A NEW BASIS IMPLEMENTATION FOR A MIXED ORDER BOUNDARY VALUE ODE SOLVER

G. Bader* and U. Ascher**

Technical Report 85-11

ABSTRACT

The numerical approximation of mixed order systems of multipoint value ordinary differential equations by collocation requires appropriate representation of the piecewise polynomial solutions. B-splines were originally implemented in the general purpose code COLSYS, but better alternatives exist. One promising alternative was proposed by Osborne and discussed by Ascher, Pruess and Russell.

In this paper we analyze the performance of the latter solution representation for cases not previously covered, where the mesh is not necessarily dense. This analysis and other considerations have led us to implement a basis replacement in COLSYS, and we discuss some implementation details. Numerical results are given which demonstrate the improvement in performance of the code.

* Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, Canada V5A 1S6 and Inst. Angew. Mathematik, Universitat Heidelberg, D-6900 Heidelberg, Germany. Supported in part under NSERC Grant A4306 and the German Research Council.

** Department of Computer Science, University of British Columbia, Vancouver, British Columbia, Canada V6T 1W5 . Supported in part under NSERC (Canada) Grant A4306.

1. INTRODUCTION

When considering the numerical solution of mixed order systems of multipoint value ordinary differential equations (ODEs), one promising method is that of collocation. Piecewise polynomial collocation at Gaussian points was implemented in a general purpose code COLSYS (Ascher, Christiansen and Russell [5]) which has proved successful in many applications. Still, improvements to this code can be made in a number of aspects.

One such aspect is the representation used for the piecewise polynomial solutions. The B-spline basis (de Boor [8]) used for this purpose in COLSYS was shown by Ascher, Pruess and Russell [6] to be inferior, both in operation count and in conditioning, to a monomial representation proposed by M. Osborne[17], when applied to a linear, higher order ODE. Recently, parts of the collocation theory of de Boor and Swartz [9] for a two-point boundary value problem (BVP) for a higher order ODE were worked out using such a monomial basis (Ascher [1]), enabling a much simpler presentation as well as sharpening of some results. It was shown that the stability constant of the numerical scheme relates to the condition number of the associated first order BVP and to the number of mesh elements alone, for any mesh. Thus, a more effective control of a damped Newton iteration for some difficult nonlinear problems can be applied, because the BVP dependent variables are more directly controlled and certain superfluous ill-conditioning effects which are present in extreme cases when using B-splines are removed.

The analysis in [17,6,1] was performed for small h, in case of one higher order ODE. In this paper we discuss extensions of this analysis to stiff BVPs, for a higher order ODE as well as for a first order system of ODEs, and describe an implementation, imbedded in

COLSYS, of a monomial solution representation for multipoint value mixed order systems of ODEs. We give numerical examples comparing COLSYS before and after the change of basis. These examples show a gain in efficiency and robustness. The coding of the new representation is simpler as well.

When making this change in COLSYS, we have decided to keep using collocation at Gaussian points. One reason for this choice is that for first order systems we have algebraic stability, see Ascher and Bader [2]. In the analysis here we therefore consider mainly Gaussian points.

2. SOLUTION REPRESENTATION

For the presentation here, it is sufficient to consider two linear differential ODEs: a first order system of order n,

$$D\,\mathbf{y} = L(x)\mathbf{y} + \mathbf{q}(x) \qquad a < x < b \tag{2.1}$$

where $D \equiv \frac{d}{dx}$, $\mathbf{y}(x), \mathbf{q}(x) \in \mathbf{R}^n$, $L(x) \in \mathbf{R}^{n \times n}$, and a higher order ODE of order $m \geq 1$,

$$D^m u = \sum_{l=1}^{m} c_l(x) D^{l-1} u + q(x) \qquad a < x < b \tag{2.2}$$

where $D^l \equiv \frac{d^l}{dx^l}$. Of course, (2.2) can be considered as a special case of (2.1) with $n := m$ and

$$\mathbf{y}(x) := (u(x), Du(x), \ldots, D^{m-1}u(x)), \tag{2.3}$$

but in practice we apply collocation directly to the form (2.2).

Consider a k-stage collocation scheme for (2.2), $k \geq m$. Thus, there is a partition π of [a,b]

$$\pi : a = x_1 < x_2 < \cdots < x_N < x_{N+1} = b \tag{2.4}$$

$$h_i := x_{i+1} - x_i, \qquad h := \max_{1 \leq i \leq N} h_i$$

and k points

$$0 \leq \rho_1 < \rho_2 < \cdots < \rho_k \leq 1 \tag{2.5}$$

which we later choose to be Gaussian points. A piecewise polynomial function $u_\pi(x)$ is sought such that $u_\pi \in \mathbf{P}_{k+m,\pi} \cap C^{m-1}[a,b]$;[1] $u_\pi(x)$ satisfies m side conditions associated with (2.2) and given on a subset of π; and $u_\pi(x)$ satisfies the ODE (2.2) at the collocation points

$$x_{ij} = x_i + h_i \rho_j \qquad 1 \leq j \leq k, \ 1 \leq i \leq N. \tag{2.6}$$

This collocation procedure can of course be applied to systems. In particular, for (2.1) we have a piecewise polynomial vector function which satisfies (componentwise) $\mathbf{y}_\pi \in \mathbf{P}_{k+1,\pi} \cap C[a,b]$.

Let us now recall the monomial Runge-Kutta basis representation [6,1]. For a fixed mesh element $[x_i, x_{i+1}]$ the approximate solution is written as

$$u_\pi(x) = \sum_{j=1}^{m} \frac{(x-x_i)^{j-1}}{(j-1)!} y_{ij} + h_i^m \sum_{j=1}^{k} \psi_{mj}\left(\frac{x-x_i}{h_i}\right) z_{ij} \tag{2.7}$$

where $\psi_{mj} \in \mathbf{P}_{k+m}$ are defined by

---

[1] We say that v is in $\mathbf{P}_{k+m}$ if $v(x)$ is a polynomial of order $k+m$ (degree $\leq k+m$) on an appropriate interval, and that v is in $\mathbf{P}_{k+m,\pi}$ if $v(x)$ is a piecwise polynomial which is in $\mathbf{P}_{k+m}$ on each subinterval of the mesh π.

$$D^{l-1}\psi_{mj}(0) = 0 \qquad 1 \le l \le m \qquad 1 \le j \le k \tag{2.8a}$$

$$D^{m}\psi_{mj}(\rho_l) = \delta_{jl} \qquad 1 \le l \le k \tag{2.8b}$$

Thus

$$y_{ij} = D^{j-1}u_{\pi}(x_i) \qquad 1 \le j \le m \tag{2.9a}$$

$$z_{ij} = D^{m}u_{\pi}(x_{ij}) \qquad 1 \le j \le k. \tag{2.9b}$$

Note that the polynomials ψ_{mj} satisfy

$$\psi_{rj} \equiv D^{m-r}\psi_{mj} \qquad 1 \le r \le m, \quad 1 \le j \le k \tag{2.10}$$

which is rather useful when collocating for mixed order systems of ODEs. Note also that these functions are independent of the mesh element i.

