Superconvergent Interpolants for Collocation Methods applied to mixed order BVODEs*

W.H. Enright and R. Sivasothinathan
Department of Computer Science,
University of Toronto,
Toronto, Canada M5S 3G4

Abstract

Continuous approximations to boundary value problems in ordinary differential equations (BVODEs), constructed using collocation at Gauss points, are more accurate at the mesh points than at off-mesh points. From these approximations, it may be possible to construct improved continuous approximations by extending the high accuracy that is available at the mesh points to off-mesh points. One possibility is the **Bootstrap** approach, which improves the accuracy of the approximate solution at the off-mesh points in a sequence of steps until the accuracy at the mesh points and off-mesh points is consistent. A Bootstrap approach for systems of mixed order BVODEs will be developed to improve approximate solutions produced by **COLNEW**, a Gauss collocation based software package. An implementation of this approach will be discussed and numerical results presented which confirm that the improved approximations satisfy the predicted error bounds and are relatively inexpensive to construct.

1 Introduction

Many real world phenomena such as fluid flows, chemical reactions, epidemics, etc, can be modeled (see for example [4, pp.7-27]) with mixed-order, boundary value ordinary differential equations (BVODEs) of the form

$$u_j^{(m_j)}(x) = f_j(x, z(u(x))), \quad \text{for } j = 1, \ldots, ncomp, \tag{1}$$

with separated multi-point boundary conditions

$$g_l(\xi_l, z(u(\xi_l))) = 0, \quad \text{for } l = 1, \ldots, m^*, \tag{2}$$

*This work was supported by the Natural Science and Engineering Research Council of Canada and by Communications and Information Technology Ontario.

where

$u(x) = [u_1(x), \ldots, u_{ncomp}(x)]$,
$z(u(x)) = [u_1^{(0)}(x), \ldots, u_1^{(m_1-1)}(x), \ldots, u_{ncomp}^{(0)}(x), \ldots, u_{ncomp}^{(m_{ncomp}-1)}(x)]$,
$f_j : \Re \times \Re^{m_1} \times \cdots \times \Re^{m_{ncomp}} \rightarrow \Re$,
$g_l : \Re \times \Re^{m_1} \times \cdots \times \Re^{m_{ncomp}} \rightarrow \Re$,
$m^* = \sum_{r=1}^{ncomp} m_r$, and $\xi_1 \leq \xi_2 \leq \cdots \leq \xi_{m^*}$ are the boundary points.

One of the most widely used software packages for approximating the solution of (1)(2) is the COLNEW/COLSYS [1][2][3] package. This package partitions the interval of integration into a mesh $x_0 < x_1 < x_2 < \ldots < x_N$ and, using collocation at the images of k Gauss points [6][4, pp.218-266] in each mesh subinterval, generates a piecewise polynomial approximation to the exact solution by requiring that the approximate solution satisfy the boundary conditions at the boundary points and the differential equation at the collocation points. At the mesh points, derivatives $0, \ldots, m_j - 1$ of the j^{th} component of this approximation have an $O(h^{2k})$ global error bound, where h is the length of the largest mesh subinterval. However, at off-mesh points, the r^{th} derivative of the j^{th} component of this approximate solution has a weaker error bound of $O(h^{k+m_j-r})$. Both these error bounds are based on the assumption that the number of Gauss points, k, is at least as large as the largest derivative order, $mmax$, present in the differential equation. In the remainder of this paper, we will assume that k is always chosen to satisfy this requirement.

Since this approximate solution is more accurate at the mesh points than at off-mesh points, we are motivated to look for inexpensive ways to improve this approximation so that the high accuracy that is present at the mesh points is also available at the off-mesh points. In the literature [8][13], such improved approximations to the solution are called **superconvergent interpolants/approximations**. We will extend and justify a Bootstrap approach for constructing superconvergent interpolants.

Besides approximating the exact solution within an error bound of $O(h^{2k})$, the following properties are desirable in a superconvergent interpolant:

- The superconvergent interpolant approximating the j^{th} solution component should have C^{m_j-1} continuity. Ideally this is accomplished by interpolating derivatives $0, \ldots, m_j - 1$ of the j^{th} component of the underlying discrete approximation at the mesh points.
- The r^{th} derivative of the superconvergent interpolant should have an error bound of at least $O(h^{2k-r})$ for each component.

In the next section, we will present an overview of COLNEW and Hermite-Birkhoff interpolation. Within this context we will provide a review of recent work on constructing superconvergent interpolants. We will then extend the Bootstrap approach for constructing superconvergent interpolants for systems of mixed order BVODEs and illustrate how it can be applied in conjunction with the approximations produced by COLNEW. Finally, we will discuss an implementation of the Bootstrap approach, and present some numerical results.

Before we proceed further, we introduce some notation and terminology. The index i will be used exclusively to indicate that a term is associated with the i^{th} mesh subinterval or mesh point and, depending on the context, it will take on values from either $\{0, \ldots, N-1\}$ or $\{0, \ldots, N\}$. The index j will be used exclusively to associate a term with the j^{th} component of the exact or approximate solution, and it will take on values from $\{1, \ldots, ncomp\}$. The width $(x_{i+1} - x_i)$ of the i^{th} mesh subinterval will be denoted by h_i and the largest mesh subinterval width will be denoted by h. We will denote the exact solution to a system of BVODEs as $u(x) = [u_1(x), \ldots, u_{ncomp}(x)]$ and COLNEW's approximate solution as $v(x) = [v_1(x), \ldots, v_{ncomp}(x)]$. The term **mesh data** will be used to refer to $\{z(v(x_i))\}$. Given an approximate solution $\hat{u}(x) = [\hat{u}_1(x), \ldots, \hat{u}_{ncomp}(x)]$ and an abscissa $\mu \in [0,1]$ we will use the term **derivative data** to refer to $\{f_1(x_i + \mu h_i, z(\hat{u}(x_i + \mu h_i))), \ldots, f_{ncomp}(x_i + \mu h_i, z(\hat{u}(x_i + \mu h_i)))\}$.

2 Background

2.1 COLNEW

The COLNEW/COLSYS software package [1][2][3] generates a spline (piecewise polynomial) approximation, $v_j(x)$, to each component of the solution of a BVODE (1)(2), using the mesh points as knots and imposing the following constraints:

- the spline approximating the j^{th} solution component must be C^{m_j-1} continuous,
- the spline approximating the j^{th} solution component must satisfy (1) at the images of the k Gauss points in each mesh subinterval,
- the spline approximating the j^{th} solution component must satisfy (2).

COLNEW represents this spline approximation using a local monomial basis [1][5]. The j^{th} component of this approximation on the i^{th} subinterval, $[x_i, x_{i+1}]$, is represented with $\tau = (x - x_i)/h_i$ as

$$v_j(x_i + \tau h_i) = \sum_{r=0}^{m_j-1} h_i^r \, \zeta_{j,i,r} \, \psi_r(\tau) \; + \; h_i^{m_j} \sum_{r=1}^{k} \phi_{j,i,r} \, \omega_{m_j,k,r}(\tau), \tag{3}$$

where

$\psi_r(\tau) = \tau^r / r!$,

$\omega_{m_j,k,r}(\tau)$ is the unique polynomial of degree $< k + m_j$ satisfying

$$\omega^{(l)}_{m_j,k,r}(0) = 0, \quad \text{for } l = 0, \ldots, m_j - 1$$
$$\omega^{(m_j)}_{m_j,k,r}(\rho_l) = \delta_{l,r}, \quad \text{for } l = 1, \ldots, k$$

$\rho_1, \ldots, \rho_k$ are the images of the Gauss points on $[0,1]$ (the collocation abscissa).

Note that the $\phi_{j,i,r}$ are the collocation **derivative data** values (i.e. $f_j(x_i+\rho_r h_i, z(v(x_i+\rho_r h_i))) \equiv v_j^{(m_j)}(x_i+\rho_r h_i) \equiv \phi_{j,i,r}$ for $r=1,\ldots,k$) and the $\zeta_{j,i,r}$ are the **mesh data** values (i.e. $v_j^{(r)}(x_i) \equiv \zeta_{j,i,r}$ for $r=0,\ldots,m_j-1$). In fact, COLNEW stores this approximation internally as tables of $\phi_{j,i,r}$ and $\zeta_{j,i,r}$ values. As shown in [6][4, pp.218-266], this approximation satisfies the following global error bound for $r=0,\ldots,m_j-1$ and $x\in[x_0,x_N]$:

$$v_j^{(r)}(x) - u_j^{(r)}(x) = \begin{cases} O(h^{2k}) & \text{if } x\in\{x_0,\ldots,x_N\} \\ O(h^{k+m_j-r}) & \text{otherwise} \end{cases} \tag{4}$$

2.2 Hermite-Birkhoff interpolation

Hermite-Birkhoff interpolation (see for example [9]) provides a natural way to develop and analyze superconvergent interpolants. Our objective is to determine, on an interval $[\alpha,\beta]$, a polynomial interpolant $w(x)$ which satisfies the following constraints, for a given d and m, with $d\geq 2m$:

- degree of $w(x)\leq d$,
- $w^{(r)}(\alpha)=A_r$, for $r=0,\ldots,m-1$,
- $w^{(r)}(\beta)=B_r$, for $r=0,\ldots,m-1$,
- $w^{(m)}(\alpha+\hat{h}\mu_r)=F_r$, for $r=0,\ldots,d-2m$,

where

$A_0,\ldots,A_{m-1},B_0,\ldots,B_{m-1},F_0,\ldots,F_{d-2m}$ are the data values that are to be interpolated,
$\mu_0,\ldots,\mu_{d-2m}$ are distinct interpolation abscissa which lie in $[0,1]$,
$\hat{h}=\beta-\alpha$.

