ON SPLINE BASIS SELECTION FOR SOLVING DIFFERENTIAL EQUATIONS

by

U. Ascher*, S. Pruess** and R.D. Russell***

Technical Report 81-4

April 1981

ABSTRACT

The suitability of B-splines as a basis for piecewise polynomial solution representation for solving differential equations is challenged. Two alternative local solution representations are considered in the context of collocating ordinary differential equations: "Hermite-type" and "monomial". Both are much easier and shorter to implement and somewhat more efficient than B-splines.

A new condition number estimate for the B-splines and Hermite-type representations is presented. One choice of the Hermite-type representation is experimentally determined to produce roundoff errors at most as large as

---

* Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada. Supported in part under NSERC (Canada) Grant A4306.

** Department of Mathematics, University of New Mexico, Albuquerque, New Mexico. A portion of this work was completed while on sabbatical at the University of British Columbia.

*** Department of Mathematics, Simon Fraser University, Burnaby, B.C., Canada. Supported in part under NSERC Grant A8781.

those for B-splines. The monomial representation is shown to have a much smaller condition number than the other ones, and correspondingly produces smaller roundoff errors, especially for extremely nonuniform meshes. The operation counts for the two local representations considered are about the same, the Hermite-type representation being slightly cheaper. It is concluded that both representations are preferable, and the monomial representation is particularly recommended.

1. INTRODUCTION

Finite element methods for solving differential equations normally produce piecewise polynomial (i.e. spline) solution approximations. For piecewise polynomials in one space variable, or their tensor products, it has become increasingly popular to use a B-spline basis representation. For instance, Ascher et al. [2] provide a general purpose code for boundary value ordinary differential equations (ODE's), and Leaf et al. [14] and Schryer [20] provide Galerkin codes for time dependent problems. This choice of B-splines is motivated primarily by the fundamental work of de Boor [5], which contains many elegant results that allow one to handle the basis with reasonable stability and efficiency.

The utility of B-splines in approximation problems such as surface fitting and curve design (cf. de Boor [5], Grosse [12]) is not in question here. Recently, however, some doubt has been expressed as to their suitability for solving differential equations, especially when low continuity piecewise polynomials are used.

In this paper we examine B-splines and two classes of alternatives:

(a) Certain extensions of the Hermite basis, briefly proposed but not pursued in de Boor - Swartz [7].

(b) "Monomial bases", recently discussed by Osborne [17].

Our purpose is to discuss and compare these alternative bases in terms of efficiency, conditioning, and actual performance on a set of test problems. While we only consider the collocation method with Gaussian points for a scalar ODE, the conclusions should apply with minor modifications to other finite element methods, especially using low continuity splines, and to mixed order systems of ODE's.

In general, the spline solution is determined by two types of constraints: that it satisfy required continuity conditions and that it satisfy some discretization equations (e.g. collocation equations) relating it to the true solution of the differential problem. One feature of B-splines is that the continuity conditions are built-in. As a result, only the discretization equations need to be explicitly satisfied. Another feature of B-splines is that they have small local support, i.e., each basis function is nonzero over only a few mesh subintervals, so that the resulting matrix for the discretization equations is nicely banded. Also, for low continuity piecewise polynomials, some of the B-splines and their derivatives are largely mesh independent and hence cheap to evaluate (Ascher - Russell [3]). Unfortunately, the discretization matrices for a higher order differential equation have condition numbers which grow very rapidly as the mesh is refined.

The Hermite-type and monomial bases are introduced from a truly local viewpoint, in that the basis functions are defined using local representations (involving only one subinterval). The resulting basis itself need not even be known. For the Hermite-type bases, the local representations for neighboring subintervals are matched to allow continuity conditions to be implicitly satisfied, while with the monomial bases the local polynomial representations are matched. The local nature of both basis representations permits efficient formation of discretization and continuity equations and solution of the resulting system using local elimination (condensation of parameters, which is possible also for B-splines). Both turn out to be somewhat more efficient than B-splines in terms of arithmetic operation counts (see also [7]).

A new condition number estimate for the B-spline and Hermite-type representations is presented. This estimate may, unfortunately, grow very large (with a corresponding loss of accuracy), if the mesh is highly nonuniform

or unreasonably dense. For monomials, local condensation of the discretization equations yields multiple-shooting-like matrices which are much better conditioned than those in the other cases. Experimentally, we find that one particular Hermite-type basis is consistently at least as accurate as B-splines or other Hermite-type bases, and that the monomial representations are in turn at least as accurate as the Hermite-type basis. Another important advantage that both classes of bases have over B-splines is that they are much easier to implement. We conclude that in many contexts, the alternative bases discussed here, especially the monomial representations, are preferable to B-splines, for the solution of differential equations.

Notation and general background material are found in the next section. Section 3 is concerned with B-splines, while sections 4 and 5 introduce the alternative two classes of bases. Numerical examples are given in the final section, which also summarizes our observations.

2. PRELIMINARIES

Our model problem is the m^{th} order ODE

$$(2.1)\quad Lu(x) := D^m u(x) - \sum_{\ell=1}^{m} c_\ell(x) D^{\ell-1} u(x) = f(x), \qquad a \le x \le b$$

with boundary conditions (BC)

$$(2.2)\quad B_a \underset{\sim}{z}(u(a)) = \underset{\sim}{b}_a, \qquad B_b \underset{\sim}{z}(u(b)) = \underset{\sim}{b}_b.$$

Here B_a is $m_a \times m$, B_b is $m_b \times m$, $m_b = m - m_a$, and

$$(2.3)\quad \underset{\sim}{z}(u(x)) := (u(x), Du(x), \ldots, D^{m-1}u(x))^T.$$

It is assumed that a sufficiently smooth, unique solution $u(x)$ to (2.1), (2.2) exists and satisfies

$$(2.4)\quad ||D^j u|| := \max_{a \le x \le b} |D^j u(x)| \le c \cdot \max\{||f||, ||\underset{\sim}{b}_a||_\infty, ||\underset{\sim}{b}_b||_\infty\} \qquad 0 \le j \le m-1$$

for some constant c. Note that when solving nonlinear problems by quasi-linearization, each iteration consists of solving a problem like (2.1), (2.2), where the coefficients $\{c_\ell(x), f(x)\}$ involve approximations to $z(u(x))$ from the previous iteration.

The numerical method for solving (2.1), (2.2) determines a piecewise polynomial approximation $u_\Delta(x)$ to the exact solution $u(x)$ on a given mesh

$$(2.5a)\quad \Delta : a = x_1 < x_2 < \ldots < x_N < x_{N+1} = b.$$

Let

$$(2.5b)\quad h_i := x_{i+1} - x_i, \quad 1 \le i \le N, \quad h_0 := h_1, \quad h_{N+1} := h_N,$$

$$h := \max_{1\le i\le N} h_i, \quad \underline{h} := \min_{1\le i\le N} h_i.$$

Given k points, $k \ge m$,

$$(2.6)\quad 0 \le \rho_1 < \ldots < \rho_k \le 1,$$

the collocation solution $u_\Delta(x)$ is defined by requiring the following four conditions:

(C1) On each subinterval (x_i, x_{i+1}), $1 \le i \le N$, $u_\Delta(x)$ is a polynomial of order $k + m$ (degree $< k + m$).

(C2) $u_\Delta(x) \in C^{(m-1)}[a,b]$

(C3) $u_\Delta(x)$ satisfies (2.1) at the kN collocation points

$$\xi_{ij} := x_i + h_i\rho_j \qquad 1 \le j \le k, \quad 1 \le i \le N$$

(C4) $u_\Delta(x)$ satisfies the BC (2.2).

If the points (2.6) satisfy the orthogonality relation

$$(2.7)\quad \int_0^1 p(t) \prod_{j=1}^{k} (t-\rho_j)dt = 0$$

for all polynomials p(t) of order n ($m \le n \le k$), then

$$(2.8)\quad ||D^j(u - u_\Delta)|| = O(h^{k+m-j}) \qquad 0 \le j \le m$$

and superconvergence occurs at the mesh points,

$$(2.9)\quad |D^j(u - u_\Delta)\,(x_i)| = O(h^{k+n}) \qquad 1 \le i \le N+1, \quad 0 \le j \le m-1.$$

(cf. de Boor-Swartz [6]). The collocation points used in section 6 are the Gaussian points, giving n=k, but the conclusions apply for other important choices of collocation points as well (e.g., cf. Ascher-Weiss [4]).

Note that the problem (2.1), (2.2) can always be converted to an equivalent first order system for $\underset{\sim}{z}(x)$. If collocation at the same points as above is applied to the converted system, with each component of $\underset{\sim}{z}_\Delta(x)$ being a continuous piecewise polynomial of order k+1, then the superconvergence results (2.9) remain unchanged. However, the number of free parameters in $\underset{\sim}{z}_\Delta(x)$ before satisfying the continuity and collocation conditions is Nkm, while that for $u_\Delta(x)$ is only N(k+m). Thus, an effective method for handling the higher order differential equation directly is generally expected to be more efficient (cf. [18]).