Application of the collocation equations now yields k equations for $\mathbf{y}_i := (y_{i1},\ldots,y_{im})^T$ and $\mathbf{z}_i := (z_{i1},\ldots,z_{ik})^T$,

$$-V\mathbf{y}_i + W\mathbf{z}_i = \mathbf{q}_i \qquad 1 \le i \le N \tag{2.11}$$

where

$$V_{rj} = \sum_{l=1}^{j} \frac{c_l(x_{ir})(h_i\rho_r)^{j-l}}{(j-l)!} \qquad 1 \le r \le k,\ 1 \le j \le m \tag{2.12}$$

$$W_{rj} = \delta_{rj} - \sum_{l=1}^{m} c_l(x_{ir})h_i^{m+1-l}D^{l-1}\psi_{mj}(\rho_r) \qquad 1 \le r,\ j \le k \tag{2.13}$$

and

$$\mathbf{q}_i := (q(x_{i1}),\ldots,q(x_{ik}))^T. \tag{2.14}$$

Defining the k×k matrices $A^{(l)}$ by

$$A_{rj}^{(m-l)} := D^l \psi_{mj}(\rho_r) = \psi_{m-l,j}(\rho_r) \quad 1 \le r, j \le k, 0 \le l \le m \tag{2.15}$$

(note $A^{(0)} = I$) and the constants

$$\xi_l := c_l(x_i) h_i^{m+1-l} \qquad 1 \le l \le m, \tag{2.16}$$

we can write W, upon freezing $c_l(x)$ along the i-th element $[x_i, x_{i+1})$, as

$$W = I - \sum_{l=1}^{m} \xi_l A^{(m+1-l)}. \tag{2.17}$$

Next, we obtain m additional relations from the global continuity requirements,

$$C\mathbf{y}_i + D\mathbf{z}_i = \mathbf{y}_{i+1} \qquad 1 \le i \le N \tag{2.18}$$

where C is an $m \times m$ upper triangular matrix with entries

$$C_{rj} = \frac{h_i^{j-r}}{(j-r)!} \qquad j \ge r \tag{2.19}$$

and D is an $m \times k$ matrix with entries

$$D_{rj} = h_i^{m+1-r} D^{r-1} \psi_{mj}(1) \qquad 1 \le r \le m, 1 \le j \le k. \tag{2.20}$$

Defining the k-vectors $\mathbf{b}^{(l)}$ by

$$b_j^{(m-l)} := D^l \psi_{mj}(1) \qquad 1 \le j \le k, 0 \le l \le m-1 \tag{2.21}$$

and $\mathbf{b} := \mathbf{b}^{(1)}$, (note that $\mathbf{b}^{(1)}$ and $A^{(1)}$ are the usual Runge-Kutta coefficients for a 1st order ODE; see e.g. [2]) we claim

Lemma 2.22: Assume that the k points of (2.5) satisfy the orthogonality conditions

$$\int_0^1 \phi(t) \prod_{l=1}^{k} (t-\rho_l) dt = 0 \qquad \phi \in \mathbf{P}_m \tag{2.22a}$$

(this certainly holds for Gaussian points and $k \ge m$). Then

$$(\mathbf{b}^{(m+1-l)})^T = \mathbf{b}^T A^{(m-l)} \qquad 1 \le l \le m \tag{2.22b}$$

Hence

$$D = diag\{h_i^m, \ldots, h_i\} \begin{pmatrix} \mathbf{b}^T A^{(m-1)} \\ \vdots \\ \mathbf{b}^T A^{(1)} \\ \mathbf{b}^T A^{(0)} \end{pmatrix} \tag{2.22c}$$

Proof: The precision of the quadrature formula with $\mathbf{b}$ is guaranteed by (2.22a) to be at least $k+m$. Hence

$$\sum_{r=1}^{k} b_r D^l \psi_{mj}(\rho_r) = \int_0^1 D^l \psi_{mj}(t)dt = D^{l-1}\psi_{mj}(1)$$

□

The linear relations (2.11) and (2.18) are now used to eliminate $\mathbf{z}_i$ and express a relation between the solution values at the end points of the i-th element,

$$\Delta_i \mathbf{y}_{i+1} = \Gamma_i \mathbf{y}_i + \mathbf{r}_i \qquad 1 \le i \le N. \tag{2.23}$$

If W is nonsingular (which clearly holds for h_i sufficiently small) then

$$\Delta_i = I, \quad \Gamma_i = C + DW^{-1}V, \quad \mathbf{r}_i = DW^{-1}\mathbf{q}_i. \tag{2.24}$$

However, exceptional cases may occur if some ξ_l of (2.16) are not small. This is considered in the next section. The relations (2.23) together with m boundary conditions form a linear algebraic system of a familiar form for $u_\pi(x)$ and its first m-1 derivatives at mesh points. The solution of this system has been discussed elsewhere, see [3,6,10] and references therein.

The above description can be applied also for the first order system (2.1), taking $m = 1$. An implicit Runge-Kutta scheme results.

In this paper we concentrate on Gaussian points, for which

$$\rho_r = 1 - \rho_{k+1-r} \qquad 1 \le r \le k \tag{2.25}$$

and the obtained schemes are symmetric. This implies that, for $0 \le t \le 1$,

$$D^m \psi_{mj}(t) = D^m \psi_{m,k+1-j}(1-t) \qquad 1 \le j \le k. \tag{2.26}$$

Also,

$$\rho_1 > 0 \tag{2.27}$$

(unlike Lobatto schemes). Then it easily follows that the matrices $A^{(l)}$ are nonsingular, $0 \le l \le m$.

3. LOCAL PARAMETER ELIMINATION

From (2.16), (2.17) it follows immediately that for h_i sufficiently small, W is nonsingular and

$$W^{-1} = I + O(h_i).$$

Hence, the elimination of $\mathbf{z}_i$ as in (2.24) is sound and the obtained method is stable, with a stability constant comparable to the condition number of the BVP being solved [1].

However, in some (stiff) cases, the step size h_i taken may be large, for efficiency purposes, so that ξ_l in (2.16) are not all small. Then it is not clear that W^{-1} exists and is bounded. Indeed, Osborne [17] proposed to apply (a more expensive) QR decomposition to $\begin{pmatrix} W \\ D \end{pmatrix}$ to eliminate $\mathbf{z}_i$. It is desired then to show that this augmented $(k+m)\times k$ matrix has a full rank. Otherwise, this results in an unexplained redundancy in the method.

3.1. First order systems

Consider at first one equation

$$Dy = \lambda y. \tag{3.1}$$

We write the resulting equations (2.11), (2.18) as

$$\left[\begin{array}{c|c|c} -\lambda 1 & I-\xi A^{(1)} & 0 \\ \hline 1 & h\,\mathbf{b}^{T} & -1 \end{array}\right] \left[\begin{array}{c} y_i \\ \mathbf{z}_i \\ y_{i+1} \end{array}\right] = \left[\begin{array}{c} \mathbf{q}_i \\ \hline 0 \end{array}\right] \tag{3.2}$$

where $\xi := \xi_1 = \lambda h$, $h := h_i$ and $1 := (1,...,1)^{\mathrm{T}} \in \mathbf{R}^k$.