We will refer to $\mu_0,\ldots,\mu_{d-2m}$ as the Hermite-Birkhoff abscissa. Note that, for some choices of Hermite-Birkhoff abscissa, there may be either no solution or multiple solutions. Later, we will impose a restriction on the abscissa so that a unique solution is always guaranteed. The Hermite-Birkhoff interpolant satisfying the above constraints can be represented with $\tau=(\frac{x-\alpha}{\beta-\alpha})$ as

$$w(x)=w(\alpha+\tau\hat{h})=\sum_{r=0}^{m-1}\hat{h}^r\left(A_r b_r(\tau)+B_r b_{m+r}(\tau)\right)+\hat{h}^m\sum_{r=2m}^{d}F_{r-2m}b_r(\tau), \tag{5}$$

where $b_0(\tau),\ldots,b_d(\tau)$ are polynomials of degree $\leq d$ satisfying the constraints

$$\left\{\begin{array}{l|l} b_l^{(r)}(0)=\delta_{l,r} & l=0,\ldots,d \\ b_l^{(r)}(1)=\delta_{l,r+m} & r=0,\ldots,m-1 \\ b_l^{(m)}(\mu_s)=\delta_{l,s+2m} & s=0,\ldots,d-2m \end{array}\right\} \tag{6}$$

We will refer to $b_0(\tau), \ldots, b_d(\tau)$ as Hermite-Birkhoff basis polynomials. If we represent these basis polynomials as

$$b_l(\tau) = \sum_{r=0}^{d} \Phi_{l,r} \frac{\tau^r}{r!}, \quad \text{for } l = 0, \ldots, d, \tag{7}$$

then the constraints in (6) can be expressed in matrix form as

$$H \, \Phi = I, \tag{8}$$

where I is the identity matrix,

$$\Phi = \begin{bmatrix} \Phi_{0,0} & \cdots & \Phi_{d,0} \\ \vdots & \vdots & \vdots \\ \Phi_{0,d} & \cdots & \Phi_{d,d} \end{bmatrix},$$

and

$$H = \begin{bmatrix}
1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 & \cdots & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\
\frac{1}{0!} & \frac{1}{1!} & \cdots & \cdots & \cdots & \cdots & \frac{1}{d!} \\
0 & \frac{1}{0!} & \cdots & \cdots & \cdots & \cdots & \frac{1}{(d-1)!} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & \cdots & \frac{1}{0!} & \cdots & \cdots & \frac{1}{(d-m+1)!} \\
0 & \cdots & \cdots & 0 & \frac{\mu_0^0}{0!} & \cdots & \frac{\mu_0^{d-m}}{(d-m)!} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & \cdots & 0 & \frac{\mu_{d-2m}^0}{0!} & \cdots & \frac{\mu_{d-2m}^{d-m}}{(d-m)!}
\end{bmatrix}.$$

If H is nonsingular then the coefficients of the basis polynomials (7) are the columns of H^{-1}. In this investigation, we will disallow those Hermite-Birkhoff abscissa that result in a singular matrix H. It can be shown that this restriction forbids exactly those abscissa which result in either no solution or multiple solutions to the above interpolation constraints. Therefore, the existence of H^{-1} automatically guarantees uniqueness of the Hermite-Birkhoff interpolant. For certain choices of Hermite-Birkhoff abscissa, the investigations [12][13] have explored whether or not there is a unique solution satisfying the above interpolation conditions.

Whenever Hermite-Birkhoff interpolants are discussed in this paper, we will assume that the abscissa are fixed on implementation (and not at run time) so that the basis polynomials can be predetermined by solving (8). Note that we gain the following advantages by predetermining the basis polynomials:

- the systems of equations in (8) can be solved to very high precision or solved exactly,

- the computational cost of solving (8) at run time is avoided,
- the abscissa can be chosen to minimize a bound on the round-off error associated with evaluating $w(x)$ for $x \in [\alpha, \beta]$,
- at runtime, the Hermite-Birkhoff interpolant is guaranteed to exist.

From standard results in approximation theory, we know that if this Hermite-Birkhoff interpolant is used to approximate a sufficiently differentiable function, then the associated interpolation error is $O(h^{d+1})$.

2.3 Survey of superconvergent interpolation schemes

Several approaches have been recently developed for the construction of superconvergent interpolants for COLNEW. In this section, we will review some of these approaches. In one approach [11], for each mesh subinterval, a superconvergent interpolant having global error of $O(h^{2k})$ is constructed simply by interpolating several **mesh data** values from adjacent subintervals. Clearly the error bounds for this approach will be sensitive to the ratio of mesh interval sizes and it may not be suitable when highly non-uniform meshes are used. In [11], a variation of the above approach can also be found. This approach constructs, for each component of $z(u(x))$, a separate superconvergent interpolant having $O(h^{2k})$ global error, by interpolating **mesh data** values from adjacent subintervals. An advantage of this approach is that, it approximates the derivatives of the solution with higher accuracy than the former approach. However, this approach also has difficulties with highly non-uniform meshes.

In order to avoid such sensitivity to non-uniform meshes, in each subinterval the superconvergent interpolant can be restricted to only use information from within the subinterval. If we insist on providing the same level of continuity as COLNEW, then we have to impose an additional constraint requiring the interpolant to reduce to the Gauss collocation scheme at the right end of the subinterval. Four approaches of this type are found in the literature.

In [8], the authors have used continuous Runge-Kutta (CRK) formulas [10] as a basis for constructing superconvergent interpolants having global error of $O(h^{2k})$. Their work is restricted to deriving CRK formulas for systems of first order BVODEs. The CRK formulas that they derive are designed to reuse COLNEW's collocation **derivative data** values, and do not require any extra Newton iterations to construct a superconvergent interpolant. One of the advantages of using CRK formulas is that, by choosing the CRK coefficients carefully, the number of extra evaluations of the differential equation can be minimized. However, the derivations of CRK formulas for higher order and mixed order BVODEs can be tedious and complex (especially for $k \geq 4$).

In [8], the authors have also discussed the details of a special type of CRK formula, derived from a **Bootstrap** approach for systems of first order BVODEs. One advantage of this approach is that the CRK continuous order conditions are automatically satisfied. CRK formulas of this type are therefore straightforward to derive (for any value of k). A disadvantage of this approach is that, it does not necessarily minimize the number of extra evaluations of the differential equation. Later, we will extend this approach to systems of mixed order BVODEs.

In [12][13], secondary collocation is defined and used as a basis for generating superconvergent interpolants. In this approach, a superconvergent interpolant on each subinterval is represented as a Hermite-Birkhoff polynomial interpolating the **mesh data** values as well as secondary collocation **derivative data** values. These secondary collocation **derivative data** values, which are defined implicitly, are determined by performing Newton iterations until the Hermite-Birkhoff interpolant satisfies the differential equation at the secondary collocation points. It has been shown that, for systems of first order BVODEs, this approach produces superconvergent interpolants having an $O(h^{2k})$ global error bound. Even though this approach was developed in the context of systems of mixed order BVODEs, a rigorous $O(h^{2k})$ global error bound for the associated superconvergent interpolant has not yet been established. However, for systems of mixed order BVODEs, it can be shown that the error bounds guaranteed for the Bootstrap approach (in section 3) also apply to this approach. Finally, it is worthwhile to note that the interpolants generated using this approach can also be viewed as a special type of CRK formula.

3 The Bootstrap Approach

The Bootstrap approach was originally introduced in [7] as a technique for constructing continuous approximations from the approximate solutions produced by discrete Runge-Kutta formulas for systems of first order initial value problems in ODEs. In this section, we will develop a general Bootstrap approach that can be used in conjunction with any BVODE solver based on Runge-Kutta formulas. Then we will discuss the details of using this Bootstrap approach in conjunction with COLNEW.

In this section, we will use the notation $w_a(x) = [w_{a,1}(x), \ldots, w_{a,ncomp}(x)]$, where '$a$' is an integer, to designate a piecewise polynomial approximation to the solution of (1)(2) which satisfies the error bound

$$w_{a,j}^{(r)}(x) = u_j^{(r)}(x) + O(h^{a+m_j-r}), \text{ for } x \in [x_0, x_N], \quad r = 0, \ldots, m_j - 1$$

We will assume that a discrete approximation to $z(u(x))$ at the mesh points, $\{\zeta_{j,i,r}\}$, is available and that, for some integer $\hat{a} > 0$, it satisfies the error bound

$$\zeta_{j,i,r} = u_j^{(r)}(x_i) + O(h^{\hat{a}+m_j-r}), \text{ for } r = 0, \ldots, m_j - 1. \tag{9}$$

We will also assume that an initial Bootstrap approximation, $w_{\bar{a}}(x)$, is available for some integer $\bar{a} > 0$. One such approximation, $w_1(x)$, can be constructed simply by interpolating the $\zeta_{j,i,r}$ values with the associated Hermite-Birkhoff interpolant on each mesh subinterval.