Any representation for $u_\Delta(x)$ which satisfies (C1) contains (k+m)N free parameters which are determined by imposing conditions (C2) - (C4), either explicitly or implicitly. After possibly eliminating some parameters locally, the remaining parameters $\underset{\sim}{\alpha}$ are determined by explicitly satisfying a system of equations

$$(2.10)\quad A\underset{\sim}{\alpha} = \underset{\sim}{f}.$$

For all of the solution representations considered in this paper, the matrix A has an almost block diagonal form, i.e.

$$A = \begin{bmatrix} \hat{B}_a & & & \\ V_1 & & & 0 \\ & V_2 & \ddots & \\ & & & V_N \\ & 0 & & \hat{B}_b \end{bmatrix} \tag{2.11}$$

with $\hat{B}_a$ and $\hat{B}_b$ matrices corresponding to B_a and B_b of (2.2). The N blocks V_i all have the same size and offset (i.e., position with respect to neighboring blocks), which are independent of N. The block V_i arises from the continuity conditions (C2) and/or the collocation equations (C3), relating to the i^{th} subinterval.

Some of the bases in this paper are nodal, i.e., some or all of the components of $\underset{\sim}{\alpha}$ are proportional to the superconvergent values

$$z_{ij} := D^{j-1}u_{\Delta}(x_i) \qquad 1 \le i \le N+1, \quad 1 \le j \le m. \tag{2.12}$$

The local nature of the representation may allow the other components of $\underset{\sim}{\alpha}$ to be eliminated locally (within the block V_i), which effectively reduces the size of A. This process, known to workers in finite elements as condensation of parameters, is also possible for B-splines, as discussed in section 4.

We shall compare the relative efficiency of various spline basis representations with respect to operation counts and storage requirements. The work estimates count only multiplications and divisions. The number of

subintervals N is assumed to be large, so that any storage or work estimates independent of N, such as those needed for initialization, are ignored. Note that, because of the form (2.11), the total work as well as the total storage is always linear in N.

Two components of the total work are considered:

W_1 := Work required for matrix assembly.

W_2 := Work required for linear system solution using Gauss elimination with scaled partial pivoting.

Not considered are the work required for evaluating the coefficient functions in (2.1), which may be a major component of the total work but is independent of the solution representation used, and the work required for the evaluation of the computed solution after obtaining $\underset{\sim}{\alpha}$, which is roughly the same for all the bases considered and hard to compare anyway.

Regarding the latter work component, it may be particularly important to consider the evaluation of $\underset{\sim}{z}(u_\Delta(\xi_{ij}))$, $1 \le j \le k$, $1 \le i \le N$, which is done after each quasilinearization iteration for a nonlinear problem. A competitive way to do this is by local interpolation of the neighboring superconvergent values

$$\underset{\sim}{z}_i = (z_{i1},\ldots,z_{im})^T \qquad 1 \le i \le N+1. \tag{2.13}$$

For instance, using the Hermite interpolate of $\underset{\sim}{z}_i$ and $\underset{\sim}{z}_{i+1}$ instead of $u_\Delta(x)$ directly in evaluating $\underset{\sim}{z}(u_\Delta(\xi_{ij}))$ is usually sufficiently accurate for $m > 1$. Obtaining these superconvergent values, which are not readily available for B-splines, costs considerably less than obtaining $\underset{\sim}{z}(u_\Delta(\xi_{ij}))$. Thus, the various solution representations perform roughly the same in this

respect.

It is important to realize, then, that the comparison between the various bases is only made for W_1 and W_2, and that while this is the major source of work when the coefficients of the ODE are not too complicated, it may not be representative of the total work estimate.

To compare storage required we consider

S := matrix storage requirements.

The storage needed for A depends upon the elimination strategy used. When performing Gauss elimination with row interchanges only, taking the block structure of A into account as in the package SOLVEBLOK [8], then some additional storage due to partial fill-in is necessary. (This is still better than just treating A as a banded matrix). This amount of fill-in does depend to some extent on the solution representation used; however, we do not consider it here because it can be avoided altogether for linear problems [8] or more generally by using a row/column interchange strategy, as in the package MARCEPAK [11].

3. THE B-SPLINE BASIS

A thorough exposition on B-splines, including a discussion of how to solve (2.1) by collocation, can be found in de Boor [5]. The details of an efficient implementation of B-splines in a collocation code are given in [1], [3], from which the operation counts given below are taken.

In contrast to the local representations considered in later sections, the B-splines on [a,b] are piecewise polynomial functions with minimal support subject to the continuity conditions (C2) being implicitly satisfied. The resulting coefficient matrix A has the form (2.11) with each V_i being of dimension k x (k+m) and being offset by k columns from the previous block. The entries of V_i are $L\Psi_r(\xi_{ij})$, $(i-1)k+1 \le r \le ik+m$, $1 \le j \le k$, where Ψ_r are the B-spline basis functions.

For storage we have

$$S = k(k+m)N. \tag{3.1}$$

The work estimates to assemble A are fairly complicated, the dominant part being

$$W_1 = [(m^2 + 5m)k^2 + (\tfrac{1}{3} m^3 + 3m^2)k]N. \tag{3.2}$$

This is actually a slight underestimate of the actual work, since only cubic or higher order terms in k and m are counted. Also

$$W_2 = [\tfrac{1}{3} k^3+(\tfrac{1}{2} m_a+\tfrac{1}{2} m+1)k^2+(m_a m+\tfrac{3}{2} m_a+\tfrac{1}{2} m+\tfrac{2}{3})k+m]N. \tag{3.3}$$

It turns out that the storage requirements for B-splines are precisely the same as for Hermite-type bases. Condensation of parameters is also possible, reducing S of (3.1). This is described in the next section.

The condition number of A for B-splines is similar to that arising with Hermite-type bases, so its discussion is delayed until the next section as well. Here, however, we note that a priori there is reason for alarm, since $L\Psi_r(\ _{ij})$ may grow like h_i^{-m}.

4. HERMITE-TYPE BASES

The bases in this section can be described via a local representation, allowing use of the same information, properly scaled, in each subinterval of a generally nonuniform mesh. We begin by defining a canonical set of k+m polynomials $\phi_j(t)$ over the interval [0,1]. Each $\phi_j(t)$ is a polynomial of order k+m characterized by

$$\lambda_\ell \phi_j = \delta_{j\ell} \qquad 1 \le \ell,\ j \le k+m \tag{4.1}$$

where the linear functionals λ_ℓ, $1 \le \ell \le k+m$, are defined with respect to the points

$$0 \le \eta_1 \le \eta_2 \le \cdots \le \eta_{k+m} \le 1 \tag{4.2}$$

as follows: If $\eta_{\ell-\mu-1} < \eta_{\ell-\mu} = \eta_{\ell-\mu+1} = \cdots = \eta_\ell$ (with $\eta_0 < 0$)

then

$$\lambda_\ell \phi_j := D^\mu \phi_j(\eta_\ell) \qquad 1 \le j \le k+m. \tag{4.3}$$

A Lagrange basis would correspond to the use of distinct points in (4.2); however, to obtain a nodal method, which is what we consider in this section, require

$$\eta_1 = \eta_2 = \cdots \eta_m = 0. \tag{4.4}$$

Our next step is to map the k+m polynomials into each subinterval of the mesh Δ. Thus, if u_Δ is expressed on $[x_i, x_{i+1}]$ by

$$(4.5) \qquad u_\Delta(x) = \sum_{j=1}^{k+m} \alpha_{(i-1)(k+m)+j}\, \phi_j\left(\frac{x-x_i}{h_i}\right) \qquad x_i \le x \le x_{i+1},$$

then

$$(4.6) \qquad \alpha_r = h_i^{j-1} z_{ij}, \qquad 1 \le j \le m,$$

where $r := (i-1)(k+m) + j$. Now, in addition to (4.4) set

$$(4.7) \qquad \eta_{k+1} = \eta_{k+2} = \ldots = \eta_{k+m} = 1,$$

so the continuity conditions (C2) at x_i give

$$(4.8) \qquad \alpha_r/h_i^{j-1} = \alpha_{r-m}/h_{i-1}^{j-1} = z_{ij} \qquad 1 \le j \le m,\ 1 < i \le N.$$

Continuity is implicitly satisfied if instead of (4.5) we write

$$(4.9) \qquad u_\Delta(x) = \sum_{j=1}^{k+m} \alpha_{(i-1)k+j}\, s_{ij} \phi_j\left(\frac{x-x_i}{h_i}\right) \qquad x_i \le x \le x_{i+1}$$

where

$$(4.10) \qquad s_{ij}\alpha_{(i-1)k+j} = h_i^{j-1} z_{ij} \qquad 1 \le j \le m,\ 1 \le i \le N.$$

and s_{ij} are scaling factors, discussed below.

For the case k=m, this corresponds to the standard Hermite basis local representation.

The coefficient matrix A which results from the representation (4.9) can be unbalanced as $h \to 0$, and the choice of the s_{ij} is equivalent to the column scaling of A.