Now, if $re(\lambda) < 0$ then the A-stability of the method implies that $I - \xi A^{(1)}$ is nonsingular. But for BVPs we must also consider the case $re(\lambda) > 0$. If $|\xi| >> 1$ then $I - \xi A^{(1)}$ is again nonsingular, because $A^{(1)}$ is. But if $|\xi| = O(1)$ then a singularity may occur. For example, the midpoint scheme ($k=1$) yields

$$y_{i+1} = \frac{1+\xi/2}{1-\xi/2} y_i$$

and a singularity occurs at $\xi = 2$. The remedy is to write

$$y_i = \frac{1-\xi/2}{1+\xi/2} y_{i+1}$$

corresponding to integration in the stable direction $-t$.

Similarly for higher order schemes. If $re(\xi) > 0$ and $I - \xi A^{(1)}$ is singular then we multiply the last row in (3.2) by λ and add to each of the first k, obtaining the matrix

$$\left[\begin{array}{c|c|c} 0 & I-\xi A^{(1)}+\xi 1\mathbf{b}^{T} & -\lambda 1 \\ \hline 1 & h\,\mathbf{b}^{T} & -1 \end{array}\right] \tag{3.3}$$

Because of the symmetry of the points (see eqn (10) in [2]),

$$A^{(1)} + EA^{(1)}E = \mathbf{1}\mathbf{b}^T \tag{3.4}$$

where $E \in \mathbf{R}^{k\times k}$ is defined as

$$E := \begin{pmatrix} & & 1 \\ & \cdot & \\ 1 & & \end{pmatrix}. \tag{3.5}$$

Hence in (3.3) we have

$$I - \xi A^{(1)} + \xi \mathbf{1}\mathbf{b}^T = I + \xi EA^{(1)}E.$$

But $EA^{(1)}E$ is easily seen to have the same eigenvalues as $A^{(1)}$. A-stability then implies that $I + \xi EA^{(1)}E$ is nonsingular. Moreover, the first k rows of (3.3) clearly correspond to a reversal in the direction of integration.

This shows, in particular, that the matrix $\binom{W}{D}$ has full rank. The results can be easily extended to a first order system of equations (2.1) where L is a constant, upper triangular matrix with eigenvalues which all have real parts with the same sign. Generally for a system with constant coefficients, the $(k+1)n \times (k+2)n$ matrix corresponding to (3.2) has the form

$$\left[\begin{array}{c|c|c} -L & & \\ \vdots & I - hA^{(1)} \otimes L & 0 \\ -L & & \\ \hline I & hb_1 I \cdots hb_k I & -I \end{array}\right] \tag{3.6}$$

and, if $I - hA^{(1)} \otimes L$ is singular (which may occur only if the eigenvalues of L have positive real parts and at least one of them has magnitude comparable to h^{-1}), then multiplication of the last n rows by L and addition to each of the other blocks of n rows produces a nonsingular matrix $I + h(EA^{(1)}E) \otimes L$.

The next extension is to the general case where L is still constant. We assume that there exists an orthogonal transformation matrix S such that

$$S^{-1}LS = \begin{pmatrix} \Lambda_- & \\ & \Lambda_+ \end{pmatrix} \tag{3.7a}$$

where Λ_- is an upper triangular block with eigenvalues each satisfying either

$$re\,(\lambda) < 0, \quad |\,\lambda h\,| \gtrsim 1, \quad |\,im\,(\lambda)\,| \leq const\ |\,re\,(\lambda)\,| \tag{3.7b}$$

or

$$|\,\lambda h\,| \ll 1 \tag{3.7c}$$

and Λ_+ is an upper triangular block with eigenvalues satisfying

$$re\,(\lambda) > 0 \qquad |\,\lambda h\,| \gtrsim 1, \quad |\,im\,(\lambda)\,| \leq const\ |\,re\,(\lambda)\,|\,. \tag{3.7d}$$

Multiplying each row block in (3.6) by S^{-1} and performing the change of variables

$$\mathbf{w}_i := S^{-1}\mathbf{y}_i, \quad \mathbf{w}_{i+1} := S^{-1}\mathbf{y}_{i+1}, \quad \hat{\mathbf{z}}_{ij} := S^{-1}\mathbf{z}_{ij} \tag{3.8}$$

we obtain from (3.6) the matrix

$$\left[\begin{array}{c|c|c} -\Lambda & & \\ \vdots & I - hA^{(1)} \otimes \Lambda & 0 \\ -\Lambda & & \\ \hline I & hb_1 I \cdots hb_k I & -I \end{array}\right] \tag{3.9}$$

Now, for the block Λ_+ in each Λ we can again change the direction of integration by multiplying the last rows in (3.9) by Λ_+ and adding to the bottom portions of each of the first k blocks of n rows. This again results in a nonsingular matrix, showing that the original $\begin{pmatrix} W \\ D \end{pmatrix}$ matrix is of full rank.

Finally, we allow the matrix $L(x)$ to vary, but assume that there exists a *smooth* transformation $S(x)$ such that

$$S^{-1}(x)L(x)S(x) = \begin{bmatrix} \Lambda_-(x) & \\ & \Lambda_+(x) \end{bmatrix} \tag{3.10}$$

$$\|S(x)\|\|S^{-1}(x)\| \leq const. \qquad x_i \leq x \leq x_{i+1} \tag{3.11}$$

with Λ_-, Λ_+ blocks as in (3.7) and the partition is independent of x (cf. Kreiss, Nichols and Brown [14]). In place of (3.6) we now get

$$\left[\begin{array}{c|c|c} -L(x_{i1}) & & \\ \vdots & I - h\ diag\{L(x_{ij})\}A^{(1)} \otimes I & 0 \\ -L(x_{ik}) & & \\ \hline I & hb_1 I \cdots hb_k I & -I \end{array}\right]$$

so, we multiply the j-th block of n rows by $S^{-1}(x_{ij})$ and define

$$\mathbf{w}_i := S^{-1}(x_i)\mathbf{y}_i, \quad \mathbf{w}_{ij} := S^{-1}(x_{ij})\mathbf{z}_{ij}, 1 \leq j \leq k, \quad \mathbf{w}_{i+1} := S^{-1}(x_{i+1})\mathbf{y}_{i+1}. \tag{3.12}$$

This gives, in place of (3.9), the matrix

$$\left[\begin{array}{c|c|c} -\Lambda(x_{i1}) & & \\ \vdots & I - h\ diag\{\Lambda(x_{ij})\}A^{(1)} \otimes I & 0 \\ -\Lambda(x_{ik}) & & \\ \hline S(x_i) & hb_1 S(x_{i1}) \cdots hb_k S(x_{ik}) & -S(x_{i+1}) \end{array}\right].$$

Multiplying the last n rows by

$$\begin{bmatrix} 0 & \\ & \Lambda_+(x_{ij}) \end{bmatrix} S^{-1}(x_i)$$

and adding to each of the first k blocks of n rows, again yields the desired result for h_i sufficiently small (without requiring that $h_i \|L(x_i)\|$ be small !). This is because

$$S^{-1}(x_i)S(x_{il}) = I + O(h_i) \qquad 1 \leq l \leq k$$

and because rows corresponding to Λ_- and to Λ_+ are not mixed in the process. We have proved

Theorem 3.13.