The entire Bootstrap process is essentially a sequence of steps, where $w_{\bar{a}+1}(x)$ is constructed from $w_{\bar{a}}(x)$ in the first Bootstrap step, then $w_{\bar{a}+2}(x)$ is constructed from $w_{\bar{a}+1}(x)$ in the second Bootstrap step and so on until $w_{\hat{a}}(x)$ is finally constructed. For any $a \in \{\bar{a}, \ldots, \hat{a}-1\}$, $w_{a+1}(x)$ can be constructed explicitly from $w_a(x)$ with $\tau = (x - x_i)/h_i$ as follows,

$$w_{a+1,j}(x_i + \tau h_i) = \sum_{r=0}^{m_j-1} h_i^r \left[\zeta_{j,i,r}\ b_{a,m_j,r}(\tau) + \zeta_{j,i+1,r}\ b_{a,m_j,m_j+r}(\tau)\right] + h_i^{m_j} \sum_{r=2m_j}^{a+m_j} F_{a,j,i,r-2m_j}\ b_{a,m_j,r}(\tau), \tag{10}$$

where

$F_{a,j,i,r-2m_j}$ is an $O(h^{a+1})$ approximation to $u_j^{(m_j)}(x_i + \mu_{a,m_j,r-2m_j} h_i)$, determined by evaluating $f_j(x, z(w_a(x)))$ at $x = x_i + \mu_{a,m_j,r-2m_j} h_i$,

$0 \le \mu_{a,m_j,0} < \ldots < \mu_{a,m_j,a-m_j} \le 1$ are a set of **Bootstrap abscissa** associated with the m_j^{th} order ODE components,

$b_{a,m_j,0}(\tau), \ldots, b_{a,m_j,a+m_j}(\tau)$ are the corresponding Hermite-Birkhoff basis polynomials of degree $a + m_j$ associated with the m_j^{th} order ODE components and they are determined as in (6)(7)(8) using $\mu_{a,m_j,0}, \ldots, \mu_{a,m_j,a-m_j}$ as abscissa.

Note that, from our assumption on $w_a(x)$, $z(w_a(x))$ is an $O(h^{a+1})$ approximation to $z(u(x))$. Therefore, with a Lipschitz assumption on (1), $f_j(x, z(w_a(x)))$ is an $O(h^{a+1})$ approximation to $u_j^{(m_j)}(x)$. Also note that, in (10), the $\zeta_{j,i,r}$ values, which have $O(h^{\hat{a}+m_j-r})$ error, are multiplied by h_i^r and the $F_{a,j,i,r-2m_j}$ values, which have $O(h^{a+1})$ error, are multiplied by $h_i^{m_j}$. Therefore $w_{a+1,j}^{(r)}(x)$ is an $O(h^{a+1+m_j-r})$ approximation to $u_j^{(r)}(x)$ for $r = 0, \ldots, m_j - 1$.

To reduce the computational cost of implementing the Bootstrap approach, we will impose two restrictions. The first restriction is that

$$\mu_{a,m_j,0} = 0 \ \text{ and } \ \mu_{a,m_j,1} = 1, \ \text{ for } a = \bar{a}, \ldots, \hat{a}-1,$$

and that the interpolated data values associated with these abscissa on the i^{th} subinterval are $F_{a,j,i,0} = \tilde{F}_{j,i}$ and $F_{a,j,i,1} = \tilde{F}_{j,i+1}$ respectively, where $\tilde{F}_{j,i} = f_j(x_i, \zeta_{1,i,0}, \ldots, \zeta_{1,i,m_1-1}, \ldots, \zeta_{ncomp,i,0}, \ldots, \zeta_{ncomp,i,m_{ncomp}-1})$. This restriction forces the Bootstrap approximations to use, as part of their interpolation data, the $O(h^{\hat{a}+1})$ approximation to $u_j^{(m_j)}(x_i)$ associated with the discrete approximation (9). Because of their high accuracy, these **derivative data** values do not need to be re-computed in any of the Bootstrap steps. Some of the advantages of this restriction are:

- the approximation $w_{a+1,j}(x)$ will have C^{m_j} continuity instead of C^{m_j-1},
- the Bootstrap process requires $2(\hat{a}-\bar{a})N$ fewer evaluations of $z(w_a(x))$ and $2(\hat{a}-\bar{a})N-(N+1)$ fewer evaluations of the differential equation than would otherwise be the case.

One should note that this restriction may lead to difficulties if the differential equation has singularities at or near the boundary points, since (in COLNEW) the boundary points are included among the mesh points. The second restriction is that

$$\forall (m_q > m_s),\ \{\mu_{a,m_q,0},\ldots,\mu_{a,m_q,a-m_q}\} \subseteq \{\mu_{a,m_s,0},\ldots,\mu_{a,m_s,a-m_s}\}.$$

This requirement reduces the number of evaluations of $z(w_a(x))$ in each step of the Bootstrap process by reusing as many of these values as possible. This requirement will also reduce the number of evaluations of the differential equation if a provision to separately evaluate individual components of (1) is not provided (which is the case in the COLNEW interface).

Note that, for $a \in \{\bar{a}+1,\ldots,\hat{a}-1\}$, $F_{a,j,i,r-2m_j}$ is determined by evaluating $f_j(x, z(w_a(x)))$ at $x = x_i + \mu_{a,m_j,r-2m_j} h_i$, and in order to compute $z(w_a(x))$ we need to evaluate

$$\left\{ b^{(q)}_{a-1,l,s}(\tau) \;\middle|\; \begin{array}{l} l \in \{m_1,\ldots,m_{ncomp}\} \\ s \in \{0,\ldots,a-1+l\} \\ q \in \{0,\ldots,l-1\} \end{array} \right\}$$

at $\tau = \mu_{a,m_j,r-2m_j}$. Therefore, there can be significant computational savings at runtime if these Hermite-Birkhoff basis polynomials and their derivatives are evaluated at $\mu_{a,m_j,r-2m_j}$ on implementation. As an added benefit, we also have the opportunity to evaluate these polynomials and their derivatives to very high precision.

Finally, one should note that the Bootstrap approach becomes more efficient as the order of the ODEs increase. This is because, during each Bootstrap step, an $(m+1)^{st}$ order ODE component requires one fewer **derivative data** value per subinterval than an m^{th} order ODE component (i.e. equation (10) requires $(a-m_j)$ **derivative data** values to construct $w_{a+1,j}(x)$). (Also see Table 4)

There are various implementation issues that arise when this Bootstrap approach is used in conjunction with COLNEW. First, we have to choose an initial Bootstrap approximation, $w_{\bar{a}}(x)$, that will be used as a starting point for the Bootstrap process. An obvious candidate for it is COLNEW's approximate solution (3), which corresponds to a $w_k(x)$ in our notation. In addition, if we set one of the abscissa for the first Bootstrap step, $\mu_{k,m_j,l}$, to coincide with one of COLNEW's collocation abscissa, ρ_s, then the **derivative data** value corresponding to $\mu_{k,m_j,l}$ on the i^{th} subinterval will be

$$F_{k,j,i,l} = f_j(x_i+\rho_s h_i, z(v(x_i+\rho_s h_i))) = v_j^{(m_j)}(x_i+\rho_s h_i) \equiv \phi_{j,i,s}.$$

Hence the value of $F_{k,j,i,l}$ can be retrieved from the output produced by COLNEW without doing any computation. Therefore, for the first Bootstrap step, there is an advantage to choose the Bootstrap abscissa from $\{\rho_1,\ldots,\rho_k\}$. Note that the j^{th} component of COLNEW's approximation, (3), is uniquely determined by $k+m_j$ pieces of information from each subinterval. Therefore, on each subinterval, $w_{k+1,j}(x)$ can use at most $k+m_j$ data values directly from COLNEW's approximation ($2m_j$ **mesh data** values and $k-m_j$ **derivative data** values); that is, any additional information would be linearly dependent and hence redundant. Since $w_{k+1,j}(x)$ requires $k+m_j+1$ pieces of interpolation data on each subinterval, it must interpolate at least one explicitly computed **derivative data** value on each subinterval. Two such **derivative data** values per subinterval are available from $\{\tilde{F}_{j,i}\}$.

There is also a theoretical result which justifies reusing the $\phi_{j,i,s}$ values in the first step of the Bootstrap process. In [4, pp.253-256], the leading term in the global error of (3) is shown to be

$$u_j^{(r)}(x) - v_j^{(r)}(x) = h_i^{k+m_j-r} u_j^{(k+m_j)}(x_i) P_{k,m_j}^{(r)}\left(\frac{x-x_i}{h_i}\right) + O(h_i^{k+m_j-r+1}) + O(h^{2k}) \quad (11)$$

where

$$P_{k,m_j}(\tau) = \frac{1}{k!(m_j-1)!} \int_0^\tau (t-\tau)^{m_j-1} \prod_{s=1}^{k} (t-\rho_s)\, dt.$$

Using numerical testing, we have verified that, for the range of m_j and k values supported by COLNEW, $\rho_1, \ldots, \rho_k$ are roots of $P_{k,m_j}^{(m_j)}(\tau)$. Hence, at $x = x_i + h_i\rho_s$, the error expression (11) corresponding to the m_j^{th} derivative of the j^{th} component satisfies

$$u_j^{(m_j)}(x_i+h_i\rho_s) - v_j^{(m_j)}(x_i+h_i\rho_s) \equiv u_j^{(m_j)}(x_i+h_i\rho_s) - \phi_{j,i,s} = O(h^{k+1}).$$

Recall from (10) that the **derivative data** values which are used in the construction of $w_{k+1,j}(x)$, should have an error bound of $O(h^{k+1})$. Therefore the $\phi_{j,i,s}$ values are indeed accurate enough to be used in the construction of $w_{k+1,j}(x)$.

