - 1)n - (n + p) - \frac{1}{2}Nn = (\frac{1}{2}N - 1)n - (n + p).$$
The second stage of our algorithm proceeds as with the first, except that we operate on $\tilde{W}_2$ instead of $\tilde{W}$. Theorem 3.9 can be applied again to deduce that at most another $n + p$ small diagonals may be produced at this stage, adding at most another $n + p$ parameters to the reduced problem. This logic applies recursively—at most $n + p$ parameters are added at each stage—so the total upper bound is $\lceil \log_2 N \rceil (n + p)$, as claimed.

5. Computational Results. Using the MPI (message-passing interface) parallel programming tools (see Gropp, Lusk, and Skjellum [5]) we have developed a portable implementation of the algorithm of Section 4.2. We report here on computational experience with the code on the IBM Sp scalable POWER-parallel system at Argonne National Laboratory. This system is a message-passing distributed-memory computer with 128 processors. Each process is equipped with xxx MB of memory and xGB of local disk space, and has a peak processing speed of xxx MFlops. To communicate data between nodes, the machine uses a high-performance switch which has a latency of xx microseconds and a transmission rate of x.x MB/sec.

At present, our code executes on $2^k$ processors, where $k$ is any nonnegative integer. Each processor is assigned an equal number of block rows, and each executes the algorithm compress until its own part of the problem is reduced to a single-stage problem with an auxiliary constraint of the form (4.1). No communication between nodes is needed during this initial reduction phase. In the second phase, the single rows of data on each processor are combined, by using the classic binary tree communication pattern. At the top level of this tree, the odd-numbered nodes 1, 3, 5, $\cdots$ pass their data to their left-neighboring even-numbered nodes 0, 2, 4, $\cdots$. The even nodes then combine the two stages of data into a single stage. The process continues recursively, exactly as described in Section 4.2. At the next level, the nodes 2, 6, 10, $\cdots$ pass their data to nodes 0, 4, 8, $\cdots$, respectively. Finally, the root problem (4.12),(4.25) is solved on node 0. The binary tree combination process is then inverted to recover the various components $z_i$ of the solution. Figure ??? (to be inserted) illustrates the binary tree combination process for a system of four processors.

Our implementation treated the matrices in (1.3),(1.4) as dense. LAPACK library routines were called instead of our technique for handling the general case (Section 4.2) was properly implemented. Correct answers were obtained, and the slight load imbalance mentioned above had only minor effects on execution times.  

Table 5.1, 5.2, and 5.3 give computational results for randomly-generated data. The purpose of these tests was to check the speed of the method for problems of a reasonable size and to show that the deterioration due to interprocessor communication is not too great. For these tests we use the "scaled speedup" approach of increasing the problem size in direct proportion to the number of processors, so that the amount of data per processor is held fixed. In our case, 100 stages of the problem were assigned to each processor. We chose the other parameters in the problem as $n = 20$, $m = 20$, and $p$ (the number of rows of the $M_i$) was set to 45, giving a slightly overdetermined set of side conditions.

used to replace with Table 5.3 and another table for rank-deficient problems on the IBM-SP.

In Tables 5.1 and 5.3, the original problem had the form (1.3),(1.4) with all matrices $A_i$, $B_i$, $F_i$, $M_i$, and $Z$ dense. There was no rank-deficiency, so that the "simple case" of elimination described in Section 4.1 suffices throughout. As can be observed from the tables, the execution times grow only slowly with the number of processors. If "perfect" speedup was attained, these times would remain constant; the time increase arises from the inter-processor communication during the binary-tree phase of the compression process, and from the fact that most processors are idle for some fraction of this phase.

In Table 5.2, some rank-deficiency was introduced into the first partition in the problem, to test that our technique for handling the general case (Section 4.2) was properly implemented. Correct answers were obtained, and the slight load imbalance mentioned above had only minor effects on execution times.

talk about absolute execution speed in terms of megaflops.

A. Useful results from numerical linear algebra. Here we prove some technical results that are used in Section 3. These are largely a consequence of results from in Stewart and Sun [9], an excellent source for material on matrix perturbations. In quoting definitions and theorems from this book, we often restrict ourselves to special cases that are sufficient for our purposes.

A notational note: For any $t \times s$ matrix $X$, we use $R(X)$ to denote the range space of $X$ (the subspace of $\mathbb{R}^t$ spanned by the columns of $X$).