If $L(x)$ can be decomposed as described above in (3.10), (3.11), (3.7), then for h_i small enough the matrix $\begin{pmatrix} W \\ D \end{pmatrix}$ has a full rank. Hence, the local parameters $\mathbf{z}_i$ can be eliminated and retrieved in terms of the nodal values $\mathbf{y}_i, \mathbf{y}_{i+1}$.

□

Remark.

The upwinding transformation above has been used for purposes of proof only. Note also that usually the matrix W is nonsingular. This matrix is guaranteed to be nonsingular if, under the conditions of Theorem 3.13, all eigenvalues satisfy

$$\text{either} \quad |\lambda h| \ll 1 \quad \text{or} \quad |\lambda h| \gg 1.$$

□

3.2. **Higher order ODE.**

Our results for the case $m > 1$ are slightly less complete than those summarized in Theorem 3.13 for $m = 1$. We state them below.

Theorem 3.14.

Consider the k-stage collocation equations (2.11), (2.18) for the ODE (2.2).

(a) If h_i is so small that

$$| \xi_l | \ll 1 =: \xi_{m+1} \qquad 1 \le l \le m$$

then W is nonsingular. More generallly, if there is an index $\tilde{l}$, $1 \le \tilde{l} \le m+1$, such that,

$$| \xi_l | \ll | \xi_{\tilde{l}} |, \qquad 1 \le l \le m+1,$$

then (for h sufficiently small, but $\xi_{\tilde{l}}$ not necessarily small) W is nonsingular.

(b) If

$$| \xi_l | \ll | \xi_m |, \qquad 1 \le l < m,$$

then (for h sufficiently small, but ξ_m not necessarily small) $\begin{pmatrix} W \\ D \end{pmatrix}$ is nonsingular.

Proof: To show part (a), consider first the case where $c_l(x)$ are constant along $[x_i, x_{i+1})$. Then we can write W as in (2.17) and the result follows from the fact that $A^{(m+1-\tilde{l})}$ is nonsingular, $1 \le \tilde{l} \le m+1$. Then, for the variable coefficient case, we use a perturbation argument for

$$c_l(x) = c_l(x_i)(1 + O(h_i)).$$

To see part (b), note that W is an $O(h_i)$ perturbation of the corresponding matrix for the first order case

$$I - h_i c_m A^{(1)}$$

and that the last row of D is just $h \mathbf{b}^T$. Hence the result follows from Theorem 3.13.

□

The results above cover, for instance, the ODE

$$\epsilon D^2 u = Du + u$$

for any combination of (sufficiently small values of) h_i and ϵ, but it does not cover some cases for the ODE

$$\epsilon D^2 u = u.$$

Nonetheless, practical experience indicates that it is rather rare that W be singular. We feel that it is hardly justifiable in a general purpose code to do more than decompose W itself and change h_i (say by halving the subinterval) in the unlikely event that W becomes singular.

4. SYMMETRIC REPRESENTATION.

One striking feature in the solution representation (2.7) is that it is not symmetric with respect to $\mathbf{y}_i$ and $\mathbf{y}_{i+1}$, even though the collocation scheme itself is. This results in the matrix Δ_i of (2.24) being the identity. The simplicity of Δ_i is usually not taken into account when solving the global linear system (i.e. (2.23) plus side conditions), so the idea of averaging the representation with respect to $\mathbf{y}_i$ and a corresponding one with respect to $\mathbf{y}_{i+1}$ naturally arises.

Thus we write, corresponding to (2.7)

$$u_\pi(x) = \sum_{j=1}^{m} \frac{(x-x_{i+1})^{j-1}}{(j-1)!} y_{i+1,j} + (-h_i)^m \sum_{j=1}^{k} \psi_{m,k+1-j}\left(\frac{x_{i+1}-x}{h_i}\right) z_{ij} \qquad (4.1)$$

$$x_i \le x \le x_{i+1}.$$

The collocation equations then yield

$$-\hat{V}\,\mathbf{y}_{i+1} + \hat{W}\,\mathbf{z}_i = \mathbf{q}_i \tag{4.2}$$

where $\hat{V}$ and $\hat{W}$ are $k \times m$ and $k \times k$ matrices respectively given by

$$\hat{V}_{rj} = \sum_{l=1}^{j} \frac{c_l(x_{ir})(-h_i \rho_{k+1-r})^{j-l}}{(j-l)!} \qquad 1 \le r \le k,\ 1 \le l \le m \tag{4.3}$$

$$\hat{W}_{rj} = \delta_{rj} - \sum_{l=1}^{m} c_l(x_{ir}) h_i^{m+1-l} D^{l-1} \psi_{m,k+1-j}(\rho_{k+1-r}) \qquad 1 \le r,\ j \le k. \tag{4.4}$$

Averaging (4.2) with (2.11) we obtain

$$\tilde{W}\,\mathbf{z}_i = \mathbf{q}_i + \frac{1}{2}(V\,\mathbf{y}_i + \hat{V}\,\mathbf{y}_{i+1}) \tag{4.5a}$$

$$\tilde{W} := \frac{1}{2}(W + \hat{W}). \tag{4.5b}$$

Assuming that $\tilde{W}$ is nonsingular, we can eliminate $\mathbf{z}_i$ and substitute in (2.18) (which can actually be symmetrized too, yielding no advantage). This gives the form (2.23) with

$$\Delta_i = I - \frac{1}{2} D \tilde{W}^{-1} \hat{V},\ \Gamma_i = C + \frac{1}{2} D \tilde{W}^{-1} V,\ \mathbf{r}_i = D\tilde{W}^{-1}\mathbf{q}_i \tag{4.6}$$

The symmetry introduced into $\tilde{W}$ by the above process can be seen by writing, similarly to (2.17),

$$\tilde{W} = I - \sum_{l=1}^{m} \xi_l \tilde{A}^{(m+1-l)} \tag{4.7}$$

with

$$\tilde{A}_{rj}^{(m-l)} := \frac{1}{2}[D^l \psi_{mj}(\rho_r) + (-1)^{m-l} D^l \psi_{m,k+1-j}(\rho_{k+1-r})], \qquad 1 \le r, j \le k,\ \ 0 \le l \le m. \tag{4.8}$$

It is apparent from (4.8) that

$$\tilde{A}^{(m-l)}_{k+1-r,k+1-j} = (-1)^{m-l}\tilde{A}^{(m-l)}_{rj} \tag{4.9a}$$

or, in matrix notation (cf. (3.5), (3.4))

$$E\tilde{A}^{(m-l)}E = (-1)^{m-l}\tilde{A}^{(m-l)}. \tag{4.9b}$$

However, for $m>1$ we cannot recommend the symmetrization idea: The simplicity gained in $\tilde{W}$ is not seen to compensate for the fact that we now have two matrices, V and $\hat{V}$, to generate, solve for and possibly store, in place of the one matrix V previously encountered.