From (4.1) and (4.3),

$$(4.11)\quad D^{\mu}u_{\Delta}(x_i+\eta_{\ell}h_i) = \alpha_{(i-1)k+j}\, s_{ij}/h_i^{\mu} \qquad 1 \le j \le k+m, \quad 1 \le i \le N,$$

so the continuity conditions force the scale factor s_{ij} to satisfy

$$(4.12)\quad s_{ij}/h_i^{j-1} = s_{i-1,k+j}/h_{i-1}^{j-1} \qquad 1 \le j \le m, \quad 1 < i \le N.$$

A simple choice of column scaling compatible with (4.3) is $s_{ij} := h_i^{\mu}$, $1 \le j \le k+m$, so $s_{ij} = s_{i,j+k} = h_i^{j-1}$, $1 \le j \le m$. Then some of the parameters in $\underset{\sim}{\alpha}$ are precisely the superconvergent values z_{ij}. However, for highly nonuniform meshes it is preferable to select a more balanced scaling. We have found the following formulas, which produce $s_{ij} \le 1$, and $s_{ij} = 1$ when possible, to work well in practice:

$$(4.13)\quad s_{ij} := \begin{cases} (h_i/h_{i-1})^{j-1} & 1 \le j \le m, \quad h_i < h_{i-1} \\ (h_i/h_{i+1})^{j-k-1} & k < j \le k+m, \quad h_i < h_{i+1} \\ 1 & \text{otherwise.} \end{cases}$$

The assembly of the matrix A and the vector $\underset{\sim}{f}$ of (2.10) is quite efficient. With continuity built-in by the choice of s_{ij}, the block V_i in (2.11) is generated from the k collocation conditions (C3) in $[x_i, x_{i+1}]$. By (4.9) we

have for each collocation point

$$(4.14)\qquad D^{\ell}u_{\Delta}(\xi_{ir}) = \sum_{j=1}^{k+m} \alpha_{(i-1)k+j}\, s_{ij}\, D^{\ell}\phi_j(\rho_r)/h_i^{\ell} \qquad 1 \le r \le k,\ 1 \le i \le N,\ 0 \le \ell \le m.$$

The constants $D^{\ell}\phi_j(\rho_r)$ are mesh independent and can be evaluated and stored once and for all. This can be done by constructing for each $\phi_j(t)$ the divided difference table with respect to the points $\{\eta_\ell\}$ of (4.2) and then evaluating ϕ_j and its derivatives at each point ρ_r using Newton's interpolation form and nested multiplication; see, e.g., Conte-de Boor [10] for details.

The entries of $V_i = (V^i_{rj})$ are the coefficients of $\alpha_{(i-1)k+j}$ in $Lu_{\Delta}(\xi_{ir})$, so from (2.1) and (4.14)

$$(4.15)\qquad V^i_{rj} = s_{ij}\{h_i^{-m}D^m\phi_j(\rho_r) - \sum_{\ell=1}^{m} c_\ell(\xi_{ir})h_i^{1-\ell}D^{\ell-1}\phi_j(\rho_r)\} \qquad 1 \le j \le k+m,\ 1 \le r \le k,\ 1 \le i \le N.$$

To minimize computation and storage, we generate and store locally the values of $h_i^{-\ell}$, $h_{i-1}^{-\ell}$, $h_{i+1}^{-\ell}$ and $\tau_{\ell r} := c_\ell(\xi_{ir})h_i^{1-\ell}$, $1 \le \ell \le m$, $1 \le r \le k$. Assuming that $h_1^{-\ell}$ values have been generated, the following algorithm is then used:

(4.16)
```
FOR i:=1,...,N DO:
    generate {h_{i+1}^{-ℓ}, ℓ=1,...,m}
    FOR r:=1,...,k DO:
        FOR ℓ:=1,...,m DO:
            τ_{ℓr}:=c_ℓ(ξ_{ir})h_i^{1-ℓ}
        FOR j:=1,...,k+m DO:
            V_{rj}^i:=D^m φ_j(ρ_r)h_i^{-m}
            FOR ℓ:=1,...,m DO:
                V_{rj}^i:=V_{rj}^i - τ_{ℓr}*D^{ℓ-1}φ_j(ρ_r)
        IF i<N and h_i<h_{i+1} THEN
            FOR j:=k+2,...,k+m DO:
                V_{rj}^i:=V_{rj}^i * (h_i/h_{i+1})^{j-k-1}
        IF i>1 and h_i<h_{i-1} THEN
            FOR j:=2,...,m DO:
                V_{rj}^i:=V_{rj}^i*(h_i/h_{i-1})^{j-1}
    adjust {h_{i-1}^{-ℓ}, h_i^{-ℓ}} for the next block
```

The operation count is

$$W_1 = [(m+1)k^2 + (m^2+4m-2)k + m]N, \tag{4.17}$$

which is significantly smaller than that for B-splines (see (3.2)). Since A has the same structure as for B-splines, W_2 and S are the same as in (3.3) and (3.1), respectively.

As noted before, some of the unknown parameters $\underset{\sim}{\alpha}$ relate directly to the nodal values $\{\underset{\sim}{z}_i, i=1,\ldots,N+1\}$. When k > m, condensation of parameters

can be used to locally eliminate the other unknowns, thus reducing W_2 and S. Specifically, since unknowns corresponding to columns m+1 to k of V_i appear only in this block, Gaussian elimination can be performed on these columns to reduce V_i to the form

$$(4.18) \qquad \begin{bmatrix} \tilde{E}_i & T_i & \tilde{F}_i \\ E_i & 0 & F_i \end{bmatrix} \begin{matrix} \} \ k-m \\ \} \ m \end{matrix}$$
$$\underbrace{\quad}_{m} \quad \underbrace{\quad}_{k-m} \quad \underbrace{\quad}_{m}$$

where T_i is upper triangular. If the top k-m rows and middle k-m columns of each block V_i are discarded, then a new m x 2m block $\hat{V}_i = [E_i \ F_i]$ is obtained, offset m columns from the previous one. The remaining unknowns are the scaled nodal values. The total operation count for this condensation plus linear system solution is

$$(4.19) \qquad W_2 = [\tfrac{1}{3}k^3 + \tfrac{1}{2}(m+1)k^2 - (\tfrac{m}{2}+\tfrac{5}{6})k + \tfrac{3}{2}(m_a+1)m^2 + (\tfrac{3}{2}m_a + \tfrac{5}{2})m]N$$

and the total storage for the reduced system is

$$(4.20a) \qquad S = 2m^2N.$$

If one wishes to recover the full solution $u_\Delta(x)$ (and not just the nodal values), then for each i the matrices $\tilde{E}_i$, T_i, $\tilde{F}_i$ need to be saved. Since T_i is upper triangular, back substitution can be applied after $\underset{\sim}{z}_i$ and $\underset{\sim}{z}_{i+1}$ are known, to retrieve the additional local parameters of $\underset{\sim}{\alpha}$. The resulting storage requirements are

(4.20b) $\quad S = (m^2+k^2)N,$

while W_2 is again given by (3.3) because the whole process now amounts to a particular arrangement of the Gauss elimination process for the uncondensed matrix.

The above representation of Hermite-type bases is in terms of local coordinates only. This representation, of course, defines a global basis $\{\Psi_\ell(x)\}_{\ell=1}^{Nk+m}$ on [a,b], which we need not be concerned with for the actual construction of A. For $1 \le i \le N$, we have

$$(4.21)\qquad \Psi_{(i-1)k+j}(x) = \begin{cases} s_{i-1,k+j}\ \phi_{k+j}\left(\dfrac{x-x_{i-1}}{h_{i-1}}\right) & x_{i-1} \le x < x_i \\ s_{ij}\ \phi_j\left(\dfrac{x-x_i}{h_i}\right) & x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \qquad 1 \le j \le m$$

$$(4.22)\qquad \Psi_{(i-1)k+j}(x) = \begin{cases} s_{ij}\ \phi_j\left(\dfrac{x-x_i}{h_i}\right) & x_i \le x < x_{i+1} \\ 0 & \text{otherwise.} \end{cases} \qquad m < j \le k$$

Representation (4.9) is then equivalent to

$$(4.23)\qquad u_\Delta(x) = \sum_{\ell=1}^{Nk+m} \alpha_\ell \Psi_\ell(x).$$

The condensation of parameters (4.18) can be performed for the B-spline matrix as well. Here the middle k-m columns correspond to those B-splines whose support is over one subinterval only and thus must vanish, together with their first m-1 derivatives, at all mesh points. Thus, the nodal values can be recovered from the remaining B-spline coefficients, which are calculated from

the condensed system. The work estimate (4.19) and storage estimates (4.20) then hold for B-splines as well.

Given m and k, from (4.4) and (4.7) the particular Hermite-type basis is determined by the choice of the interpolation points $\{\eta_j\}_{j=m+1}^{k}$. We now turn to the question of choosing these points for k > m. Three particular choices have been considered:

(4.24a) $\eta_j = 0$

(4.24b) $\eta_j = 1/2 \qquad m+1 \le j \le k$

(4.24c) $\eta_j = \frac{j-m}{k-m+1}$

Defining the operator M for [a,b] := [-1,1] by

(4.25) $$(M\underset{\sim}{\gamma})(t) := \sum_{\ell=1}^{Nk+m} \gamma_\ell \Psi_\ell(t),$$

for any vector $\underset{\sim}{\gamma} = (\gamma_1,\ldots,\gamma_{Nk+m})^T$, the condition number of the basis is

(4.26) $$\kappa(M) := ||M||_\infty \, ||M^{-1}||_\infty.$$

Using appropriate derivatives of Chebyshev polynomials, J. Christiansen has obtained bounds on $\kappa(M)$ similar to those for B-splines [9]. These bounds grow with the largest derivative μ appearing in (4.3). For k > 2m, this suggests that the choice (4.24c) is better conditioned than (4.24b), which in turn is better conditioned than (4.24a). We have not checked this numerically for interpolation processes, but our numerical experience indicates that when collocating differential equations with these bases, the opposite is in

fact true. In all our numerical experiments the choice (4.24a) has performed at least as well, and many times marginally better, than choices (4.24b) or (4.24c) or B-splines.