The following theorem and definitions serve to define some terms.
Table 5.1
Timing Results for Touchstone Delta, Random Problems, Full Rank, N = 100 per processor, n = 20, m = 20, p = 45

<table>
<thead>
<tr>
<th>processors</th>
<th>time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.28</td>
</tr>
<tr>
<td>2</td>
<td>7.38</td>
</tr>
<tr>
<td>4</td>
<td>7.48</td>
</tr>
<tr>
<td>8</td>
<td>7.56</td>
</tr>
<tr>
<td>16</td>
<td>7.92</td>
</tr>
<tr>
<td>32</td>
<td>8.02</td>
</tr>
<tr>
<td>64</td>
<td>8.09</td>
</tr>
<tr>
<td>128</td>
<td>8.20</td>
</tr>
</tbody>
</table>

Table 5.2
Timing Results for Touchstone Delta, Random Problems, Rank Deficient in First Partition, N = 100 per processor, n = 20, m = 20, p = 45

<table>
<thead>
<tr>
<th>processors</th>
<th>time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7.30</td>
</tr>
<tr>
<td>2</td>
<td>7.38</td>
</tr>
<tr>
<td>4</td>
<td>7.49</td>
</tr>
<tr>
<td>8</td>
<td>7.59</td>
</tr>
<tr>
<td>16</td>
<td>7.68</td>
</tr>
<tr>
<td>32</td>
<td>7.77</td>
</tr>
</tbody>
</table>

Table 5.3
Timing Results for IBM SP, Random Problems, Full Rank, N = 100 per processor, n = 20, m = 20, p = 45

<table>
<thead>
<tr>
<th>processors</th>
<th>time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.04</td>
</tr>
<tr>
<td>2</td>
<td>2.07</td>
</tr>
<tr>
<td>4</td>
<td>2.14</td>
</tr>
<tr>
<td>8</td>
<td>2.17</td>
</tr>
<tr>
<td>16</td>
<td>2.29</td>
</tr>
<tr>
<td>32</td>
<td>2.35</td>
</tr>
</tbody>
</table>
THEOREM A.1. Let $X$ and $Y$ be $t \times s$ real matrices with $s \leq t$ and full rank $s$. If $2s \leq t$, there is a $t \times t$ orthogonal matrix $Q$ such that

$$\mathcal{R}(QX) \text{ is spanned by the columns of } \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix},$$

$$\mathcal{R}(QY) \text{ is spanned by the columns of } \begin{bmatrix} \Gamma \\ \Sigma \\ 0 \end{bmatrix},$$

where $\Gamma$ and $\Sigma$ are both diagonal matrices with

\begin{align*}
(A.1a) & \quad \Gamma = \text{diag}(\gamma_1, \ldots, \gamma_s), \quad \Sigma = \text{diag}(\mu_1, \ldots, \mu_s), \\
(A.1b) & \quad 0 \leq \gamma_1 \leq \cdots \leq \gamma_s \leq 1, \quad 1 \geq \mu_1 \geq \cdots \geq \mu_s \geq 0, \\
(A.1c) & \quad \gamma_i^2 + \mu_i^2 = 1, \quad i = 1, 2, \ldots, s.
\end{align*}

On the other hand, if $2s > t$, there is a $t \times t$ orthogonal matrix $Q$ such that

$$\mathcal{R}(QX) \text{ is spanned by the columns of } \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix},$$

$$\mathcal{R}(QY) \text{ is spanned by the columns of } \begin{bmatrix} \Gamma \\ 0 \\ \Sigma \end{bmatrix},$$

where $\Gamma$ and $\Sigma$ are diagonal matrices of dimension $t - s$ with

\begin{align*}
(A.2a) & \quad \Gamma = \text{diag}(\gamma_1, \ldots, \gamma_{t-s}), \quad \Sigma = \text{diag}(\mu_1, \ldots, \mu_{t-s}), \\
(A.2b) & \quad 0 \leq \gamma_1 \leq \cdots \leq \gamma_{t-s} \leq 1, \quad 1 \geq \mu_1 \geq \cdots \geq \mu_{t-s} \geq 0, \\
(A.2c) & \quad \gamma_i^2 + \mu_i^2 = 1, \quad i = 1, 2, \ldots, t-s.
\end{align*}

Proof. See Stewart and Sun [9, Theorem 1.5.2]. \( \Box \)

Note that the quantities $\Gamma$ and $\Sigma$ are properties of the subspace pair $(\mathcal{R}(X), \mathcal{R}(Y))$ and are independent of the specific representation in terms of $X$ and $Y$. Hence, in the following definition, it is reasonable to talk in terms of subspaces rather than matrices.