The case is different for $m=1$. Here V and $\hat{V}$ coincide (as do C and I). Moreover, we now prove

Theorem 4.10.

For collocation at k Gaussian points with $m = 1$, not only

$$E\tilde{A}^{(1)}E = -\tilde{A}^{(1)} \tag{4.10a}$$

but also

$$\tilde{A}^{(1)}_{jj} = 0 \qquad 1\leq j\leq k. \tag{4.10b}$$

Furthermore, all the eigenvalues of $\tilde{A}^{(1)}$ are on the imaginary axis.

Proof: Note that (4.10a) follows directly from (4.9b) and that, in general, for any $m\geq 1$

$$D^{m-1}\psi_{mj}(\rho_r) + D^{m-1}\psi_{m,k+1-j}(\rho_{k+1-r}) = D^{m-1}\psi_{mj}(1) = b_j, \qquad 1\leq r,j\leq k.$$

Hence we can write

$$\tilde{A}^{(1)} = A^{(1)} - \frac{1}{2}\mathbf{1}\mathbf{b}^T. \tag{4.11}$$

Now, since collocation at Gaussian points yields a symmetric, algebraically stable scheme, we have

$$A_{jj}^{(1)} = \frac{1}{2}b_j \qquad 1 \leq j \leq k$$

(see [2, eqn. (6.1)]) and (4.10b) follows.

Further, let X be a $k \times k$ matrix defined as in [2, eqn (12)] (following Hairer and Wanner [12]). Thus

$$X = W^{(1)^{-1}} A^{(1)} W^{(1)}$$

for a certain matrix $W^{(1)}$ and, since the scheme is symmetric and algebraically stable,

$$X + X^T = \mathbf{e}_1 \mathbf{e}_1^T, \quad \mathbf{e}_1 := (1,0,\ldots,0)^T = W^{(1)^T}\mathbf{b}.$$

Defining similarly

$$\tilde{X} := W^{(1)^{-1}} \tilde{A}^{(1)} W^{(1)}$$

we obtain from (4.11)

$$\tilde{X} = X - \frac{1}{2} W^{(1)^T} \mathbf{b}\,\mathbf{b}^T W^{(1)} = X - \frac{1}{2}\mathbf{e}_1 \mathbf{e}_1^T.$$

Hence

$$\tilde{X} + \tilde{X}^T = 0.$$

The fact that $\tilde{A}^{(1)}$ is similar to a skew-symmetric matrix establishes our claim regarding its eigenvalues.

□

The above proof of Theorem 4.10 obviously generalizes for any symmetric, algebraically stable Runge-Kutta scheme, cf. [2].

Thus we see for the first order ODE (3.1) that the matrix $\tilde{W} = I - \xi\tilde{A}^{(1)}$ (unlike W) has a bounded inverse for any $|\xi| = O(1)$, provided that either $|im(\xi)| \leq const\ |re(\xi)|$ or $|\xi| \ll 1$. If λ is imaginary then $|\xi|$ has to be taken small for approximation purposes anyway.

On the other hand, when k is odd the matrix $\tilde{A}^{(1)}$ is clearly singular. For instance, the midpoint scheme ($k=1$) yields $\tilde{A}^{(1)} = 0$, and (4.6) gives ($\tilde{W}=1$)

$$(1-\xi/2)y_{i+1} = (1+\xi/2)y_i\ .$$

This equation has to be rescaled when $|\xi| \gg 1$.

These results extend to the general system considered in Theorem 3.13. In particular, if $L(x)$ can be decomposed as described in (3.7), (3.10), (3.11) then the corresponding matrix $\tilde{W}$ has a bounded inverse. The resulting difference equations (2.23) may have to be rescaled (row-wise) in case that $|\xi| \gg 1$ and k is odd.

We conclude that the symmetric implementation considered here is worthwhile for first order ODEs, but not for higher order ones.

5. MIXED ORDER SYSTEMS

The class of problems treated by COLSYS has the following general form (see [5]): A system of d (generally) nonlinear ODEs of orders m_n, $1 \leq n \leq d$,

$$D^{m_n}u_n(x) = F_n(x;u_1,Du_1,\ldots,D^{m_1-1}u_1,u_2,\ldots,D^{m_d-1}u_d) \equiv F_n(x;\mathbf{v}(\mathbf{u}))$$
$$a < x < b$$

is subject to nonlinear side conditions

$$g_j(\varsigma_i; \mathbf{v}(\mathbf{u})) = 0, \quad \varsigma_1 \le \varsigma_2 \le \cdots \le \varsigma_{m*}, \quad \varsigma_j \in [a,b], \; j = 1,\ldots,m* \tag{5.2}$$

where $m* := \sum_{n=1}^{d} m_n$. For the nonlinearities, some techniques based on Newton's method in quasilinearization form are applied, see Ascher, Christiansen and Russell [3, 4]. Thus, for solution representation it suffices to consider the linear ODEs

$$D^{m_n} u_n(x) = \sum_{\mu=1}^{m*} c_{n\mu}(x) v_\mu(x) + q_n(x) \qquad a < x < b \tag{5.3}$$

$$c_{n\mu} \equiv \frac{\partial F_n}{\partial v_\mu}. \tag{5.4}$$

With a mesh π of (2.4) so constructed that $\varsigma_j \in \pi$, $1 \le j \le m*$, we seek approximate solutions

$$\mathbf{u}_\pi = (u_{1\pi},\ldots,u_{d\pi})^T, \quad u_{n\pi} \in \mathbf{P}_{k+m_n,\pi} \cap C^{m_n-1}[a,b] \qquad 1 \le n \le d \tag{5.5}$$

by collocation at Gaussian points. In view of the considerations in the two previous sections (and in [6,2,1]), we have implemented an extension of the solution representation of §2 which is now described.

We write as in (2.7), (2.8)

$$u_{n\pi}(x) = \sum_{j=1}^{m_n} \frac{(x-x_i)^{j-1}}{(j-1)!} y_{nij} + h_i^{m_n} \sum_{j=1}^{k} \psi_{m_n,j}\left(\frac{x-x_i}{h_i}\right) z_{nij} \qquad x_i \le x \le x_{i+1} \tag{5.6}$$

where for each n, $1 \le n \le d$,

$$y_{nij} = D^{j-1} u_{n\pi}(x_i) \qquad 1 \le j \le m_n \tag{5.7a}$$

$$z_{nij} = D^{m_n} u_{n\pi}(x_{ij}) \qquad 1 \le j \le k \tag{5.7b}$$

and let $\mathbf{y}_i \in \mathbf{R}^{m*}$ and $\mathbf{z}_i \in \mathbf{R}^{kd}$ be defined by

$$\mathbf{y}_i := (y_{1i1},\ldots,y_{1im_1},y_{2i1},\ldots,y_{dim_d})^T \equiv \mathbf{v}_\pi(x_i) \tag{5.8a}$$

$$\mathbf{z}_i := (z_{1i1},\ldots,z_{1ik},z_{2i1},\ldots,z_{2ik},\ldots,z_{di1},\ldots,z_{dik})^T. \tag{5.8b}$$

The side conditions (5.2) can now be written in a straightforward manner as

$$g_j(\zeta_j;\mathbf{y}_{i_j}) = 0 \qquad 1\le j\le m^* \tag{5.9}$$

where for each j, i_j is the index such that $\zeta_j = x_{i_j}$. (But, like the ODEs, the side conditions are linearized as well.)