This section is concluded with a discussion of the asymptotic behaviour of the condition number of the collocation matrix, $\kappa(A)$, as the maximum mesh width h tends to 0. Only the dependence of $\kappa(A)$ on the mesh Δ of (2.5) is of interest, so other factors are lumped into a generic constant K.

To make $\kappa(A)$ correspond to realistic implementations we assume that A is row equilibrated, i.e. each row has been divided by its largest entry in magnitude. Thus, the system (2.10) becomes

$$PA\underset{\sim}{\alpha} = P\underset{\sim}{f} \tag{4.27}$$

where P is the diagonal matrix of scale factors. Assume that h is small enough so that A is nonsingular, i.e. there exists a unique collocation solution for a given $\underset{\sim}{f}$. Let G(x,t) be the Green's function for the differential problem (2.1), (2.2). We argue below that

$$\kappa(PA) \approx K \max_{1\le j\le N+1} \sum_{i=1}^{N} h_i^{-m} \int_{x_i}^{x_{i+1}} |G(x_j,t)|\,dt. \tag{4.28}$$

While our discussion falls short of a rigorous proof, we feel that it, together with the supporting numerical results of section 6, strongly indicate that (4.28) does indeed hold.

From the block structure of A and the row scaling we get

$$1 \le ||PA||_\infty \le k+m \tag{4.29}$$

for the uncondensed system. Hence it remains to consider $||(PA)^{-1}||_\infty$. For an arbitrary, sufficiently smooth f(x), let u(x) satisfy (2.1) and (2.2), where we assume without loss of generality that $\underset{\sim}{b}_a = \underset{\sim}{b}_b = \underset{\sim}{0}$. From (4.15), as $h \to 0$ the largest entry in magnitude for row $\ell = (i-1)k+r+m_a$ of A ($1 \leq r \leq k$) has the form $h_i^{-m}(\sigma_{\ell\ell} + O(h_i))$ where $|\sigma_{\ell\ell}|$ can be bounded above and below away from 0, independently of the mesh. Thus, as $h \to 0$,

$$(4.30) \qquad ||PA\underset{\sim}{\alpha}||_\infty = ||P\underset{\sim}{f}||_\infty \sim \max_{1\leq i\leq N} h_i^m \max_{x_i\leq x\leq x_{i+1}} |f(x)|.$$

By (4.11) the coefficients $\underset{\sim}{\alpha}$ satisfy

$$(4.31) \qquad \alpha_{(i-1)k+j} = D^\mu u_\Delta(x_i+\eta_\ell h_i)h_i^\mu/s_{ij}$$

and by (4.13) $h_i^\mu/s_{ij} = O(h^\mu)$. Since $u_\Delta(x)$ converges to u(x) we get

$$||\underset{\sim}{\alpha}||_\infty \sim ||u_\Delta|| \sim \max_{x_j\in\Delta} |u_\Delta(x_j)| \sim \max_{x_j\in\Delta} |u(x_j)|.$$

Using the Green's function representation for u(x),

$$u(x) = \int_a^b G(x,t)f(t)dt,$$

we have

$$(4.32) \qquad \max_{x_j\in\Delta} |u(x_j)| = \max_{1\leq j\leq N+1} \left| \sum_{i=1}^{N} \int_{x_i}^{x_{i+1}} G(x_j,t)[h_i^m f(t)]dt/h_i^m \right| \leq$$

$$\leq \{ \max_{1\leq j\leq N+1} \sum_{i=1}^{N} \int_{x_i}^{x_{i+1}} |G(x_j,t)|dt/h_i^m \} \cdot \{ \max_{1\leq i\leq N} \max_{x_i\leq x\leq x_{i+1}} h_i^m |f(x)| \}.$$

Combining (4.30) and (4.32), as $h \to 0$,

$$(4.32) \qquad ||PA\underset{\sim}{\alpha}||_\infty / ||\underset{\sim}{\alpha}||_\infty \gtrsim \{ \max_{1 \le j \le N+1} \sum_{i=1}^{N} h_i^{-m} \int_{x_i}^{x_{i+1}} |G(x_j,t)|dt \}^{-1}.$$

So

$$(4.34) \qquad ||(PA)^{-1}||_\infty \lesssim \max_{1 \le j \le N+1} \sum_{i=1}^{N} h_i^{-m} \int_{x_i}^{x_{i+1}} |G(x_j,t)|dt.$$

To derive a lower bound for $||(PA)^{-1}||_\infty$ it suffices to bound $||\underset{\sim}{\alpha}||_\infty / ||PA\underset{\sim}{\alpha}||_\infty$ below for a particular $\underset{\sim}{\alpha}$ associated with any sufficiently dense mesh Δ. Recall that we are only considering meshes Δ with h sufficiently small that a unique collocation solution exists for any f(x). Thus, it is sufficient to find f(x) for which (2.1), (2.2) has as its solution the collocation solution $u_\Delta(x)$ which corresponds to $\underset{\sim}{\alpha}$ and then bound $||\underset{\sim}{\alpha}||_\infty / ||P\underset{\sim}{f}||_\infty$ below.

Let Q be the interpolatory Hermite projector on $P_{2m,\Delta}$, i.e. on each subinterval (x_i, x_{i+1}) QF is a polynomial of order 2m and for $x_j \in \Delta$

$$(4.35) \qquad D^{\ell-1}(QF)(x_j) = D^{\ell-1}F(x_j) \qquad\qquad 1 \le \ell \le m.$$

It is known [21] that for F piecewise smooth

$$(4.36) \qquad \max_{x_i \le x \le x_{i+1}} |D^\ell (1-Q)F| \le K h_i^{2m-\ell} \max_{x_i \le x \le x_{i+1}} |D^{2m}F| \qquad 0 \le \ell \le 2m.$$

Define now $v_\Delta(x) \in P_{2m,\Delta} \cap C^{(m-1)}[a,b]$ by

$$(4.37) \qquad v_\Delta(x) := \int_a^b Q\, G(x,t) g_\Delta(t) dt$$

where Q operates on G(x,t) as a function of x and

$$(4.38)\qquad g_\Delta(t) := h_i^{-m} \operatorname{sgn} G(x_\nu,t), \qquad t\in[x_i,x_{i+1}],$$

with ν chosen such that

$$(4.39)\qquad \sum_{i=1}^{N} h_i^{-m} \int_{x_i}^{x_{i+1}} |G(x_\nu,t)|\,dt = \max_{x_j\in\Delta} \sum_{i=1}^{N} h_i^{-m} \int_{x_i}^{x_{i+1}} |G(x_j,t)|\,dt.$$

If $f(x) := Lv_\Delta(x)$, then the unique collocation solution of $Lu(x) = f(x)$ with homogeneous BC (2.2) must be $u_\Delta(x) = v_\Delta(x)$, since $2m \le k+m$. Let the scaled coefficients $\underset{\sim}{\alpha}$ correspond to $v_\Delta(x)$. Then from previous arguments

$$(4.40)\qquad \frac{||\underset{\sim}{\alpha}||_\infty}{||\underset{\sim}{Pf}||_\infty} \sim \frac{||\underset{\sim}{\alpha}||_\infty}{\max\limits_{1\le\ell\le N} h_\ell^m \max\limits_{x_\ell\le x\le x_{\ell+1}} |Lv_\Delta(x)|} \ge \frac{\max\limits_{x_j\in\Delta} |v_\Delta(x_j)|}{\max\limits_{1\le\ell\le N} h_\ell^m \max\limits_{x_\ell\le x\le x_{\ell+1}} |Lv_\Delta(x)|}$$

Since

$$(4.41)\qquad Lv_\Delta(x) = g_\Delta(x) - L\int_a^b (1-Q)G(x,t)g_\Delta(t)dt$$

$$= g_\Delta(x) - \sum_{i=1}^{N} h_i^{-m} L \int_{x_i}^{x_{i+1}} (1-Q)G(x,t) \operatorname{sgn} G(x_\nu,t)dt,$$

we obtain for $x\in(x_\ell,x_{\ell+1})$,

$$(4.42)\qquad h_\ell^m|Lv_\Delta(x)| \le 1+h_\ell^m \sum_{i=1}^{N} h_i^{-m} \left|L \int_{x_i}^{x_{i+1}} (1-Q)G(x,t) \operatorname{sgn} G(x_\nu,t)dt\right|.$$

It then follows from (4.36) that for $x\in(x_\ell,x_{\ell+1})$

$$(4.43)\qquad h_\ell^m|Lv_\Delta(x)| \le 1+h_\ell^m \sum_{i=1}^{N} h_i^{-m}\cdot Kh_i^{m+1} \le 1+Kh_\ell^m .$$

Finally,

$$v_\Delta(x_j) = \int_a^b G(x_j,t)g_\Delta(t)dt = \sum_{i=1}^{N} h_i^{-m} \int_{x_i}^{x_{i+1}} G(x_j,t)\ \mathrm{sgn}\ G(x_\nu,t)dt$$

implies, by (4.39),

$$\max_{x_j\in\Delta} |v_\Delta(x_j)| \geq |v_\Delta(x_\nu)| = \sum_{i=1}^{N} h_i^{-m} \int_{x_i}^{x_{i+1}} |G(x_\nu,t)|dt \tag{4.44}$$

$$= \max_{x_j\in\Delta} \sum_{i=1}^{N} h_i^{-m} \int_{x_i}^{x_{i+1}} |G(x_j,t)|dt.$$

Inequalities (4.43) and (4.44) substituted into (4.40) produce the required lower bound and lead us to expect that (4.28) holds. Again, numerical evidence supporting the above arguments is given in Section 6.