There is also another way to achieve savings in computation at the beginning of the Bootstrap process. We will describe this alternative for the case where $m_1 = \ldots = m_{ncomp} = \hat{m}$. Let σ be one of the roots of $P_{k,\hat{m}_j}^{(\hat{m}_j-1)}(\tau)$. Then, according to (11), $z(v(x_i+\sigma h_i))$ has an error bound of $O(h^{k+2})$. Therefore $f_j(x_i+\sigma h_i, z(v(x_i+\sigma h_i)))$ is an $O(h^{k+2})$ approximation to $u_j^{(m_j)}(x_i+\sigma h_i)$. If sufficient number of such $O(h^{k+2})$ approximations to $u_j^{(m_j)}(x)$ are available, then we may be able to construct $w_{k+2,j}(x)$ directly from COLNEW's approximation. Note that here again $\{\tilde{F}_{j,i}\}$ values are at our disposal. Since it is not clear whether this alternative generalizes to mixed order systems of BVODEs and the costs are the same as those of the previous approach employing the $\phi_{j,i,s}$ values directly, we will not pursue it any further.

Next, we have to choose an appropriate $\hat{a}$. The obvious choice for $\hat{a}$ is the largest $\hat{a}$ for which the error bound (9) does not exceed $O(h^{2k})$ (i.e. $\hat{a}+m_j-r \leq 2k$ for $r=0,\ldots,m_j-1$). This $\hat{a}$ is $2k-mmax$. The approximate solution that will be constructed by the Bootstrap approach for this $\hat{a}$ will satisfy the error bound

$$w_{2k-mmax,j}^{(r)}(x) = u_j^{(r)}(x) + O(h^{2k+(m_j-mmax)-r}), \text{ for } r=0,\ldots,m_j-1.$$

Note that, because of the $O(h^{2k})$ error in the **mesh data** values, we cannot expect the approximations constructed in (10) to have an error bound better than $O(h^{2k})$. Therefore, those components that have reached the $O(h^{2k})$ error bound cannot be improved further by the Bootstrap process. However, those components which have not yet reached the $O(h^{2k})$ error bound can still be improved by the Bootstrap process, provided that we can somehow generate sufficiently accurate **derivative data** values. This concept gives rise to a variation of the Bootstrap approach which only improves the approximation for a selected subset of the components of the ODE (i.e. those which have not reached the $O(h^{2k})$ error bound), while keeping the approximation for the other components (i.e. those that have reached the $O(h^{2k})$ error bound) fixed. We will use the notation $\hat{w}_a(x)=[\hat{w}_{a,1}(x),\ldots,\hat{w}_{a,ncomp}(x)]$, where '$a$' is an integer, to designate an approximation to the solution of (1)(2) which satisfies the error bound

$$\hat{w}_{a,j}^{(r)}(x)=u_j^{(r)}(x)+O(h^{a+m_j-r})+O(h^{2k-r}),\ \text{for } r=0,\ldots,m_j-1.$$

During the $(a-k+1)^{st}$ Bootstrap step of this approach, $\hat{w}_{a+1}(x)$ is constructed from $\hat{w}_a(x)$ as follows

$$\hat{w}_{a+1,j}(x_i+\tau h_i)=\begin{cases}\hat{w}_{a,j}(x_i+\tau h_i), & \text{if } a+m_j\geq 2k,\\ \sum_{r=0}^{m_j-1} h_i^r[\zeta_{j,i,r}b_{a,m_j,r}(\tau)+\zeta_{j,i+1,r}b_{a,m_j,m_j+r}(\tau)] & \\ \quad +h_i^{m_j}\sum_{r=2m_j}^{a+m_j}\hat{F}_{a,j,i,r-2m_j}b_{a,m_j,r}(\tau), & \text{otherwise,}\end{cases}$$

where

$\hat{F}_{a,j,i,r-2m_j}$ is an $O(h^{a+1})$ approximation to $u_j^{(m_j)}(x_i+\mu_{a,m_j,r-2m_j}h_i)$, determined by evaluating $f_j(x,z(\hat{w}_a(x)))$ at $x=x_i+\mu_{a,m_j,r-2m_j}h_i$.

Note that this variation of the Bootstrap approach proceeds exactly like the standard Bootstrap approach during Bootstrap steps $1,\ldots,k-mmax$. However, when different orders are present in the system of ODEs, it carries on the Bootstrap process beyond the $(k-mmax)^{th}$ Bootstrap step while the standard Bootstrap approach does not do so. Next, we will determine the number of Bootstrap steps that this variation of the Bootstrap approach can perform and still improve the accuracy of the approximation.

When we described the procedure for computing $\hat{F}_{a,j,i,r-2m_j}$ values, we assumed that the computed values had an $O(h^{a+1})$ error bound (i.e. for $l\in\{j\mid a+m_j<2k\}$ and $x=x_i+\mu_{a,m_j,r-2m_j}h_i$, $f_l(x,z(\hat{w}_a(x)))=u_l^{(m_l)}(x)+O(h^{a+1})$). However, this assumption is not necessarily true in general. We will now derive a sufficient condition under which this assumption is valid. From our assumption on $\hat{w}_a(x)$, for $r=0,\ldots,m_j-1$, if $2k-r\geq a+1$, we can expect $\hat{w}_{a,j}^{(r)}(x)$ to have an error bound of at least $O(h^{a+1})$. Therefore, with a Lipschitz assumption, we can expect $f_l(x,z(\hat{w}_a(x)))$ to have an error bound of $O(h^{a+1})$ if the only components of $z(\hat{w}_a(x))$ that are actually used when evaluating $f_l(x,z(\hat{w}_a(x)))$ are $\{\hat{w}_{a,j}^{(r)}(x)\mid r=0,\ldots,m_j-1 \text{ and } 2k-r\geq a+1\}$. This sufficient condition can be formally expressed as

$$\begin{array}{l}\forall l\in\{j\mid a+m_j<2k\}\\ \forall s\in\{1,\ldots,ncomp\}\\ \forall r\in\{0,\ldots,m_s-1\}\end{array}\left[2k-r<a+1\Rightarrow\frac{\partial f_l}{\partial\left[u_s^{(r)}\right]}=0\right]. \qquad (12)$$

Therefore the Bootstrap process can proceed as long as (12) is satisfied. Note that for $a=k,\ldots,2k-mmax$, (12) is always true because for $r=0,\ldots,m_j-1$, $2k-r \geq a+1$. Therefore, for any system of BVODEs of the form (1), this Bootstrap approach can generate at least $\hat{w}_{2k-mmax+1}(x)$.

Now we will use (12) to characterize a class of BVODEs for which this Bootstrap approach is guaranteed to generate $O(h^{2k})$ approximations. In order to construct an $O(h^{2k})$ Bootstrap approximation from COLNEW's approximate solution, the Bootstrap process must construct $\hat{w}_{k+1}(x),\ldots,\hat{w}_{2k-mmin}(x)$, where $mmin = min(m_1,\ldots,m_{ncomp})$. Hence this class of BVODEs must satisfy (12) for $a=k,\ldots,2k-mmin-1$. This requirement can be simplified to

$$\begin{array}{l}\forall l \in \{1,\ldots,ncomp\}\\ \forall s \in \{1,\ldots,ncomp\}\end{array} \left[m_s > m_l \Rightarrow \forall r \in \{m_l+1,\ldots,m_s-1\}, \frac{\partial f_l}{\partial \left[u_s^{(r)}\right]} = 0 \right]. \quad (13)$$

Suppose that we allowed this Bootstrap process to always perform $k-mmin$ Bootstrap steps, as if (13) were satisfied. Then we may end up doing more Bootstrap steps than necessary (and introduce more round-off error), but the error bound on the approximation produced would still be the same as if we had stopped the Bootstrap process early using specific information available from the structures of the partial derivatives.

This Bootstrap approach may fail to produce $O(h^{2k})$ approximations if the mixed system of BVODEs does not satisfy (13). Another variation of this approach which will always produce $O(h^{2k})$ approximations will now be developed. Note that the system of mixed order BVODEs in (1)(2) can be converted to the following equivalent system of first order BVODEs

$$\tilde{u}_l'(x) = \begin{cases} f_j(x,\tilde{u}_1(x),\ldots,\tilde{u}_{m^*}(x)), & \text{if } l=\sum_{r=1}^{j} m_r \\ \tilde{u}_{l+1}(x), & \text{otherwise} \end{cases}, \quad \text{for } l=1,\ldots,m^* \quad (14)$$

with boundary conditions

$$g_l(\xi_l,\tilde{u}_1(\xi_l),\ldots,\tilde{u}_{m^*}(\xi_l))=0, \text{ for } l=1,\ldots,m^*$$

Also note that COLNEW's approximation, (3), to the solution of (1) can be converted to an $O(h^{k+1})$ approximation to the solution of (14) as follows

$$\tilde{v}_{S_j+r}(x) = v_j^{(r)}(x), \text{ for } j=1,\ldots,ncomp \text{ and } r=0,\ldots,m_j-1, \quad (15)$$

where

$$S_j = 1+\sum_{l=1}^{j-1} m_l.$$

Further, note that $\tilde{v}(x) = [\tilde{v}_1(x),\ldots,\tilde{v}_{m^*}(x)]$ has an $O(h^{2k})$ error bound at the mesh points and that it satisfies the BVODE (14) at the images of the k Gauss points on each subinterval. Finally, note that systems of first order BVODEs will always satisfy (13) because the sub-expression $m_s > m_l+1$ in (13) is clearly false. Therefore the Bootstrap approach can, using (15) as the initial approximation, construct an $O(h^{2k})$ approximation to the solution of (14). This approximation can then be viewed as an $O(h^{2k})$ approximation of $z(u(x))$. Note that, at the implementation level, the conversion from (1) to (14) and from (3) to (15) can be automated. An advantage of this approach is that the $O(h^{2k})$ error bound applies not only to the solution components, but also to the first m_j-1 derivatives. On the other hand the disadvantages include:

- as noted earlier, the Bootstrap approach becomes less efficient as the order of a BVODE component decreases (i.e. more **derivative data** values are required),
- when handling mixed order BVODEs directly, at most k–$mmin$ Bootstrap steps are performed, but, when handling its equivalent first order form, exactly $k-1$ Bootstrap steps are performed.
- it provides a separate polynomial approximation for each of the first m_j–1 derivatives of $u_j(x)$.