Next we write down the collocation and the continuity equations on the i-th subinterval of the mesh π. We use the same notation as in (2.11), (2.18), but now

$$C = diag\,\{C^{(n)}\} \in \mathbf{R}^{m^*\times m^*} \tag{5.10}$$

where $C^{(n)}$, $1\le n\le d$, is an $m_n\times m_n$ upper triangular matrix with entries given in (2.19). Similarly, $D\in\mathbf{R}^{m^*\times kd}$ is given by

$$D = \begin{bmatrix} D^{(1)} & & & \\ & D^{(2)} & & \\ & & \ddots & \\ & & & D^{(d)} \end{bmatrix} \tag{5.11}$$

where $D^{(n)}\in\mathbf{R}^{m_n\times k}$ is given by (2.20) with m replaced by m_n. Also $W\in\mathbf{R}^{kd\times kd}$ is given by

$$W = (W^{(n,\nu)})_{n,\nu=1}^{d} \tag{5.12a}$$

where each $W^{(n,\nu)}$ is a $k\times k$ matrix given by

$$W_{rj}^{(n,\nu)} = \delta_{n\nu}\delta_{rj} - \sum_{l=1}^{m_\nu} c_{n\mu}(x_{ir})h_i^{m_\nu+1-l}D^{l-1}\psi_{m_\nu j}(\rho_r) \qquad 1\le r,j\le k \tag{5.12b}$$

with

$$\mu \equiv \sum_{\eta=1}^{\nu-1} m_\eta + l. \tag{5.12c}$$

The advantage of using functions ψ_{mj} which satisfy (2.10) is now apparent, because the only form in which these functions appear in W and D is

$$h_i^{\xi}\, \psi_{\xi j}(t) := h_i^{\xi}\, D^{m-\xi}\psi_{mj}(t) \qquad 0\le\xi\le m \qquad (\xi := m_\nu+1-l) \tag{5.13a}$$

where m is a predetermined constant (e.g. $m=4$) satisfying

$$m \ge \max\{m_n ; 1\le n \le d\}. \tag{5.13b}$$

Thus, the mesh independent constants ρ_r, $\psi_{\xi j}(\rho_r)$ and $\psi_{\xi j}(1)$ are stored in the code for $1\le k\le 7$, $m := min(k,4)$, $1\le\xi\le m$, $1\le j\le k$, $1\le r\le k$. Using these, the assembly of W and D is straightforward.

Finally, for V of (2.11) we have a $kd \times m*$ matrix given by

$$V = (V^{(n,\nu)})_{n,\nu=1}^{d} \tag{5.14a}$$

where $V^{(n,\nu)} \in \mathbf{R}^{k,m_\nu}$ is given by

$$V_{rj}^{(n,\nu)} = \sum_{l=1}^{j} \frac{c_{n\mu}(x_{ir})(h_i\rho_r)^{j-l}}{(j-l)!} \qquad 1\le r\le k,\ 1\le j\le m_\nu \tag{5.14b}$$

with μ related to l by (5.12c), and

$$\mathbf{q}_i := (q_1(x_{i1}),\ldots,q_1(x_{ik}),\ldots,q_d(x_{i1}),\ldots,q_d(x_{ik}))^T \tag{5.15}$$

Remark.

A natural extension of this implementation can be made to a special case of *differential-algebraic ODEs*, where the ODEs in (5.1) depend also on $\hat{m}$ unknown functions $\mathbf{w}(x) \equiv (w_1(x),\ldots,w_{\hat{m}}(x))^T$ and are supplemented by $\hat{m}$ algebraic equations, tying $\mathbf{v}(x)$ and $\mathbf{w}(x)$, $a\le x\le b$. These algebraic equations can be considered as

ODEs of order 0 and, correspondingly, the collocation approximation is in the space

$$\mathbf{w}_\pi \in \mathbf{P}_{k,\pi}$$

(which admits discontinuities at mesh points). The obtained representation (5.6) for $w_{n\,\pi}(x)$ is just a Lagrange interpolant form.

□

In our implementation we eliminate for each i, $1 \le i \le N$, the local unknowns $\mathbf{z}_i$ from (2.18), and form (2.23), (2.24). If W is deemed too ill-conditioned, the interval $[x_i, x_{i+1}]$ is halved. The obtained equations for $\mathbf{y}_i$, $1 \le i \le N+1$ are merged with those resulting from the side conditions to form a large, sparse system of order $(N+1)m*$. The resulting matrix is almost block diagonal (see [5,§4] and references therein), with the i-th block having $m*+l_i$ rows (and l_i is the number of side conditions already encountered), $2m*$ columns and an offset of $m*$ columns with respect to its neighboring blocks.

6. IMPLEMENTATION AND NUMERICAL RESULTS

When considering a general purpose collocation code, the following aspects arise:

1. Basis representation
2. Linear system solution
3. Error estimation
4. Mesh selection
5. Nonlinear problem solution.

Our aim when replacing the basis representation in COLSYS has been to make as little change as possible in the modules dealing with the other aspects, thus allowing for a

meaningful assessment of the performed replacement. Nonetheless, some considerations are needed which may not seem at a first glance to be directly related to the basis representation. Thus, while error estimation and mesh selection are essentially not affected by changing the basis, the linear and nonlinear equation solving procedures are, as described below.

Using the monomial basis representation, the linear algebra part of the solution process appears in two places: in the local parameter elimination and in the solution of the resulting, global linear system for $\{\mathbf{y}_{ij}\}$. For the latter, the code SOLVEBLOK [10] is used in COLSYS, see [4, 5]. When using B-splines, the block sizes of the global linear system are larger and no additional parameter elimination is done. Thus, a more extensive use of SOLVEBLOK is made in COLSYS than in its replacement. Consequently, if SOLVEBLOK can be improved then the comparison between the old and the new bases is affected.

And SOLVEBLOK can indeed be improved. As written by de Boor and Weiss [10], it is row-oriented in its handling of matrices. Since it is written in FORTRAN, Multidimensional arrays are stored columnwise, so a column orientation is preferable, i.e. the innermost loops should attempt to access consecutive memory locations as much as possible, see Moler [16]. For the reasons mentioned above, we have not replaced SOLVEBLOK by another linear system solver altogether, but we have rewritten its loops and reduced the number of array shifts. The resulting savings in computing time are machine and compiler dependent, and can be significant when the block sizes which SOLVEBLOK has to handle are large, as in example 5 below. Hay and Gladwell [13] have considered these questions, especially for vector machines.