If condensation of parameters is used then some minor changes in the above arguments are necessary. The most important of these changes is that the largest entry in magnitude of each row of A has to be determined. Without condensation, this is clearly $h_i^{-m}(\sigma_{\ell\ell}+O(h_i))$ as mentioned above; with condensation it is less clear. However, it seems unlikely that terms containing the h_i^{-m} factors would all cancel out during the condensation process. Thus, we expect similar scale factors in P, and (4.28) should hold for this case too. All numerical examples we have tried support this.

A similar analysis sheds light on the condition numbers for the matrices arising from the B-spline representation. Again, the critical quantities are the scale factors in P. The B-splines pertinent to block V_i have support (x_{i-1},x_{i+1}), (x_i,x_{i+1}), or (x_i,x_{i+2}). An analysis of the algorithm [1] for

generating B-splines indicates that we should expect $(h_{i-1}+h_i)^{-m}$, h_i^{-m}, and $(h_i+h_{i+1})^{-m}$ factors to arise from $L\psi_r(\xi_{ij})$ for these basis functions. The dominant one is h_i^{-m}, so again we expect (4.28) to apply. The numerical evidence in Section 6 agrees with this. Recently, Jespersen [13] has given rigorous proofs concerning the behaviour of $\kappa(PA)$ on the mesh for several special cases of (2.1) when B-splines are used. These are consistent with (4.28).

5. MONOMIAL BASES

In this section we consider local basis representations for which continuity is not built-in. Emphasis is placed upon local monomial basis representations, for which we choose the monomials $t^{j-1}/(j-1)!$, $1 \le j \le m$, as the first m canonical polynomials on [0,1] (cf. the beginning of Section 4). Thus we write for $x_i \le x \le x_{i+1}$, $1 \le i \le N$,

$$(5.1) \qquad u_\Delta(x) = \sum_{j=1}^{m} z_{ij}(x-x_i)^{j-1}/(j-1)! + h_i^m \sum_{j=1}^{k} w_{ij}\phi_j\left(\frac{x-x_i}{h_i}\right)$$

where $\{\phi_j\}_{j=1}^k$ are the remaining canonical polynomials of order k+m on [0,1], satisfying

$$(5.2) \qquad D^{\ell-1}\phi_j(0) = 0 \qquad \ell = 1,\ldots,m, \quad j = 1,\ldots,k.$$

The scale factor h_i^m for w_{ij} in (5.1) is only introduced for notational convenience, as we shall see later.

The relation between $\{w_{ij}\}_{j=1}^k$ and $u_\Delta(x)$ depends on the specific choice of $\{\phi_j\}_{j=1}^k$. For instance, the choice

$$(5.3) \qquad D^m\phi_j(\rho_r) = \delta_{jr} \qquad 1 \le j, \quad r \le k$$

implies $w_{ij} = D^m u_\Delta(\xi_{ij})$. Even though this choice is not included in the family defined by (4.1) - (4.3), it is readily seen to be well-defined and for m = 1 yields an implicit Runge-Kutta formula for $u_\Delta(x_{i+1})$ (see Ascher-Weiss [4]). Another choice,

$$(5.4) \qquad \phi_j(t) = t^{m+j-1}/(m+j-1)! \,, \qquad 1 \le j \le k,$$

yields the local monomial representation considered by Osborne [17]. Others have used the latter local representation as well (see Russell-Shampine [19]), but to our knowledge not with the efficient implementation considered here. For this choice,

$$(5.5) \qquad w_{ij} = h_i^{j-1} D^{m+j-1} u_\Delta(x_i^+) \qquad 1 \le j \le k.$$

For the monomial representation given by (5.1), the continuity conditions become

$$(5.6) \qquad \underset{\sim}{z}_{i+1} = C_i \underset{\sim}{z}_i + D_i \underset{\sim}{w}_i \qquad 1 \le i \le N$$

where $\underset{\sim}{w}_i = (w_{i1},\ldots,w_{ik})^T$, $C_i = (C^i_{rj})$ is an mxm upper triangular matrix with entries

$$(5.7) \qquad C^i_{rj} = h_i^{j-r}/(j-r)! \qquad j \ge r,$$

and $D_i = (D^i_{rj})$ is an mxk matrix with entries

$$(5.8) \qquad D^i_{rj} = h_i^{m+1-r} D^{r-1}\phi_j(1).$$

Since

$$(5.9) \qquad Lu_\Delta(x) = h_i^m \sum_{j=1}^{k} w_{ij} L[\phi_j(\frac{x-x_i}{h_i})] - \sum_{\ell=1}^{m} c_\ell(x) \sum_{j=\ell}^{m} z_{ij}(x-x_i)^{j-\ell}/(j-\ell)! \,,$$

the collocation conditions (C3) give

$$(5.10) \qquad H_i z_i + G_i w_i = f_i \qquad 1 \le i \le N,$$

where $f_i = (f(\xi_{i1}),\ldots,f(\xi_{ik}))^T$, $H_i = (H^i_{rj})$ is a kxm matrix with entries

$$(5.11) \qquad H^i_{rj} = -\sum_{\ell=1}^{j} c_\ell(\xi_{ir})(h_i\rho_r)^{j-\ell}/(j-\ell)! \ ,$$

and $G_i = (G^i_{rj})$ is a kxk matrix with entries

$$(5.12) \qquad G^i_{rj} = D^m\phi_j(\rho_r) - \sum_{\ell=1}^{m} c_\ell(\xi_{ir})h_i^{m+1-\ell}D^{\ell-1}\phi_j(\rho_r).$$

Thus the blocks V_i ($1 \le i \le N$) of A of (2.11) are (k+m) x (k+2m) and have the structure

$$(5.13) \qquad V_i = \begin{bmatrix} H_i & G_i & 0 \\ & & \\ -C_i & -D_i & I \end{bmatrix}$$

where I is an mxm identity matrix.

For the efficient assembly of V_i we compute and store once all mesh independent values like $D^\ell\phi_j(\rho_r)$, $D^\ell\phi_j(1)$, ρ_r^j and $j!$. Then the assembly algorithm for each i, $1 \le i \le N$, is

(5.14) Generate $\{h_i^j, h_i^j/j!,\ 1 \le j \le m\}$

```
FOR r:=1,...,m DO:
    FOR j:=r,...,m DO:
        C^i_rj := h_i^(j-r)/(j-r)!
    FOR j:=1,...,k DO:
        D^i_rj := h_i^(m+1-r) * D^(r-1) φ_j(1)
FOR r:=1,...,k DO:
    Generate {ρ_r^j * (h_i^j/j!), 1≤j≤m-1}
    FOR j:=1,...,m DO:
        H^i_rj := 0
        FOR ℓ:=1,...,j DO:
            H^i_rj := H^i_rj - c_ℓ(ξ_ir)*[(ρ_r h_i)^(j-ℓ)/(j-ℓ)!]
    Generate {c_ℓ(ξ_ir) h_i^(m+1-ℓ), 1≤ℓ≤m}
    FOR j:=1,...,k DO:
        G^i_rj := D^m φ_j(ρ_r)
        FOR ℓ:=1,...,m DO
            G^i_rj := G^i_rj - c_ℓ(ξ_ir) h_i^(m+1-ℓ) * D^(ℓ-1) φ_j(ρ_r)
```

This implies

(5.15) $$W_1 = [mk^2 + \tfrac{1}{2}\, m(m+7)k + 2m]N,$$

which is slightly less than for the Hermite-type representation in (4.17) and significantly less that for the B-splines in (3.2).

The storage requirements for V_i are considerably greater than in previous sections, however, so we consider further only the case where condensation of

parameters is performed on V_i by removing columns m+1 to m+k (corresponding to the unknowns $\underset{\sim}{w}_i$) and rows 1 to k. Because of the choice of the scaling factor h_i^m in (5.1), G_i is an $O(h_i)$ perturbation of the mesh independent matrix $(D^m\phi_j(\rho_r))$. This latter matrix is an identity matrix for the choice (5.3) and a scaled Vandermonde matrix for the choice (5.4). We require that $(D^m\phi_j(\rho_r))$ be nonsingular for any acceptable choice of $\{\phi_j\}_{j=1}^k$ and hence have $\kappa(G_i)$ bounded independently of Δ for h sufficiently small. Thus we may proceed to eliminate $\underset{\sim}{w}_i$, viz.

$$\underset{\sim}{w}_i = G_i^{-1}\underset{\sim}{f}_i - G_i^{-1}H_i\underset{\sim}{z}_i. \tag{5.16}$$

Substituting this into the continuity conditions (5.6) gives

$$\underset{\sim}{z}_{i+1} = \Gamma_i\underset{\sim}{z}_i + \underset{\sim}{g}_i \qquad 1 \le i \le N \tag{5.17}$$

where

$$\Gamma_i := C_i - D_iG_i^{-1}H_i, \qquad \underset{\sim}{g}_i := D_iG_i^{-1}\underset{\sim}{f}_i. \tag{5.18}$$

The coefficient matrix A corresponding to the BC and (5.17), written in the form (2.11), now has blocks V_i of size mx2m with offset m and

$$V_i = [-\Gamma_i \; I]. \tag{5.19}$$

Also $\hat{B}_a = B_a$, $\hat{B}_b = B_b$.