In order to avoid some of these disadvantages, we can modify this algorithm.

Suppose that we stopped the above first order Bootstrap process at the $(k-mmin\text{–}1)^{st}$ step. Let the resulting approximation be $\dot{w}(x)=[\dot{w}_1(x),\ldots,\dot{w}_{m^*}(x)]$. Then, $f_j(x,\dot{w}_1(x),\ldots,\dot{w}_{m^*}(x))$ will be an $O(h^{2k\text{–}mmin})$ approximation to $u_j^{(m_j)}(x)$. Hence an $O(h^{2k})$ approximation to the j^{th} component of the solution of (1) on the i^{th} subinterval can be constructed as a Hermite-Birkhoff interpolant which interpolates the $2m_j$ **mesh data** values as well as $(2k-2m_j)$ **derivative data** values computed using $f_j(x,\dot{w}_1(x),\ldots,\dot{w}_{m^*}(x))$. Note that, as a result of this modification, the number of Bootstrap steps has been reduced to $k-mmin$ steps and the approximation to the derivatives of the solution have been consolidated into the approximation to the solution. However, we give up the extra accuracy in the approximation to the derivatives, as a result of this modification.

4 Implementation and Numerical Results

In this section, we will describe an implementation of one version of the Bootstrap approach introduced in the previous section. We will refer to this particular implementation as BOOTS. Since most of the information required by BOOTS (such as $ncomp, m_1,\ldots,m_{ncomp}, N, x_i, \zeta_{j,i,r}$ and $\phi_{j,i,r}$) are available in COLNEW's work arrays, the only inputs that BOOTS will need are COLNEW's work arrays and a subroutine for evaluating (1).

Until now, we have only been concerned with the global errors of the superconvergent interpolants. However, it is sometimes useful to use superconvergent interpolants that have a local error bound that is $O(h)$ stronger than the global error. Since the discrete mesh data values for COLNEW have a local error bound of $O(h^{2k+1})$, the Bootstrap approach can construct superconvergent interpolants satisfying this stronger local error requirement by performing an extra

Bootstrap step beyond the standard number of Bootstrap steps. In BOOTS, we provide the user with an option to construct superconvergent interpolants satisfying this stronger local error bound requirement. In Tables 1 and 2, we list the expected error bounds for the approximations produced by BOOTS.

Type of BVODE	Global error bound
Any BVODE (1)	$O(h^{2k-mmax+m_j+1})+O(h^{2k})$
BVODE satisfying (13)	$O(h^{2k})$

Table 1: Global error bounds for BOOTS approximations

Type of BVODE	LE option	Local error bound
Any BVODE (1)	No	$O(h^{2k-mmax+m_j+1})+O(h^{2k})$
Any BVODE (1)	Yes	$O(h^{2k-mmax+m_j+2})+O(h^{2k+1})$
BVODE satisfying (13)	No	$O(h^{2k})$
BVODE satisfying (13)	Yes	$O(h^{2k+1})$

Table 2: Local error bounds for BOOTS approximations

For each value of k, we have chosen to provide a separate Bootstrap configuration which is designed to work with any combination of m_j values (provided that $m_j \leq k$). The abscissa values specifying the configuration of BOOTS were selected, after some numerical experiments, to minimize the amplitudes of the Hermite-Birkhoff basis polynomials. In Table 3 these abscissa are reported. In this table the abscissa for the first Bootstrap step are specified in terms of Gauss points $\{\rho_1, \ldots, \rho_k\}$ and, the abscissa values 0 and 1 are implicitly included in all the Bootstrap steps.

The computational cost of using BOOTS can be divided into three components. The first component is the cost of evaluating the derivative function (1). The number of times BOOTS evaluates (1) is reported in Table 4. The second component is the cost of evaluating $z(w_a(x))$ at the appropriate Bootstrap abscissa. In Table 5, we report the maximum possible number of floating point operations required by BOOTS to perform these evaluations. The third component of the cost is incurred performing tasks such as flow control and data management. This component is hard to quantify in hardware independent terms and is most likely to be small compared to the other two.

BOOTS was implemented in FORTRAN and our numerical results were obtained using the Sun f77 4.2 compiler on a Sun Ultra 2/2170 running Solaris 2.5. Before using the COLNEW package in our testing, we replaced line 1779 of colnew.f with **110 RHO(7)=.94910791234275852452D0** to correct a typographical error in the source code of COLNEW. Some of the test runs were conducted in quadruple precision. In these cases, the compiler flag **-r8** was used to automatically convert double precision variables and constants to quadruple

k	BOOTS step	BOOTS abscissa			
		$m_j=1$	$m_j=2$	$m_j=3$	$m_j=4$
1	1	$\{\}$			
	LE opt	$\{\}$			
2	1	$\{\}$	$\{\}$		
	LE opt	$\{\frac{1}{3}\}$	$\{\}$		
3	1	$\{\rho_1\}$	$\{\}$	$\{\}$	
	2	$\{\frac{1}{3},\frac{2}{3}\}$			
	LE opt	$\{\frac{1}{2},\frac{1}{3},\frac{4}{5}\}$	$\{\frac{1}{2}\}$	$\{\}$	
4	1	$\{\rho_2,\rho_3\}$	$\{\rho_2\}$	$\{\}$	$\{\}$
	2	$\{\frac{1}{3},\frac{4}{5},\frac{1}{2}\}$	$\{\frac{1}{3},\frac{4}{5}\}$		
	3	$\{\frac{1}{3},\frac{4}{5},\frac{1}{5},\frac{2}{3}\}$			
	LE opt	$\{\frac{30}{100},\frac{50}{100},\frac{75}{100},\frac{10}{100},\frac{85}{100}\}$	$\{\frac{30}{100},\frac{50}{100},\frac{75}{100}\}$	$\{\frac{30}{100}\}$	$\{\}$
5	1	$\{\rho_1,\rho_3,\rho_4\}$	$\{\rho_2,\rho_4\}$	$\{\rho_2\}$	$\{\}$
	2	$\{\frac{3}{14},\frac{11}{14},\frac{6}{14},\frac{8}{14}\}$	$\{\frac{3}{14},\frac{11}{14},\frac{6}{14}\}$	$\{\frac{3}{14},\frac{11}{14}\}$	
	3	$\{\frac{3}{14},\frac{11}{14},\frac{6}{14},\frac{8}{14},\frac{1}{14}\}$	$\{\frac{3}{14},\frac{11}{14},\frac{6}{14},\frac{8}{14}\}$		
	4	$\{\frac{3}{14},\frac{11}{14},\frac{6}{14},\frac{8}{14},\frac{4}{14},\frac{10}{14}\}$			
	LE opt	$\{\frac{3}{6},\frac{1}{6},\frac{4}{6},\frac{2}{6},\frac{5}{6},\frac{1}{12},\frac{9}{12}\}$	$\{\frac{3}{6},\frac{1}{6},\frac{4}{6},\frac{2}{6},\frac{5}{6}\}$	$\{\frac{3}{6},\frac{1}{6},\frac{4}{6}\}$	$\{\frac{3}{6}\}$
6	1	$\{\rho_3,\rho_4,\rho_2,\rho_5\}$	$\{\rho_3,\rho_4,\rho_2\}$	$\{\rho_3,\rho_4\}$	$\{\rho_3,\rho_4\}$
	2	$\{\frac{3}{9},\frac{6}{9},\frac{1}{6},\frac{5}{6},\frac{2}{5}\}$	$\{\frac{3}{9},\frac{6}{9},\frac{1}{6},\frac{5}{6}\}$	$\{\frac{3}{9},\frac{6}{9},\frac{1}{6}\}$	$\{\frac{3}{9},\frac{6}{9}\}$
	3	$\{\frac{3}{9},\frac{6}{9},\frac{2}{9},\frac{7}{9},\frac{4}{9},\frac{5}{9}\}$	$\{\frac{3}{9},\frac{6}{9},\frac{2}{9},\frac{7}{9},\frac{4}{9}\}$	$\{\frac{3}{9},\frac{6}{9},\frac{2}{9},\frac{7}{9}\}$	
	4	$\{\frac{3}{9},\frac{6}{9},\frac{2}{9},\frac{7}{9},\frac{4}{9},\frac{5}{9},\frac{1}{9}\}$	$\{\frac{3}{9},\frac{6}{9},\frac{2}{9},\frac{7}{9},\frac{4}{9},\frac{5}{9}\}$		
	5	$\{\frac{3}{9},\frac{6}{9},\frac{2}{9},\frac{7}{9},\frac{4}{9},\frac{5}{9},\frac{1}{9},\frac{8}{9}\}$			
	LE opt	$\{\frac{8}{100},\frac{38}{100},\frac{79}{100},\frac{19}{100},\frac{59}{100},\frac{48}{100},\frac{90}{100},\frac{28}{100},\frac{69}{100}\}$	$\{\frac{8}{100},\frac{38}{100},\frac{79}{100},\frac{19}{100},\frac{59}{100},\frac{48}{100},\frac{90}{100}\}$	$\{\frac{8}{100},\frac{38}{100},\frac{79}{100},\frac{19}{100},\frac{59}{100}\}$	$\{\frac{8}{100},\frac{38}{100},\frac{79}{100}\}$
7	1	$\{\rho_1,\rho_2,\rho_3,\rho_4,\rho_6\}$	$\{\rho_2,\rho_3,\rho_5,\rho_6\}$	$\{\rho_2,\rho_3,\rho_5,\rho_6\}$	$\{\rho_3,\rho_5\}$
	2	$\{\frac{1}{7},\frac{3}{7},\frac{6}{7},\frac{4}{7},\frac{2}{7},\frac{5}{7}\}$	$\{\frac{1}{7},\frac{3}{7},\frac{6}{7},\frac{4}{7},\frac{2}{7}\}$	$\{\frac{1}{7},\frac{3}{7},\frac{6}{7},\frac{4}{7}\}$	$\{\frac{1}{7},\frac{3}{7},\frac{6}{7}\}$
	3	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11},\frac{6}{11},\frac{1}{11}\}$	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11},\frac{6}{11}\}$	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11}\}$	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11}\}$
	4	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11},\frac{6}{11},\frac{1}{11},\frac{10}{11}\}$	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11},\frac{6}{11},\frac{1}{11}\}$	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11},\frac{6}{11}\}$	
	5	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11},\frac{6}{11},\frac{3}{11},\frac{8}{11},\frac{1}{11}\}$	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11},\frac{6}{11},\frac{3}{11},\frac{8}{11}\}$		
	6	$\{\frac{2}{11},\frac{4}{11},\frac{7}{11},\frac{9}{11},\frac{5}{11},\frac{6}{11},\frac{3}{11},\frac{8}{11},\frac{1}{11},\frac{10}{11}\}$			
	LE opt	$\{\frac{16}{100},\frac{32}{100},\frac{49}{100},\frac{67}{100},\frac{82}{100},\frac{41}{100},\frac{74}{100},\frac{24}{100},\frac{57}{100},\frac{6}{100},\frac{91}{100}\}$	$\{\frac{16}{100},\frac{32}{100},\frac{49}{100},\frac{67}{100},\frac{82}{100},\frac{41}{100},\frac{74}{100},\frac{24}{100},\frac{57}{100}\}$	$\{\frac{16}{100},\frac{32}{100},\frac{49}{100},\frac{67}{100},\frac{82}{100},\frac{41}{100},\frac{74}{100}\}$	$\{\frac{16}{100},\frac{32}{100},\frac{49}{100},\frac{67}{100},\frac{82}{100}\}$