The dense linear system solver used for the local parameter elimination has also been efficiently implemented, following the LINPACK routines [11]. An additional advantage of this is that it allows an easy adaptation of the code to vector machines or the use of a sparse system solver when appropriate.

The nonlinear system solution algorithm also has to be changed when changing the basis representation, because the unknowns are now differently scaled. This has an effect on the control of the damped Newton iteration used in COLSYS, and is felt only in highly (nonlinearly) sensitive situations. Our first instinct has been to use only the variables $\{\mathbf{y}_i\}$ for controlling the iteration. This, however, proved to be insufficient in some applications, and the variables $\{\mathbf{z}_i\}$ are now used as well. Consulting (5.6), (5.8), we use a relative scaling of $\dfrac{h_i^j}{j!\,(\,|\,y_{n,i+1,1}\,|\,+1)}$ for a variable representing the jth derivative of the nth solution component on the ith subinterval. Note that this scaling is independent of a stretching transformation to the independent variable. The rest of the damped Newton strategy in COLSYS has not been changed.

In the course of implementation of the monomial basis (we refer to the resulting program as COLNEW in this paper) and the inevitable comparison to COLSYS which followed, we have become once again painfully aware of two known facts in mathematical software: That a simple comparison of two proposed schemes can be completely different than a comparison of their implementation in a general purpose setting, and that a comparison of two general purpose codes, even fairly similar ones, can be complicated, depending on the machine, the precision and the compiler used, as well as on the examples chosen.

Our first goal during implementation has been not to compromise on any feature of COLSYS, i.e. that COLNEW be comparable or better than COLSYS for any BVP. We believe we have achieved this; however, the amount of improvement in COLNEW over COLSYS in our experiments is varied. We have run our experiments on an IBM 3081 using, unless otherwise noted, double precision (56 bit mantissa) and occasionally also single precision (24 bit mantissa).

Example 1. From [6] we know that COLNEW should be faster than COLSYS for a linear ODE of a high order. Indeed, consider example 1 in [6] and [5]. For this 4th order ODE, with tolerances of 10^{-6} and 10^{-10}, the run time of COLSYS is almost halved by COLNEW. The actual maximum error resulting from the runs with the stiffer tolerance is also more accurate when using COLNEW by about two digits (cf. [6]).

□

Note that for linear problems, the monomial representation also offers an implementation which is cheaper in terms of storage, because the W matrices do not have to be saved. For nonlinear problems, however, we do save these matrices in order to be able to restore the full solution, and the total storage requirements are about 10% higher than with the B-spline implementation.

When using the monomial basis to solve a BVP, solution values at mesh points are readily available. For solution values elsewhere, the values of $\mathbf{z}_i$ are used as well, in a straightforward manner. However, the latter solution evaluations are not cheaper than when using B-splines. They become a more important component of the total cost in nonlinear problems and when more error estimation and mesh selection is done in COLSYS. Also, since the same number of Jacobian evaluations is required regardless of the basis (given that the nonlinear iteration control is as effective), the performance

difference between COLSYS and COLNEW tends to be less significant the more expensive these Jacobian evaluations are.

Example 2. For example 2 in [5], which has two 2nd order ODEs featuring boundary and interior layers, the two codes perform similarly, COLSYS being slightly faster. But COLSYS is more sensitive than COLNEW to a good compiler optimization. When using the FTN compiler with a lower level of optimization, COLNEW is about 20% faster. This is important when no high level optimizing compiler is available in a particular computing environment.

Another singularly perturbed BVP which involves a 4th order and a 2nd order ODE is considered in [3, example 3]. Here COLNEW is faster than COLSYS by about 25% using the optimized object codes.

□

Example 3. The third example in [5] consists of three continuation steps for a coupled pair of 2nd and 3rd order ODEs. This is also a sensitive nonlinear BVP. Here COLNEW performs marginally better when using the optimized codes and by about 20% better with the low-level compiler optimization.

This BVP contains a solution component G which tends to 1 as the independent variable $x \rightarrow \infty$. If we now make the change of variable $G := G - 1$ then the runs with COLNEW are not seriously affected, but the COLSYS run times get significantly worse, because the scaling in the nonlinear iteration control is affected by the tail noise in G. This is unusual, though, and may be considered as a bug in COLSYS which is eliminated in COLNEW.

□

The other, probably more important, theoretical improvement which the monomial basis has to offer is the improved condition number of the linear systems arising, in case of higher order ODEs. See [6] for an extended discussion on this point. This effect becomes apparent in our setting here only when the error tolerances are close to the machine precision, which normally is more practically relevant when a short word length is used.

Example 4. Consider example 7.2 in Ascher and Russell [7]. This is a nonlinear 2nd order ODE on a semi-infinite interval. Using double precision, the results in table 1 of [7] for $L = 20$ are easily recovered with both codes, COLNEW being about 10% faster. But when switching to single precision, COLNEW is still able to produce the desired results, while the nonlinear iteration in COLSYS on a mesh with 32 elements fails, because the nonlinear iteration control cannot handle a case where roundoff error dominates.

A similar effect occurs when attempting to solve example 3 in single precision. With COLNEW, the first two continuation steps are successfully completed (but the last step fails), while with COLSYS only the first continuation step is completed and the second one fails. The onset of roundoff error domination occurs sooner in COLSYS than in COLNEW !

□

The advantage of COLNEW demonstrated in the last example above may be important in certain computing environments. If computations need to be performed with short word length then COLNEW should be clearly preferred, being more robust.

The comments made earlier regarding the linear system solver are not of major importance in the above examples. But the efficient implementation of the linear system solver becomes significant in the following.

Example 5. This problem is described in Stephenson et al. [18], Mejia and Stephenson [15]. Our numerical solution procedure for it using COLSYS has not appeared anywhere before, so we briefly describe it here. The model describes mass and energy balance of the renal counterflow system. With F_{iv} - the axial volume flow in the i-th tube, J_{iv} - the outward transmural volume flux, $F_{iv}\,C_{ik}$ - the axial flow of the k^{th} solute in the i^{th} tube and J_{ik} - the outward transmural flux per unit length of the k^{th} solute from the i^{th} tube, there are 18 ODEs with nonseparated boundary conditions for F_{iv}, C_{ik}, $1 \le i \le 6$, $1 \le k \le 2$, describing the steady state case, see [18].