The operation count for condensation followed by solution of the linear

system is

$$(5.20)\qquad W_2 = [\tfrac{1}{3}k^3 + (m+1)k^2 + (m^2+m-\tfrac{1}{3})k + \tfrac{5}{6}m^3 + \tfrac{3}{2}(m_a+1)m^2 + \\ + (\tfrac{3}{2}m_a+\tfrac{5}{3})m]N$$

If only the superconvergent values $\{z_i\}_{i=1}^{N+1}$ are desired then

$$(5.21)\qquad S = 2m^2N.$$

If the values $\{w_i\}_{i=1}^{N}$ are also desired, so that $u_\Delta(x)$ can be recovered, then $G_i^{-1}f_i$ and $G_i^{-1}H_i$ must be saved, requiring $(m+1)kN$ additional storage locations.

The chief advantage of the monomial representation over the Hermite-type representation is its increased stability, especially when the mesh is highly nonuniform. As mentioned above, the condensation process is done entirely locally and is independent of the mesh Δ. We now examine the condition number of the condensed matrix A. From (5.7) and (5.8), $C_i = I + O(h_i)$ and $||D_i||_\infty = O(h_i)$, so $\Gamma_i = I + O(h_i)$ in (5.18). Thus, for h sufficiently small, A of (2.11), (5.19) is well-balanced and

$$(5.22)\qquad ||A||_\infty = \max\{2+O(h),\ ||B_a||_\infty,\ ||B_b||_\infty\}.$$

On $[x_i,x_{i+1}]$, let $\{\theta_j(x_i;x)\}_{j=1}^{m}$ be the set of linearly independent solutions of $Lu = 0$, subject to the initial conditions

$$(5.23)\qquad D^{\ell-1}\theta_j(x_i;x_i) = \delta_{j\ell} \qquad 1 \le j,\ \ell \le m.$$

If $u(x)$ is the solution to (2.1) with $f \equiv 0$, then

$$\underset{\sim}{z}(u(x_{i+1})) = \Theta(x_i;x_{i+1})\underset{\sim}{z}(u(x_i)) \qquad 1 \le i \le N \tag{5.24}$$

where $\Theta(x_i;x)$ is the fundamental matrix

$$\Theta(x_i;x) := (\underset{\sim}{z}(\theta_1(x_i;x)),\ldots,\underset{\sim}{z}(\theta_m(x_i;x))). \tag{5.25}$$

On the other hand, by (5.17)

$$\underset{\sim}{z}(u(x_{i+1})) = \Gamma_i\underset{\sim}{z}(u(x_i)) + \underset{\sim}{\tau}_i \qquad 1 \le i \le N \tag{5.26}$$

where the local truncation error $\underset{\sim}{\tau}_i$ satisfies $||\underset{\sim}{\tau}_i|| = O(h_i^{k+n+1})$ (n as in (2.9)) by an analysis similar to that in de Boor-Swartz [6]. Since (5.24), (5.26) hold for arbitrary vectors $\underset{\sim}{z}(u(x_i))$, $\Gamma_i = \Theta(x_i;x_{i+1}) + O(h_i^{k+n+1})$. By standard arguments for a multiple shooting matrix (Mattheij [15], Osborne [16]) and (2.4),

$$\kappa(A) = O(N). \tag{5.27}$$

Compared to the estimate (4.28) for the bases in the previous sections, the estimate (5.27) is very small. In particular, it is independent of $\underline{h}$.

The above argument on conditioning assumes that $O(h_i)$ terms can be neglected. While this is always true for sufficiently fine meshes, it is not always the case in practical situations, e.g. for singular perturbation problems. In such situations, instead of using (5.18)-(5.19) it may be desirable to

eliminate w_i and generate the new mx2m block V_i by decomposing the entire block $\begin{bmatrix} G_i \\ -D_i \end{bmatrix}$. Osborne [17] suggests performing a QR decomposition of $\begin{bmatrix} G_i \\ -D_i \end{bmatrix}$, and we have indeed found his strategy to produce smaller roundoff errors than (5.18)-(5.19) in some isolated cases.

Recall from the previous section that when a local basis representation is used and A includes the (collocation) discretization equations with the corresponding high order derivative approximations, $\kappa(A)$ grows rapidly as $h \to 0$ (see (4.28)). For the monomial basis (5.1), the condensed system has the form of a finite difference formula for the first order system corresponding to (2.1), with unknowns $\underset{\sim}{z}(u(x))$ [17], from which one might expect (5.27). Indeed, the whole process of condensation of parameters with the monomial basis may be regarded as a way to construct a very efficient and nonobvious finite difference scheme for the equivalent first order system. For instance, it can be verified that the difference scheme (5.17) satisfies the consistency requirements

$$h_i^{-1}(\Gamma^i_{rj} - \delta_{rj}) \underset{h_i \to 0}{\to} \begin{cases} c_j(x_i) & r = m \\ 1 & 1 \le j = r + 1 \le m \\ 0 & \text{otherwise} \end{cases} \tag{5.28}$$

and

$$h_i^{-1} \underset{\sim}{g}_i \underset{h_i \to 0}{\to} (0,\ldots,0,f(x_i))^T. \tag{5.29}$$

The choice of the monomials $t^{j-1}/(j-1)!$ $(1 \le j \le m)$ in (5.1) is a special one, and it is natural to ask whether other polynomial representations, with continuity not built in, can be used. Indeed, any local representation

$$(5.30) \qquad u_\Delta(x) = \sum_{j=1}^{m} z_{ij} h_i^{j-1} \psi_j\left(\frac{x-x_i}{h_i}\right) + h_i^m \sum_{j=1}^{k} w_{ij} \phi_j\left(\frac{x-x_i}{h_i}\right)$$

would yield, in exact arithmetic, precisely the same difference scheme (5.17), provided that $\psi_1(t),\ldots,\psi_m(t),\phi_1(t),\ldots,\phi_k(t)$ are k+m linearly independent polynomials of order k+m on [0,1] satisfying (5.2) and

$$(5.31) \qquad D^{\ell-1}\psi_j(0) = \delta_{j\ell}, \qquad 1 \le j, \quad \ell \le m.$$

However, representation (5.1) should be employed, as argued below.

For any representation $\psi_1(t),\ldots,\psi_m(t)$ we can write

$$(5.32) \qquad \psi_j(t) = \frac{t^{j-1}}{(j-1)!} + (\phi_1(t),\ldots,\phi_k(t))Q\underset{\sim}{P}^j \qquad 1 \le j \le m$$

where $\underset{\sim}{P}^j$ is the j-th column of the kxm matrix

$$(5.33) \qquad P = \begin{bmatrix} D^m\psi_1(\rho_1) & \cdots & D^m\psi_m(\rho_1) \\ \vdots & & \vdots \\ D^m\psi_1(\rho_k) & \cdots & D^m\psi_m(\rho_k) \end{bmatrix}$$

and the kxk matrix Q is defined by

$$(5.34) \qquad Q = \begin{bmatrix} D^m\phi_1(\rho_1) & \cdots & D^m\phi_k(\rho_1) \\ \vdots & & \vdots \\ D^m\phi_1(\rho_k) & \cdots & D^m\phi_k(\rho_k) \end{bmatrix}^{-1}$$

Note that Q is well-defined because the linearly independent polynomials $\phi_1(t),\ldots,\phi_k(t)$ satisfy (5.2). For the same reason, the matrix G_i constructed as in (5.12) is nonsingular for h_i sufficiently small.

Now, the continuity and collocation equations read

$$\bar{C}_i z_i + D_i w_i = z_{i+1}, \tag{5.35}$$

$$\bar{R}_i z_i + G_i w_i = f_i, \tag{5.36}$$

where by (5.32),

$$\bar{C}_i = C_i + h_i^{-m} D_i Q P R_i , \qquad R_i = \mathrm{diag}\{h_i^{j-1}\}_{j=1}^{m}$$
$$\bar{H}_i = H_i + h_i^{-m} G_i Q P R_i \tag{5.37}$$

and C_i, D_i, H_i and G_i are as for (5.1), see (5.7), (5.8), (5.11) and (5.12). It then follows that

$$\Gamma_i = \bar{C}_i - D_i G_i^{-1} \bar{H}_i = (C_i - D_i G_i^{-1} H_i) + h_i^{-m} D_i Q P R_i - h_i^{-m} D_i Q P R_i \tag{5.38}$$
$$= C_i - D_i G_i^{-1} H_i .$$

Thus, in exact arithmetic, any representation (5.30) is equivalent to (5.1), producing a matrix A whose condition number satisfies (5.27). (Particularly attractive would be a Hermite-type representation which gives $\bar{C}_i = 0$, $D_i = h_i^m R_i^{-1}[0,I]$). However, unless (5.1) is used, the process is subject

to a severe cancellation error, because when h_i is small H_i is much smaller element-wise than the other term defining $\bar{H}_i$ in (5.37). This larger term is subsequently cancelled out in (5.38). For the representation (5.1), P = 0 and so this cancellation error does not occur.

6. NUMERICAL EXAMPLES AND CONCLUSIONS

Before presenting specific examples, we must emphasize that usually the observed error with all the bases considered is the same when the mesh is not highly non-uniform or unnecessarily dense. In such situations the truncation error dominates; to see roundoff error effects it is necessary to use very fine meshes or quite nonuniform ones. Expecting a basis to perform well under such circumstances is not entirely unreasonable, however, as a general purpose code must be designed to robustly handle these situations. Highly nonuniform meshes do arise naturally in the solution of singular perturbation problems [4].