Table 3: BOOTS abscissa

	Without local error option				With local error option			
$mmin =$	1	2	3	4	1	2	3	4
$k=1$	1	-	-	-	1	-	-	-
$k=2$	1	1	-	-	2	1	-	-
$k=3$	3	1	1	-	6	2	1	-
$k=4$	8	3	1	1	13	6	2	1
$k=5$	16	8	3	1	23	13	6	2
$k=6$	27	16	8	3	36	23	13	6
$k=7$	41	27	16	8	52	36	23	13

Table 4: Number of times BOOTS evaluates (1) in each subinterval

	Without local error option				With local error option			
$mmin =$	1	2	3	4	1	2	3	4
$k=1$	0	-	-	-	0	-	-	-
$k=2$	0	0	-	-	6	0	-	-
$k=3$	16	0	0	-	40	8	0	-
$k=4$	70	20	0	0	120	50	10	0
$k=5$	150	70	20	0	220	120	50	10
$k=6$	312	180	84	24	420	264	144	60
$k=7$	541	348	197	88	695	474	295	158

Table 5: Number of FLOPS per $(N \times m^*)$ spent evaluating $z(w_a(x))$

precision. The COLNEW code that was used in the quadruple precision test runs was obtained by making the following modifications to the standard double precision package:

- each instance of **REAL *8** was replaced with **DOUBLE PRECISION**,
- the accuracy of the constants representing the Gauss points was increased to 34 significant digits.

We divide the numerical tests into two sets. The first set of tests verifies the error bounds of the superconvergent approximations from BOOTS on uniform meshes and the second set of tests evaluates the performance of BOOTS when it is used to improve COLNEW's approximation on a mesh that is determined by COLNEW's adaptive mesh selection algorithm.

We will report the performance on two problems. The test problem [4, p23] is a system of non-linear BVODEs,

$$\epsilon f'''' + f f''' + g g' = 0, \quad \epsilon g'' + f g' - f' g = 0, \quad \epsilon = 0.075, \tag{16}$$
$$f(0) = f(1) = f'(0) = f'(1) = 0, \quad g(0) = -1, \quad g(1) = 1.$$

Note that this system satisfies (13) and hence we should observe an $O(h^{2k})$ error for the BOOTS approximation and a $O(h^{k+mmin})$ error for COLNEW's continuous approximation. Since there is no known analytical solution for (16), we will use a numerical approximation (obtained using a quadruple precision version of COLNEW with k=7 and TOL=1D-27) to determine a reference solution. Finally, note that the above system of BVODEs has many equivalent mixed order forms. After writing (16) as

$$g''(x) = f_1(x, g(x), g'(x), f(x), f'(x), f''(x), f'''(x)),$$

$$f''''(x) = f_2(x, g(x), g'(x), f(x), f'(x), f''(x), f'''(x)),$$

it can be converted to the equivalent mixed order systems shown in Table 6. We will use these BVODEs to test BOOTS under different mixed order configurations. The second test problem [11] is a system of linear BVODEs

$$u_1''' = u_1' - 4e^{2x}u_2 + 4(x^2+1)e^x, \quad u_2' = xu_1'' - xu_1 - u_2 - 4x^2e^x + (2x-1)e^{-x}, \tag{17}$$

$$u_1(0) = u_2(0) = u_1(1) = u_2(1) = 0.$$

The exact solution to this problem is $u_1(x) = x(x-1)e^x$, $u_2(x) = x(x-1)e^{-x}$. Note that this system of BVODEs does not satisfy (13). When submitting this problem to COLNEW, we set the parameter IPAR(1) to 0, to enable it to solve this problem without performing any Newton iterations.

mixed order configuration	system of BVODEs
$m=1,1,1,1,1,1$	$u_1' = u_2,\ u_2' = f_1(x, z(u)),\ u_3' = u_4,\ u_4' = u_5,\ u_5' = u_6,$ $u_6' = f_2(x, z(u))$
$m=2,1,1,1,1$	$u_1'' = f_1(x, z(u)),\ u_2' = u_3,\ u_3' = u_4,\ u_4' = u_5,\ u_5' = f_2(x, z(u))$
$m=2,2,2$	$u_1'' = f_1(x, z(u)),\ u_2'' = u_3,\ u_3'' = f_2(x, z(u))$
$m=1,1,1,3$	$u_1' = u_2,\ u_2 = f_1(x, z(u)),\ u_3' = u_4,\ u_4''' = f_2(x, z(u))$
$m=2,1,3$	$u_1'' = f_1(x, z(u)),\ u_2' = u_3,\ u_3''' = f_2(x, z(u))$
$m=1,1,4$	$u_1' = u_2,\ u_2' = f_1(x, z(u)),\ u_3'''' = f_2(x, z(u))$
$m=2,4$	$u_2'' = f_1(x, z(u)),\ u_2'''' = f_2(x, z(u))$

Table 6: BVODEs equivalent to (16)

In our first set of tests, we will verify the error bounds given in Table 1 by studying the actual rate at which the BOOTS approximations converge to the true solution. To estimate the convergence rate, we generate two approximations, one over a uniform mesh of size 2^{r+1} and the other over a uniform mesh of size 2^r and define the **convergence ratio** to be the ratio of the maximum error of the **course mesh approximation** to the maximum error of the **fine mesh approximation**. The base-2 logarithm of this ratio should be a reasonable indication of the **convergence rate**. In order to measure the convergence rate as described above, we had to prevent COLNEW from redistributing the initial uniform mesh. This was accomplished by setting the parameters IPAR(8) to 2 and and IPAR(10) to 2 as described in [8]. The maximum error in the continuous approximations was determined by sampling the error at 2561 evenly spaced points over the interval of intergration. The Tables 7-14 report statistics which reflect the performance on these test problems. In these tables we use the notation:

CR	- Convergence Rate (expressed as the closest integer)
GE	- Global Error
w le	- BOOTS approximation constructed with local error option
w/o le	- BOOTS approximation constructed without local error option

In Tables 7-13 we report the maximum error as well as the convergence rate of the approximations constructed by COLNEW and BOOTS for the various mixed order configurations of (16). For $k=1,2,3,4$ (Tables 7-10) the tests were conducted in double precision and for $k=5,6,7$ (Tables 11-13) the tests were conducted in quadruple precision. Note that, for $k=4$ (Table 10), round-off error started to dominate the error observed in the **mesh data** values when more than 16 mesh points were used. In fact, for k values 5, 6 and 7, we found that the error in the **mesh data** values was dominated by round-off error when more than 4 mesh points were used in the double precision tests. Even in the quadruple precision test results for $k=7$ (Table 13), the error in the **mesh data** values was dominated by round-off error when more than 32 mesh points were used. However, such early dominance of round-off error is inevitable given the high convergence ratios of 2^{10}, 2^{12} and 2^{14} for $k=5,6,7$. The results in Tables 7-13 do confirm the theoretical convergence rates predicted by our analysis. In Table 15, we report the time required to construct the approximations generated for k=4 (see [14] for the times for other values of k). The times reported are measured in seconds on a SUN Ultra 2/2170. For $k \leq 4$, BOOTS constructed its approximation in less than 6% of the time that it took COLNEW to construct its approximation to this test problem. The cost of constructing BOOTS approximations for k values 5, 6 and 7 were roughly 11%, 14% and 18% respectively.