Now, it can be verified that this BVP can be simplified to have only 14 ODEs with separated BC, as follows:

$$C_{12}' = 200(C_{12})^2[C_{41} + C_{42} - 21C_{12}]$$

$$C_1 = 20C_{12}, \quad F_{1v} = \frac{0.05}{C_{12}}, \quad F_{2v} = -\frac{0.05}{C_{22}}$$

$$C_{21}' = 20C_{22}J_{21}, \quad C_{22}' = 0$$

$$C_{31}' = \frac{10}{K_1}(C_{31})^2[C_{41} + C_{42} - C_{31} - C_{32}]$$

$$C_{32}' = \frac{C_{31}}{K_1}[J_{3v}\,C_{32} - J_{32}]$$

$$K_1' = 0, \quad K_2' = 0, \quad F_{4v}' = -J_{4v}$$

$$K_1 = \frac{C_{31}(0)}{2C_{32}(0)}, \quad F_{3v} = \frac{K_1}{C_{31}}, \quad F_{5v} = 5, \quad F_{6v} = \frac{0.05}{C_{62}}$$

$$C_{41}' = \frac{1}{F_{4v}}[J_{4v}\,C_{41} - J_{41}]$$

$$C_{42}' = \frac{1}{F_{4v}}[J_{4v}\, C_{42} - J_{42}]$$

$$C_{51}' = -200(C_{51} - C_{41}), \qquad C_{52}' = -200(C_{52} - C_{42})$$

$$C_{61}' = 20C_{62}[J_{6v}\, C_{61} - J_{61}]$$

$$C_{62}' = 20(C_{62})^2 J_{6v}$$

$$C_{12}(0) = 0.05, \quad C_{51}(0) = 1, \quad C_{52}(0) = 0.05, \quad F_{4v}(1) = -5$$

$$C_{31}(0) - 20K_1(0)C_{32}(0) = 0, \quad C_{22}(0) = C_{62}(0), \quad C_{61}(0) = C_{21}(0)$$

$$C_{12}(1) = C_{22}(1), \; C_{21}(1) = 20C_{12}(1), \; C_{41}(1) = C_{51}(1), \; C_{42}(1) = C_{52}(1)$$

$$C_{61}(1) - 20K_1(1)C_{62}(1) = 0, \; C_{31}(0) = K_2(0), \; C_{61}(1) = K_2(1).$$

Further, the functions J_{i1}, J_{i2} and J_{iv}, $1 \leq i \leq 6$, are now specified. Transmural volume fluxes are defined as follows:

$$J_{iv} = h_{iv} \sum_{k=1}^{2} (C_{4k} - C_{ik}), \qquad i = 1,2,3,5$$

$$= (1. - C_{i1}) + (0.05 - C_{i2}), \qquad i = 6,$$

where $h_{1v} = h_{3v} = 10$ and $h_{iv} = 0$ for i = 2,5.

$$J_{4v} = - \sum_{i \neq 4,6} J_{iv}$$

Transmural solute fluxes are:

$$J_{i1} = 0, \qquad i = 1,3$$

$$= 0.75\, C_{i1}/(1. + C_{i1}), \qquad i = 6$$

$$= 1000(C_{i1} - C_{41}), \qquad i = 4.$$

$$J_{21}(x) = \begin{cases} 1.8, & 0. \le x \le 0.4 \\ 1.8+[-18.+100(C_{21}(x)-C_{41}(x))]\cdot(x-0.4), & 0.4 < x < 0.5 \\ 10[C_{21}(x)-C_{41}(x)], & 0.5 \le x \le 1. \end{cases}$$

$$J_{i2} = 0, \qquad i = 1,2,6$$

$$= 1000(C_{i2} - C_{42}), \qquad i = 5.$$

$$J_{32}(x) = \begin{cases} 0., & 0. \le x \le 0.4 \\ 0.1[C_{32}(x)-C_{42}(x)]\cdot(x-0.4), & 0.4 < x < 0.5 \\ 0.01[C_{32}(x)-C_{42}(x)], & 0.5 \le x \le 1. \end{cases}$$

$$J_{4k}(x) = -\sum_{i \neq 4,6} J_{ik}(x), \qquad 0. \le x \le 1, \qquad 1 \le k \le 2.$$

This completes the specification of the problem.

For the solution of this BVP, beginning with the simple initial guess that all solution components are identically equal to 1, we apply a chain of simple continuation steps. Thus we use COLSYS or COLNEW to solve the BVP defined above with $\lambda = h_{1v} = h_{3v}$ taking the values 0, 0.8, 1.6, 2.4, 4, 6, 8, 10, using the solution obtained with each λ as the initial guess for the BVP with the next λ in the sequence.

Since this is a first order system, we expect the difference between the codes, if any, to be mainly due to the linear system solver. The resulting run using COLSYS to complete the entire sequence of continuation steps, takes about 93 cpu seconds, while COLNEW needs only about 58 seconds. When linking COLSYS to the improved SOLVEBLOK as described above, the COLSYS run time is reduced to 70 seconds.

A similar effect can be expected when running other examples with many ODEs, because these give rise to large blocks in the Jacobian of the nonlinear system resulting from collocation discretization. In examples 1-4 above these blocks are not very large and

other considerations dominate the codes performance.

□

In conclusion, the implementation of the monomial basis, as described here, provides a cleaner solution representation which has generally resulted in a somewhat more robust code than COLSYS.

References

1. U. Ascher, "Collocation for two-point boundary value problems revisited", to appear SIAM J. Numer. Anal.

2. U. Ascher and G. Bader, "Stability of collocation at Gaussian points", to appear SIAM J. Numer. Anal.

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

4. _____ ,"COLSYS - A collocation code for boundary value problems", in Springer Lecture Notes in Computer Science 76, B. Childs et al (Eds.), New York 1979.

5. _____ ,"Collocation software for boundary value ODEs", ACM Trans. Math Software 7 (1981), 209-222.

6. U. Ascher, S. Pruess and R.D. Russell, "On spline basis selection for solving differential equations", SIAM J. Numer. Anal. 20 (1983), 121-142.

7. U. Ascher and R.D. Russell, "Reformulation of boundary value problems into "standard form"", SIAM Review 23 (1981), 238-254.

8. C.de Boor, A Practical Guide to Splines, Applied Math. Sciences vol. 27, Springer, New York, 1978.

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

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

11. J.J. Dongarra, J.R. Bunch, C.B. Moler and G.W. Stewart, LINPACK User's Guide, SIAM Publications, Philadelphia, 1979.

12. E. Hairer and G. Wanner, "Characterization of non-linearly stable implicit Runge-Kutta methods", in Springer Lecture Notes in Math. 968 (1982), 207-219.

13. R.I. Hay and I. Gladwell, "Solving almost block diagonal linear equations on the CDC Cyber 205", Numerical Analysis Report 98, Dept. Mathematics, Univ. of Manchester, 1985.

14. H. Kreiss, N. Nichols and D. Brown, "Numerical methods for stiff two-point boundary value problems", to appear SIAM J. Numer. Anal.

15. R. Mejia and J.L. Stephenson, "Numerical solution of multinephron kidney equations", J. Comp. Phys. 32 (1979), 235-246.

16. C. Moler, Algorithm 423, Linear equation solver, CACM 15 (1972), 274-#.

17. M.R. Osborne, "Collocation, difference equations, and stitched function representations", in Topics in Numerical Analysis II, 121-132, J. Miller (Ed.), AP 1975.

18. J.L. Stephenson, R.P. Tewarson and R. Mejia, "Quantitative analysis of mass and energy balance in non-ideal models of the renal counterflow system", Proc. Nat. Acad. Sci. USA 71 (1974), 1618-1622.