DEFINITION 2. Let $X$ and $Y$ are any two subspaces of $\mathbb{R}^t$ with common dimension $w$. The matrix of canonical angles $\Phi$ between $X$ and $Y$ is

$$\Phi = \sin \Sigma,$$

where $\Sigma$ is the matrix from Theorem A.1; that is, the diagonal matrix of dimension $\min(s, t - s)$ whose diagonal elements are $\sin \mu_i$, $i = 1, 2, \ldots, \min(s, t - s)$.

It is easy to show that the matrix of canonical angles between two subspaces $X$ and $Y$ is the same as between their complements $X^\perp$ and $Y^\perp$.

We are now ready to prove Theorem 3.4

Proof of Theorem 3.4. We prove the theorem only for the case of $2s \leq t$, since the proof for $2s > t$ is similar.

Denote the SVD of $A$ by

$$\begin{bmatrix} U^T \\ V^T \end{bmatrix} A V = \begin{bmatrix} \Lambda \\ 0 \end{bmatrix},$$

where $\Sigma$ is the matrix from Theorem A.1; that is, the diagonal matrix of dimension $\min(s, t - s)$ whose diagonal elements are $\sin \mu_i$, $i = 1, 2, \ldots, \min(s, t - s)$.
where \( U_1 \) is \( t \times s \), \( U_2 \) is \( t \times (t-s) \), \( V \) is \( s \times s \), and \( \Lambda = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_s) \). Denote the SVD of \( \tilde{A} \), partitioned conformally with (A.3), by

\[
\begin{bmatrix}
\tilde{U}^T & \tilde{U}_2^T
\end{bmatrix}
\begin{bmatrix}
\tilde{A}V
\end{bmatrix} =
\begin{bmatrix}
\tilde{\Lambda}
0
\end{bmatrix}.
\]

(Note that \( \mathcal{R}(U_2) \) and \( \mathcal{R}(\tilde{U}_2) \) span the null spaces of \( A^T \) and \( \tilde{A}^T \), respectively.) We now apply Theorem V.4.1 of Stewart and Sun [9] (attributed to Wedin), identifying \( A, \tilde{A}, \) and \( \sigma \), with \( \tilde{A}, A, \) and \( \delta \), respectively, in the statement of that theorem. We obtain

\[
\| \sin \Phi \|_F \leq \frac{1}{\sigma_s} \sqrt{\| R \|_F^2 + \| S \|_F^2},
\]

where

\[
R = \tilde{A}V - U_1\Lambda, \quad S = \tilde{A}^TU_1 - V\Lambda,
\]

and \( \Phi \) is the matrix of canonical angles between the subspaces \( \mathcal{R}(U_2) \) and \( \mathcal{R}(\tilde{U}_2) \). Now

\[
\| R \|_F = \| U^T(\tilde{A}V + U_1\Lambda) \|_F = \left\| U^TAV + U^T(\tilde{A} - A)V + \begin{bmatrix} \Lambda \\ 0 \end{bmatrix} \right\| = \left\| U^T(\tilde{A} - A)V \right\|_F = \| \tilde{A} - A \|_F,
\]

while, similarly,

\[
\| S \|_F = \| \tilde{\Lambda} - \Lambda \|_F.
\]

Hence from (A.4), we have

\[
\| \sin \Phi \|_F \leq \frac{\sqrt{2}}{\sigma_s} \| \tilde{\Lambda} - \Lambda \|_F.
\]

From Theorem A.1 we deduce that there is a \( t \times t \) orthogonal matrix \( Q \) such that

\[
\mathcal{R}(Q\tilde{U}_2) \text{ is spanned by the columns of } \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix},
\]

\[
\mathcal{R}(QU_2) \text{ is spanned by the columns of } \begin{bmatrix} \Gamma \\ \Sigma \\ 0 \end{bmatrix},
\]

where \( \Sigma = \sin \Phi \) by Definition 2, and \( \Sigma \) and \( \Gamma \) are diagonal matrices with dimension \( s \) whose properties are described in Theorem A.1. Recall that \( \tilde{Y} \) is our chosen matrix whose columns span \( \mathcal{R}(U_2) \) (the null space of \( \tilde{A}^T \)). Since \( \mathcal{R}(Q\tilde{U}_2) = \mathcal{R}(Q\tilde{Y}) \), we have from (A.6a) that there is an \( s \times s \) nonsingular matrix \( T \) such that

\[
Q\tilde{Y} = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} T.
\]