We have also conducted the convergence rate tests on problem (17). This problem is interesting because we can only guarantee an error bound of $O(h^{2k-1})$, instead of $O(h^{2k})$, for this problem. In Table 14, we report the maximum errors as well as the observed convergence rate of the COLNEW and BOOTS approximations for test runs conducted in quadruple precision. These results confirm that the approximations produced by BOOTS for this problem have an error bound of at least $O(h^{2k-1})$. For k values 3,...,7, BOOTS constructed its approximation in roughly 41%, 70%, 85%, 103% and 122% of the time that it took COLNEW to generate its approximation. This seemingly poor performance of BOOTS is due to the fact that COLNEW computes its approximations to linear BVODEs much more efficiently than non-linear BVODEs because no Newton iterations are required.

m	N	COLNEW				BOOTS			
		Mesh data		Cont. aprx.		w/o le		w le	
		CR	GE	CR	GE	CR	GE	CR	GE
1,1,1,1,1,1	8	2	$1.3{\cdot}10^{-01}$	2	$1.3{\cdot}10^{-01}$	2	$1.3{\cdot}10^{-01}$	2	$1.3{\cdot}10^{-01}$
	16	2	$3.3{\cdot}10^{-02}$	2	$3.3{\cdot}10^{-02}$	2	$3.3{\cdot}10^{-02}$	2	$3.3{\cdot}10^{-02}$
	32	2	$8.2{\cdot}10^{-03}$	2	$8.2{\cdot}10^{-03}$	2	$8.2{\cdot}10^{-03}$	2	$8.2{\cdot}10^{-03}$
	64	2	$2.0{\cdot}10^{-03}$	2	$2.0{\cdot}10^{-03}$	2	$2.0{\cdot}10^{-03}$	2	$2.0{\cdot}10^{-03}$
	128	2	$5.1{\cdot}10^{-04}$	2	$5.1{\cdot}10^{-04}$	2	$5.1{\cdot}10^{-04}$	2	$5.1{\cdot}10^{-04}$
	256	2	$1.3{\cdot}10^{-04}$	2	$1.3{\cdot}10^{-04}$	2	$1.3{\cdot}10^{-04}$	2	$1.3{\cdot}10^{-04}$

Table 7: Convergence rate of problem 1 for k=1 (Double precision)

The next set of tests evaluate the overall performance of BOOTS as a technique for improving COLNEW's approximation. In these tests the problems were submitted to COLNEW with error tolerances set only on the solution (i.e. 0^{th} derivative) components and COLNEW was allowed to adaptively choose an appropriate mesh for its approximation. The tolerances that we used were 10^{-4}, 10^{-6}, 10^{-8}, and 10^{-10}. BOOTS was then used to construct the superconvergent approximations from the approximations generated by COLNEW. In Tables 16-22 we list the global errors in the COLNEW and BOOTS approximations to the solution of (16) under different mixed order configurations and also the time required to generate these approximations. For $k=1,\ldots,5$ (Tables 16-20) the test runs were conducted in double precision. However, for $k=6$ and $k=7$ we found the round-off error to be so dominant (in the double precision test runs) that the approximations produced by BOOTS were frequently inferior to the approximation from COLNEW. Therefore, for $k=6$ and $k=7$ we have provided the results obtained using quadruple precision test runs. Apart from the inaccuracies caused due to round-off error, these results confirm that BOOTS does provide significantly improved approximations. For k values 1,...,7, the extra cost involved in constructing BOOTS approximations was roughly 0.5%, 2.3%, 6%, 14%, 30% and 30% respectively.

Many other variations of the above test runs (such as imposing error tolerance on all components of COLNEW's approximations and test runs of problem (16) with ϵ set to 0.0075) were also conducted and they produced results similar to those presented here.

5 Conclusion

We have developed the Bootstrap approach as an efficient yet simple means of providing high accuracy, C^{m_j}-continuous, approximations from the approximations produced by a method based on collocation. We have provided the theoretical justification for this approach and identified some advantages and limitations. We have also discussed a specific implementation of this approach. The numerical results that were obtained using this implementation confirmed

m	N	COLNEW				BOOTS			
		Mesh data		Cont. aprx.		w/o le		w le	
		CR	GE	CR	GE	CR	GE	CR	GE
1,1,1,1,1,1	4	4	$4.2{\cdot}10^{-03}$	3	$8.2{\cdot}10^{-03}$	4	$4.7{\cdot}10^{-03}$	4	$5.9{\cdot}10^{-03}$
	8	4	$2.6{\cdot}10^{-04}$	3	$9.8{\cdot}10^{-04}$	4	$3.1{\cdot}10^{-04}$	4	$3.5{\cdot}10^{-04}$
	16	4	$1.6{\cdot}10^{-05}$	3	$1.3{\cdot}10^{-04}$	4	$1.9{\cdot}10^{-05}$	4	$2.0{\cdot}10^{-05}$
	32	4	$1.0{\cdot}10^{-06}$	3	$1.8{\cdot}10^{-05}$	4	$1.3{\cdot}10^{-06}$	4	$1.1{\cdot}10^{-06}$
	64	4	$6.3{\cdot}10^{-08}$	3	$2.4{\cdot}10^{-06}$	4	$8.2{\cdot}10^{-08}$	4	$6.6{\cdot}10^{-08}$
	128	4	$4.0{\cdot}10^{-09}$	3	$3.1{\cdot}10^{-07}$	4	$5.3{\cdot}10^{-09}$	4	$4.0{\cdot}10^{-09}$
2,1,1,1,1	4	4	$4.3{\cdot}10^{-03}$	3	$8.2{\cdot}10^{-03}$	4	$4.7{\cdot}10^{-03}$	4	$4.3{\cdot}10^{-03}$
	8	4	$2.7{\cdot}10^{-04}$	3	$9.7{\cdot}10^{-04}$	4	$3.1{\cdot}10^{-04}$	4	$2.7{\cdot}10^{-04}$
	16	4	$1.7{\cdot}10^{-05}$	3	$1.2{\cdot}10^{-04}$	4	$2.0{\cdot}10^{-05}$	4	$1.7{\cdot}10^{-05}$
	32	4	$1.0{\cdot}10^{-06}$	3	$1.4{\cdot}10^{-05}$	4	$1.2{\cdot}10^{-06}$	4	$1.0{\cdot}10^{-06}$
	64	4	$6.5{\cdot}10^{-08}$	3	$1.8{\cdot}10^{-06}$	4	$7.8{\cdot}10^{-08}$	4	$6.5{\cdot}10^{-08}$
	128	4	$4.1{\cdot}10^{-09}$	3	$2.2{\cdot}10^{-07}$	4	$4.9{\cdot}10^{-09}$	4	$4.1{\cdot}10^{-09}$
2,2,2	4	4	$1.3{\cdot}10^{-03}$	4	$1.3{\cdot}10^{-03}$	4	$1.3{\cdot}10^{-03}$	4	$1.3{\cdot}10^{-03}$
	8	4	$9.1{\cdot}10^{-05}$	4	$9.2{\cdot}10^{-05}$	4	$9.1{\cdot}10^{-05}$	4	$9.1{\cdot}10^{-05}$
	16	4	$5.9{\cdot}10^{-06}$	4	$6.3{\cdot}10^{-06}$	4	$5.9{\cdot}10^{-06}$	4	$5.9{\cdot}10^{-06}$
	32	4	$3.7{\cdot}10^{-07}$	4	$4.3{\cdot}10^{-07}$	4	$3.7{\cdot}10^{-07}$	4	$3.7{\cdot}10^{-07}$
	64	4	$2.3{\cdot}10^{-08}$	4	$2.9{\cdot}10^{-08}$	4	$2.3{\cdot}10^{-08}$	4	$2.3{\cdot}10^{-08}$
	128	4	$1.4{\cdot}10^{-09}$	4	$1.8{\cdot}10^{-09}$	4	$1.4{\cdot}10^{-09}$	4	$1.4{\cdot}10^{-09}$

Table 8: Convergence rate of problem 1 for k=2 (Double precision)

that the approximations produced by the Bootstrap approach do satisfy the error bounds predicted by our analysis (even for the rather course meshes often encountered on practical problems). The results also show that the incremental cost required to generate these approximations is relatively small (particularly when $k \leq 4$).

Although we have presented our approach as a **post processing** procedure which can be used to improve the accuracy of an underlying numerical solution, it can be used in other ways. For example, the error control and mesh selection strategies of COLNEW could be modified so that they are based on estimates of the error in the new superconvergent interpolants (rather than on estimates of the error in lower order interpolants as is the case now).