In the tables which follow we use the notation $D^{\ell}e_b$ for the maximum magnitude error $|D^{\ell}(u-u_{\Delta})|$ at the mesh points when basis b is used. The possible subscripts b are

B - B-spline basis, as in Section 3,

H1 - Hermite representation, with $\underset{\sim}{\eta}$ as in (4.24a),

H2 - Hermite representation, with $\underset{\sim}{\eta}$ as in (4.24b),

H3 - Hermite representation, with $\underset{\sim}{\eta}$ as in (4.24c),

and M - monomial representation (5.1) with $\phi_j(t)$ as in (5.4).

The infinity-norm condition numbers of the coefficient matrices for basis b are denoted by $cond_b$. These coefficient matrices are explicitly row-equilibrated so that the computed condition numbers reflect the actual loss of accuracy. In practice, of course, such explicit row-scaling can be avoided, provided that Gauss elimination with <u>scaled</u> partial pivoting is used. The notation .5-4 for $.5\text{x}10^{-4}$ is used. All results are for Gauss-Legendre collocation points, and condensation of parameters as described in Sections 4 and 5 is used for the Hermite and monomial representations, respectively. The computations were performed on an Amdahl V/6 II or V/8 in double precision (14 hexadecimal digits) at the University of British Columbia. Many examples were run with the different

basis representations for various choices of partitions and values of k and m; a representative selection of results follows.

Example 1 [2] The problem

$$(6.1) \qquad D^2(x^3D^2u) = 1 \qquad 1 \le x \le 2,$$

$$(6.2) \qquad u(1) = D^2u(1) = u(2) = D^2u(2) = 0$$

has the exact solution

$$(6.3) \qquad u(x) = \frac{1}{4}(10 \ln 2-3)(1-x) + \frac{1}{2}\left[\frac{1}{x} + (3+x)\ln x - x\right].$$

Table 1 gives the condition numbers for the coefficient matrices for various bases when uniform meshes are used. Only cond_{H1} is given for the Hermite representations since the condition numbers for H2 and H3 are identical. For uniform meshes, (5.27) and (4.28) predict that as N is doubled the condition numbers for the monomial representation should double while those for the other representations should increase by a factor of 16. The calculations support this. Table 2 contains errors for the various representations using uniform meshes. Note that the monomial representation is the most accurate; also, of the three Hermite representations, H1 is the best.

To study the effect of non-uniformity we use the following sequence of partitions: Δ_1 consists of 4 equally spaced points plus $x_4 = 1.51$, so $\underline{h} = 10^{-2}$; Δ_2 has 8 equally spaced points plus $x_6 = 1.501$ so $\underline{h} = 10^{-3}$; and Δ_3 has 16 equally spaced points plus $x_{10} = 1.5001$ so $\underline{h} = 10^{-4}$. Equation (5.27) predicts that cond_M should double in each case, while (4.28) says that the condition numbers for the Hermite and B-spline representations should increase by a factor of 10^3 (from $\underline{h}^{1-m}$) in each case. The data in Table 3 indicates this, and also shows how these condition numbers are reflected in the errors.

Table 1. Condition numbers for example 1 on uniform meshes

k	N	$cond_B$	$cond_{H1}$	$cond_M$
4	4	.61+4	.48+4	.73+2
	8	.89+5	.53+5	.12+3
	16	.14+7	.85+6	.23+3
6	4	.20+5	.87+4	.73+2
	8	.29+6	.10+6	.12+3
	16	.44+7	.17+7	.23+3

Table 2. Errors for example 1 on uniform meshes.

k =	4		6		
N =	8	16	4	8	16
e_B =	.60-11	.13-12	.11-13	.43-13	.79-12
e_{H1} =	.60-11	.42-13	.10-13	.75-14	.55-13
e_{H2} =	.60-11	.42-13	.25-13	.53-12	.91-11
e_{H3} =	.60-11	.42-13	.11-13	.46-13	.14-11
e_M =	.60-11	.24-13	.96-14	.18-15	.18-15
D^2e_B =	.11-12	.17-11	.23-13	.44-12	.82-11
D^2e_{H1} =	.34-13	.23-12	.85-14	.72-13	.54-12
D^2e_{H2} =	.34-13	.23-12	.35-12	.57-11	.97-10
D^2e_{H3} =	.34-13	.23-12	.35-13	.64-12	.17-10
D^2e_M =	.16-15	.18-15	.16-15	.16-15	.18-15

Table 3. Condition numbers and errors for example 1 on non-uniform meshes.

k =	4			6		
partition =	Δ_1	Δ_2	Δ_3	Δ_1	Δ_2	Δ_3
$cond_B$ =	.22+8	.20+11	.20+14	.91+8	.85+11	.83+14
$cond_{H1}$ =	.23+8	.20+11	.20+14	.47+8	.40+11	.40+14
$cond_M$ =	.88+2	.14+3	.24+3	.88+2	.14+3	.24+3
e_B =	.13-8	.56-8	.21-5	.34-10	.31-7	.71-4
e_{H1} =	.13-8	.71-10	.12-5	.21-11	.82-9	.30-5
e_M =	.13-8	.60-11	.24-13	.96-14	.18-15	.18-15
D^2e_B =	.25-10	.63-7	.24-4	.37-9	.35-6	.12-3
D^2e_{H1} =	.80-12	.76-9	.13-4	.23-10	.92-8	.33-4
D^2e_M =	.14-15	.16-15	.19-15	.16-15	.18-15	.19-15

For the remaining examples, the solutions are low order piecewise polynomials so all observed errors are due to roundoff. As in the previous example, H1 is the most accurate of the three Hermite representations; for brevity results for H2 and H3 entries are omitted.

Example 2: The problem

$$u'' - 4u = 16x + 12x^2 - 4x^4 \qquad 0 \leq x \leq 1 \tag{6.4}$$

$$u(0) = u'(1) = 0 \tag{6.5}$$

has the exact solution

$$u(x) = x^4 - 4x. \tag{6.6}$$

For uniform meshes, doubling N quadruples the condition numbers associated with Hermite and B-spline representations but merely doubles those for monomials (see Table 4). The results in Table 5 show that again the monomial representation is most accurate for a given partition.

To study the validity of the condition number estimate (4.28), solutions are computed corresponding to the following partitions:

$$\Delta_1 = \langle 0,\ 10^{-4},\ .25,\ .5,\ .75,\ 1\rangle \tag{6.7a}$$

$$\Delta_2 = \langle 0,\ 10^{-6},\ .25,\ .5,\ .75,\ 1\rangle \tag{6.7b}$$

(6.7c) $\Delta_3 = \langle 0, .25, .5, .75, 1\text{-}10^{-4}, 1\rangle$

(6.7d) $\Delta_4 = \langle 0, .25, .5, .75, 1\text{-}10^{-6}, 1\rangle$.

The Green's function for this example is

$$(6.8)\qquad G(x,t) = \begin{cases} \sinh 2x \cosh 2(1-t)/[2 \cosh 2] & x \le t \\ \sinh 2t \cosh 2(1-x)/[2 \cosh 2] & x \ge t. \end{cases}$$

For Δ_1 and Δ_2 $G(\cdot,t) = O(\underline{h})$ on $[0,x_2]$ so $\max_j \sum_{i=1}^{N} h_i^{-2} \int_{x_i}^{x_{i+1}} G(x_j,t)dt = O(1)$. The prediction from (4.28) is borne out by the calculations summarized in Tables 4 and 5: for the Hermite and B-spline representations condition numbers and errors are essentially unchanged between Δ_1 and Δ_2. This is in sharp contrast to Δ_3 and Δ_4 which have the same h and $\underline{h}$ as Δ_1 and Δ_2, respectively. Moving the extremely narrow subinterval to the right of [0,1] where $G(\cdot,t)$ is not small, has produced much larger condition numbers and much lower accuracy. Since $\underline{h}^{-2} \int_{1-\underline{h}}^{1} G(1,t)dt = O(\underline{h}^{-1})$, equation (4.28) predicts that condition numbers for Δ_4 should be 100 times those for Δ_3 when Hermites or B-splines are used; this is observed in Table 4. The monomial representation is essentially unaffected by the small $\underline{h}$ in all cases.

Table 4. Condition numbers for example 2 (k=4).

	N	$cond_B$	$cond_{H1}$	$cond_M$
uniform meshes	10	.42+3	.22+3	.20+2
	20	.17+4	.87+3	.34+2
	40	.69+4	.34+4	.64+2
	80	.28+5	.14+5	.12+3
non-uniform meshes	Δ_1	.68+2	.43+2	.17+2
	Δ_2	.68+2	.43+2	.12+2
	Δ_3	.96+5	.54+5	.12+2
	Δ_4	.96+7	.54+7	.12+2

Table 5. Errors for example 2 (k=4).