In fact, \( T \) is orthogonal since \( Q \) and \( \tilde{Y} \) are orthonormal. We now define our matrix \( Y \) to be

\[
Y = Q^T \begin{bmatrix} \Gamma \\ \Sigma \\ 0 \end{bmatrix} T.
\]
PARALLEL PARAMETER IDENTIFICATION FOR ODEs

It is easy to verify, using (A.1), that \( Y \) is orthonormal. To check that its columns span \( \mathcal{R}(U_2) \), note that

\[
\mathcal{R}(U_2) = \mathcal{R} \left( Q^T \begin{bmatrix} \Gamma & \Sigma \\ \Sigma & 0 \end{bmatrix} \right) = \mathcal{R}(YT^T) = \mathcal{R}(Y).
\]

Hence, from (A.7) and (A.8), we obtain

\[
\|Y - \hat{Y}\|_F = \left\| Q^T \begin{bmatrix} \Gamma - I \\ \Sigma \\ 0 \end{bmatrix} \right\|_F = \left\| \begin{bmatrix} \Gamma - I \\ \Sigma \\ 0 \end{bmatrix} \right\|_F = \left[ \sum_{i=1}^s (1 - \gamma_i)^2 + \sigma_i^2 \right]^{1/2}.
\]

Now since \( \gamma_i \) and \( \sigma_i \) have the properties described in (A.1), we have

\[
(1 - \gamma_i)^2 = 1 - 2\gamma_i + \gamma_i^2 \leq 1 - 2\gamma_i^2 + \gamma_i^2 = \sigma_i^2.
\]

Hence, from (A.5), we obtain

\[
\|Y - \hat{Y}\|_F = \sqrt{2} \left[ \sum_{i=1}^s \sigma_i^2 \right]^{1/2} = \sqrt{2} \| \Phi \|_F \leq \frac{2}{\sigma_*} \| A - \hat{A} \|_F,
\]

proving the result. \( \square \)

Content-Length: 2335 X-Lines: 63 Status: RO

REFERENCES

PREVIOUS PUBLICATIONS IN THIS SERIES:

<table>
<thead>
<tr>
<th>Number</th>
<th>Author(s)</th>
<th>Title</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>96-20</td>
<td>A.A. Reusken</td>
<td>An approximate cyclic reduction multilevel preconditioner for general sparse matrices</td>
<td>October '96</td>
</tr>
<tr>
<td>96-21</td>
<td>A.F.M. ter Elst</td>
<td>Heat kernels and Riesz transforms on nilpotent Lie groups</td>
<td>November '96</td>
</tr>
<tr>
<td></td>
<td>D.W. Robinson</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A. Sikora</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96-22</td>
<td>A.J.H. Frijns</td>
<td>A validation of the quadriphasic mixture theory for intervertebral disc tissue</td>
<td>November '96</td>
</tr>
<tr>
<td></td>
<td>J.M. Huyghe</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>J.D. Janssen</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96-23</td>
<td>L.P.H. de Goey</td>
<td>The Mass Burning Rate of Stretched Flames with Multi-Component Transport and Chemistry</td>
<td>December '96</td>
</tr>
<tr>
<td></td>
<td>J.H.M. ten Thije</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Boonkamp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96-24</td>
<td>A.F.M. ter Elst</td>
<td>Second-order subelliptic operators on Lie groups III: Hölder continuous coefficients</td>
<td>December '96</td>
</tr>
<tr>
<td></td>
<td>D.W. Robinson</td>
<td></td>
<td></td>
</tr>
<tr>
<td>96-25</td>
<td>J. de Graaf</td>
<td>Evolution equations</td>
<td>December '96</td>
</tr>
<tr>
<td>96-26</td>
<td>J. de Graaf</td>
<td>Spaces of harmonic functions and evolution equations in them</td>
<td>December '96</td>
</tr>
<tr>
<td>96-27</td>
<td>R.M.M. Mattheij</td>
<td>Parallel algorithms for parameter identification in odes</td>
<td>December '96</td>
</tr>
<tr>
<td></td>
<td>S.J. Wright</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>