1. Introduction

The methods which have been implemented in general purpose codes to solve systems of boundary-value ODE's can be divided into two groups. The first is that of initial-value methods (shooting), in which most of the current working codes fall (e.g., [35], [116], [238]). The second is that of global methods--finite differences and finite elements. Finite differences with deferred corrections have been implemented by Lentini and Pereyra [167,170]. As far as we know, ours is the only general-purpose code of this type based on finite elements.

Finite elements have been considered to be generally slower than other methods for ODE's and therefore have been ruled out of competition despite their attractive theory. We believe that finite-element methods, other than collocation, are indeed too slow; however, collocation at Gaussian points, when implemented efficiently, is competitive with finite differences [226], [227]. At the same time, collocation's sound theoretical footing has resulted in the derivation of effective algorithms for error estimation and mesh selection (see Section 2). These aspects and the stable performance of the procedure make it very attractive.

Consider a mixed order system of \( d \) (nonlinear) differential equations of orders \( m_1 \leq m_2 \leq \ldots \leq m_d \),

\[
\begin{equation}
(1.1) \quad u_n(x) = F_n(x; z(u)) \quad a < x < b, \ n = 1, \ldots, d,
\end{equation}
\]

where \( y = (u_1, \ldots, u_d) \) is the sought solution vector and
\[ z(u) = (u_1, u_1', \ldots, u_1^{(m_1-1)}, u_2, \ldots, u_2^{(m_2-1)}, \ldots, u_d, \ldots, u_d^{(m_d-1)} ) \] is the vector of unknowns that would result from converting (1.1) to a first order system. These differential equations are subject to \( m* = \sum_{n=1}^{d} m_n \) (nonlinear) multi-point separated boundary conditions

\[ g_j(\xi_j; z(u)) = 0 \quad j = 1, \ldots, m* \]

where \( \xi_j \) is the location of the \( j \)-th boundary or side condition,
\[ a \leq \xi_1 \leq \xi_2 \leq \ldots \leq \xi_{m*} \leq b. \]

The collocation method implemented in COLSYS solves (1.1)-(1.2) directly, without converting it to a first-order system. This is in contrast to other codes. A conversion to a first-order system would increase the size of the problem and change the algebraic structure of its discretization.

Let \( \pi \) be a mesh on \([a,b]\),

\[ \pi : a = x_1 < x_2 < \ldots < x_N < x_{N+1} = b \]

and \( P_{\ell,\pi} = \{ v \mid v \} \) is a polynomial of degree \( \ell \), \( i = 1, \ldots, N \). The collocation approximation is a vector \( v = (v_1, \ldots, v_d) \) such that

\[ v_n \in P_{k+m_n,\pi} \cap C^{(m_n-1)} [a,b], \quad n = 1, \ldots, d, \text{ with } k \geq m_d \text{ being the number of collocation points per subinterval } (x_i, x_{i+1}). \] If \( \{ p_j \}_{j=1}^{k} \) are the Gauss-Legendre points on \([-1,1]\), then the collocation points are defined by

\[ x_{ij} := \frac{x_i + x_{i+1}}{2} + \frac{1}{2} h_i p_j =: x_{i+1/2} + \frac{1}{2} h_i p_j \quad i = 1, \ldots, N, \quad j = 1, \ldots, k. \]

The collocation solution \( v \) is determined by requiring that