	N	e_B	e_{H1}	e_M	De_B	De_{H1}	De_M
uniform	10	.50-13	.21-13	.24-14	.10-12	.30-13	.38-14
	20	.26-12	.11-12	.33-14	.38-12	.14-12	.51-14
	40	.23-11	.72-13	.82-14	.35-11	.11-12	.20-13
	80	.68-11	.18-11	.13-13	.10-10	.26-11	.33-13
non-uniform	Δ_1	.10-13	.62-14	.67-15	.98-14	.16-13	.67-15
	Δ_2	.40-14	.58-14	.18-14	.67-14	.13-13	.89-15
	Δ_3	.41-10	.94-11	.18-14	.84-10	.21-10	.89-15
	Δ_4	.38-8	.15-8	.18-14	.78-8	.30-8	.89-15

Example 3: The problem

$$(6.9) \qquad D^3u = \begin{cases} 2 & 0 \le x \le \frac{1}{2} \\ 0 & \frac{1}{2} < x \le 1 \end{cases}$$

$$(6.10) \qquad u(0) = 1, \quad u'(0) = \tfrac{1}{4}, \quad u(1) = 25/24,$$

has the exact solution

$$(6.11) \qquad u(x) = \begin{cases} \frac{1}{3}x^3 - \frac{1}{2}x^2 + \frac{1}{4}x + 1 & 0 \le x \le \frac{1}{2} \\ 25/24 & \frac{1}{2} \le x \le 1. \end{cases}$$

All partitions used contain the point of discontinuity $x = \frac{1}{2}$, so the discretization error is zero. The problem is solved using Δ_1, Δ_2, Δ_3 and Δ_4 as in (6.7), and also using

$$(6.12a) \qquad \Delta_5 = \langle 0, .25, .5, .51, .75, 1 \rangle$$

$$(6.12b) \qquad \Delta_6 = \langle 0, .25, .5, .5001, .5002, .75, 1 \rangle$$

$$(6.12c) \qquad \Delta_7 = \langle 0, .25, .5, .500001, .500002, .500003, .500004, .75, 1 \rangle.$$

For this problem the Green's function is

$$(6.13) \qquad G(x,t) = \begin{cases} -(1-t)^2x^2/2 & x \le t \\ (t-t^2/2)x^2 - tx + t^2/2 & x \ge t. \end{cases}$$

Here, $G(\cdot,t) = O(\underline{h})$ on $[0,x_2]$ so the condition number estimate (4.28) indicates that $\kappa(PA) \sim O(\underline{h}^{-1})$ for Δ_1 and Δ_2. In contrast, $G(\cdot,t) = O(\underline{h}^2)$ on $[x_N,1]$ so we expect $\kappa(PA) \sim O(1)$ for Δ_3 and Δ_4 when Hermite or B-spline representations are used. This is apparent from the data in Table 6. For the final three partitions, (4.28) yields $\kappa(PA) \sim$ constant $\cdot \sum_i h_i^{1-m}$. Thus, in comparing Δ_5 to Δ_6 and Δ_6 to Δ_7 we expect condition numbers to grow like 2×10^4 in the first two columns of Table 6. This is indeed the case. Finally, observe that the monomial representation is very accurate for all of these partitions and that (5.27) is satisfied for the condition numbers.

Table 6. Condition numbers and errors for example 3 (k=6).

partition	$cond_B$	$cond_{H1}$	$cond_M$	e_B	e_{H1}	e_M
Δ_1	.48+6	.26+6	.43+2	.31-11	.17-12	.44-15
Δ_2	.48+8	.26+8	.43+2	.23-9	.20-9	.67-15
Δ_3	.69+3	.40+3	.39+2	.16-13	.42-14	.0
Δ_4	.69+3	.40+3	.39+2	.18-13	.40-14	.0
Δ_5	.14+6	.86+5	.41+2	.64-11	.15-11	.0
Δ_6	.30+10	.17+10	.48+2	.48-7	.32-7	.0
Δ_7	.63+14	.34+14	.62+2	.22-2	.13-3	.0

The above examples illustrate the stability advantage that the monomial representation has over the others. The conditioning of its coefficient matrix is unaffected by the distribution of points within the partition, in sharp contrast to the situation for the Hermite or B-spline representations. For the latter two, the nature of the dependence of the condition number on the partition imposes a limitation on the nonuniformity of meshes which can be used in practice. In particular, the numerical examples demonstrate the rapid growth in the condition number predicted by (4.28), and this growth causes the expected loss of accuracy due to roundoff errors ($\sim$ q digits when $\kappa(PA) \approx 10^q$). This difficulty can be overcome with monomials, and in higher precision arithmetic storage limits would probably be reached before conditioning problems arose. In fact, the monomial representation seems so stable that it should be possible to safely solve many boundary value problems to modest accuracies with single precision even for machines with short word lengths.

The highly nonuniform meshes used in our examples are quite extreme in their mesh ratio $h/\underline{h}$. However, very large mesh ratios can occur in practice as well, see, e.g., [1], [2], [4].

In Table 7 we have summarized the work estimates W_1 and W_2 and the storage S for the various representations. It is assumed that $m_a = m/2$, and the factor N has been omitted from all entries. For the Hermite or B-spline representation, W_2 and S are given in three cases: no condensation of parameters, condensation but with the entire solution $\underset{\sim}{\alpha}$ computed so that $u_\Delta(x)$ itself is available (full), and condensation with only superconvergent quantities $\underset{\sim}{z}_i$ computed (partial). Similarly, the monomial representation is considered for these two distinct implementations of condensation, full or partial, depending on whether or not $\underset{\sim}{w}_i$ is explicitly calculated. Recalling that W_1

for B-splines is an underestimate of the amount of work actually needed, it is easily seen that the B-spline representation is by far the most expensive. The Hermite representation is the cheapest if full advantage is taken of condensation of parameters. In terms of programming ease and brevity of codes, there is little question that the Hermite and monomial representations are much preferred over B-splines.

In summary, in our context the monomial representation is much superior to B-splines both in terms of stability and efficiency, and its slight inferiority in efficiency relative to Hermite representations is more than compensated for by its robustness.

Table 7. Operation counts and storage requirements (N factor omitted).

	m =	2			4	
	k =	2	4	6	4	6
W_1	B	$85\frac{1}{3}$	$282\frac{2}{3}$	592	$853\frac{1}{3}$	1712
	H	34	90	170	204	364
	M	30	72	130	160	284
W_2	B, H (no cond.)	25	84	195	144	302
	cond/full	25	84	195	144	302
	cond/partial	25	58	135	144	239
	M full	55	125	251	340	538
	partial	51	117	239	324	514
S	B, H (no cond.)	8	24	48	32	60
	cond/full	8	20	40	32	52
	cond/partial	8	8	8	32	32
	M full	14	20	26	52	62
	partial	8	8	8	32	32

7. REFERENCES

1. U. Ascher, J. Christiansen, and R.D. Russell, A collocation solver for mixed order systems of boundary value problems. Math. Comp. 33 (1979), pp. 659-679.

2. U. Ascher, J. Christiansen, and R.D. Russell, Collocation software for boundary value ODE's, ACM Trans. on Math. Software 7 (1981), pp. 209-229.

3. U. Ascher and R.D. Russell, Evaluation of B-splines for solving systems of boundary value problems, Comp. Sci. Tech. Rep. 77-14, University of British Columbia, 1977.

4. U. Ascher and R. Weiss, Collocation for singular perturbation problems I: First order systems with constant coefficients, Comp. Sci. Tech. Rep. 81-2, Univ. of B.C., 1981.

5. C. de Boor, A Practical Guide to Spines, Applied Math. Sciences Vol. 27, Springer-Verlag, New York, 1978.

6. C. de Boor and B. Swartz, Collocation at Gaussian points, SIAM J. Numer. Anal. 10 (1973), pp. 582-606.

7. C. de Boor and B. Swartz, Comment on "a comparison of global methods for linear two-point boundary value problems", Math. Comp. 31 (1977), pp. 916-921.

8. C. de Boor and R. Weiss, SOLVEBLOK: A package for solving almost block diagonal linear systems, ACM Trans. on Math. Software 6 (1980), 80-87.

9. J. Christiansen, private communication.

10. S.D. Conte and C. de Boor, Elementary Numerical Analysis: An Algorithmic Approach, 3rd edition, McGraw-Hill, New York, 1980.

11. J.C. Diaz, G. Fairweather, and P. Keast, Fortran packages for solving almost block diagonal linear systems by modified alternate row and column elimination, Tech. Report 148-81, Dept. of Computer Science, Univ. of Toronto, 1981.

12. E. Grosse, Ph.D. Thesis, Dept. of Computer Science, Stanford University, 1980.

13. D. Jespersen, The condition number of collocation matrices, manuscript, Dept. of Math., Oregon State Univ., Corvallis, 1981.

14. G.K. Leaf, M. Minkoff, G.D. Byrne, D. Sorenson, T. Bleakney, and J. Saltzman, DISPL: A software package for one and two spatially dimensioned kinetic diffusion problems, Argonne Nat. Lab. Tech. Rep. ANL-77-12, 1977.

15. R.M.M. Mattheij, The conditioning of linear boundary value problems, Rep. 7927, Mathematisch Inst., Katholieke Univ., Nijmegen, 1979. To appear in SIAM J. Numer. Anal.

16. M.R. Osborne, Aspects of the numerical solution of boundary value problems with separated boundary conditions, working paper, Computing Research Group, Australian National University, Canberra.

17. M.R. Osborne, Collocation methods for boundary value problems, manuscript.

18. R.D. Russell, Efficiencies of B-spline methods for solving differential equations, Proc. Fifth Conference on Numerical Mathematics, Manitoba (1975), pp. 599-617.

19. R.D. Russell and L.F. Shampine, A collocation method for boundary value problems, Num. Math. 19 (1972), pp. 13-36.

20. N.L. Schryer, Numerical solution of time-varying partial differential equations in one space variable, Bell Lab. Comp. Sci. Rep. 53, 1976.

21. B.K. Swartz and R.S. Varga, Error bounds for spline and L-spline interpolation, J. Approx. Th., 6 (1972), pp. 6-49.