Acknowledgement

The authors would like to thank P.H. Muir for several helpful discussions and suggestions.

m	N	COLNEW				BOOTS			
		Mesh data		Cont. aprx.		w/o le		w le	
		CR	GE	CR	GE	CR	GE	CR	GE
1,1,1,1,1,1	4	6	$1.3{\cdot}10^{-05}$	3	$6.9{\cdot}10^{-04}$	5	$3.9{\cdot}10^{-04}$	7	$4.1{\cdot}10^{-05}$
	8	6	$2.6{\cdot}10^{-07}$	3	$6.3{\cdot}10^{-05}$	6	$7.7{\cdot}10^{-06}$	6	$5.2{\cdot}10^{-07}$
	16	6	$4.3{\cdot}10^{-09}$	4	$5.3{\cdot}10^{-06}$	6	$1.4{\cdot}10^{-07}$	6	$6.2{\cdot}10^{-09}$
	32	6	$6.9{\cdot}10^{-11}$	4	$3.8{\cdot}10^{-07}$	6	$2.3{\cdot}10^{-09}$	6	$7.4{\cdot}10^{-11}$
	64	6	$1.1{\cdot}10^{-12}$	4	$2.6{\cdot}10^{-08}$	6	$3.7{\cdot}10^{-11}$	6	$1.1{\cdot}10^{-12}$
	128	6	$2.0{\cdot}10^{-14}$	4	$1.7{\cdot}10^{-09}$	6	$5.8{\cdot}10^{-13}$	6	$2.3{\cdot}10^{-14}$
2,1,1,1,1	4	6	$1.0{\cdot}10^{-05}$	3	$5.5{\cdot}10^{-04}$	5	$9.4{\cdot}10^{-05}$	5	$1.2{\cdot}10^{-04}$
	8	6	$2.0{\cdot}10^{-07}$	3	$6.2{\cdot}10^{-05}$	6	$1.9{\cdot}10^{-06}$	6	$2.5{\cdot}10^{-06}$
	16	6	$3.3{\cdot}10^{-09}$	4	$5.3{\cdot}10^{-06}$	6	$3.4{\cdot}10^{-08}$	6	$4.5{\cdot}10^{-08}$
	32	6	$5.3{\cdot}10^{-11}$	4	$3.8{\cdot}10^{-07}$	6	$5.7{\cdot}10^{-10}$	6	$7.5{\cdot}10^{-10}$
	64	6	$8.2{\cdot}10^{-13}$	4	$2.6{\cdot}10^{-08}$	6	$9.3{\cdot}10^{-12}$	6	$1.2{\cdot}10^{-11}$
	128	6	$1.5{\cdot}10^{-14}$	4	$1.7{\cdot}10^{-09}$	6	$1.5{\cdot}10^{-13}$	6	$1.9{\cdot}10^{-13}$
2,2,2	4	3	$1.4{\cdot}10^{-05}$	4	$2.7{\cdot}10^{-05}$	5	$1.4{\cdot}10^{-05}$	4	$1.4{\cdot}10^{-05}$
	8	6	$2.8{\cdot}10^{-07}$	4	$1.2{\cdot}10^{-06}$	6	$2.8{\cdot}10^{-07}$	6	$2.8{\cdot}10^{-07}$
	16	6	$4.6{\cdot}10^{-09}$	5	$4.8{\cdot}10^{-08}$	6	$5.1{\cdot}10^{-09}$	6	$4.6{\cdot}10^{-09}$
	32	6	$7.4{\cdot}10^{-11}$	5	$1.7{\cdot}10^{-09}$	6	$8.8{\cdot}10^{-11}$	6	$7.4{\cdot}10^{-11}$
	64	6	$1.2{\cdot}10^{-12}$	5	$5.8{\cdot}10^{-11}$	6	$1.4{\cdot}10^{-12}$	6	$1.2{\cdot}10^{-12}$
	128	6	$1.8{\cdot}10^{-14}$	5	$1.8{\cdot}10^{-12}$	6	$2.4{\cdot}10^{-14}$	6	$1.8{\cdot}10^{-14}$
1,1,1,3	4	6	$1.2{\cdot}10^{-05}$	3	$6.9{\cdot}10^{-04}$	6	$4.6{\cdot}10^{-05}$	6	$1.3{\cdot}10^{-05}$
	8	6	$1.9{\cdot}10^{-07}$	3	$6.1{\cdot}10^{-05}$	6	$7.6{\cdot}10^{-07}$	6	$2.0{\cdot}10^{-07}$
	16	6	$3.0{\cdot}10^{-09}$	4	$4.5{\cdot}10^{-06}$	6	$1.2{\cdot}10^{-08}$	6	$3.1{\cdot}10^{-09}$
	32	6	$4.7{\cdot}10^{-11}$	4	$3.0{\cdot}10^{-07}$	6	$2.0{\cdot}10^{-10}$	6	$4.8{\cdot}10^{-11}$
	64	6	$7.4{\cdot}10^{-13}$	4	$2.0{\cdot}10^{-08}$	6	$3.1{\cdot}10^{-12}$	6	$7.4{\cdot}10^{-13}$
	128	6	$1.5{\cdot}10^{-14}$	4	$1.2{\cdot}10^{-09}$	6	$5.1{\cdot}10^{-14}$	6	$1.6{\cdot}10^{-14}$
2,1,3	4	5	$7.3{\cdot}10^{-07}$	4	$4.1{\cdot}10^{-05}$	5	$1.2{\cdot}10^{-05}$	6	$7.3{\cdot}10^{-07}$
	8	6	$1.0{\cdot}10^{-08}$	4	$3.0{\cdot}10^{-06}$	6	$2.2{\cdot}10^{-07}$	6	$1.0{\cdot}10^{-08}$
	16	6	$1.6{\cdot}10^{-10}$	4	$2.0{\cdot}10^{-07}$	6	$3.6{\cdot}10^{-09}$	6	$1.6{\cdot}10^{-10}$
	32	6	$2.4{\cdot}10^{-12}$	4	$1.3{\cdot}10^{-08}$	6	$5.9{\cdot}10^{-11}$	6	$2.4{\cdot}10^{-12}$
	64	6	$3.8{\cdot}10^{-14}$	4	$8.3{\cdot}10^{-10}$	6	$9.4{\cdot}10^{-13}$	6	$3.8{\cdot}10^{-14}$
	128	5	$9.1{\cdot}10^{-16}$	4	$5.2{\cdot}10^{-11}$	6	$1.5{\cdot}10^{-14}$	5	$9.2{\cdot}10^{-16}$

Table 9: Convergence rate of problem 1 for k=3 (Double precision)

m	N	COLNEW				BOOTS			
		Mesh data		Cont. aprx.		w/o le		w le	
		CR	GE	CR	GE	CR	GE	CR	GE
1,1,1,1,1,1	4	8	$4.3\cdot10^{-07}$	4	$8.9\cdot10^{-05}$	8	$2.2\cdot10^{-06}$	8	$4.9\cdot10^{-07}$
	8	8	$1.9\cdot10^{-09}$	4	$4.2\cdot10^{-06}$	8	$8.9\cdot10^{-09}$	8	$2.1\cdot10^{-09}$
	16	8	$7.6\cdot10^{-12}$	5	$1.6\cdot10^{-07}$	8	$3.5\cdot10^{-11}$	7	$1.2\cdot10^{-11}$
	32	7	$4.3\cdot10^{-14}$	5	$5.6\cdot10^{-09}$	6	$4.4\cdot10^{-13}$	2	$2.3\cdot10^{-12}$
	64	0	$3.0\cdot10^{-14}$	5	$1.8\cdot10^{-10}$	2	$1.6\cdot10^{-13}$	2	$8.1\cdot10^{-13}$
	128	0	$2.0\cdot10^{-14}$	5	$5.8\cdot10^{-12}$	1	$6.5\cdot10^{-14}$	1	$3.7\cdot10^{-13}$
2,1,1,1,1	4	8	$3.6\cdot10^{-07}$	4	$8.6\cdot10^{-05}$	7	$5.9\cdot10^{-07}$	8	$3.8\cdot10^{-07}$
	8	8	$1.5\cdot10^{-09}$	4	$4.2\cdot10^{-06}$	8	$2.9\cdot10^{-09}$	8	$1.6\cdot10^{-09}$
	16	8	$6.1\cdot10^{-12}$	5	$1.6\cdot10^{-07}$	8	$1.3\cdot10^{-11}$	7	$1.0\cdot10^{-11}$
	32	7	$4.2\cdot10^{-14}$	5	$5.6\cdot10^{-09}$	5	$3.2\cdot10^{-13}$	2	$2.4\cdot10^{-12}$
	64	2	$1.2\cdot10^{-14}$	5	$1.8\cdot10^{-10}$	1	$1.7\cdot10^{-13}$	2	$8.1\cdot10^{-13}$
	128	0	$1.8\cdot10^{-14}$	5	$5.8\cdot10^{-12}$	1	$7.5\cdot10^{-14}$	1	$3.7\cdot10^{-13}$
2,2,2	4	8	$2.5\cdot10^{-07}$	5	$2.8\cdot10^{-06}$	8	$8.3\cdot10^{-07}$	8	$2.5\cdot10^{-07}$
	8	8	$1.1\cdot10^{-09}$	5	$6.6\cdot10^{-08}$	8	$3.8\cdot10^{-09}$	8	$1.1\cdot10^{-09}$
	16	8	$4.5\cdot10^{-12}$	6	$1.3\cdot10^{-09}$	8	$1.7\cdot10^{-11}$	8	$4.5\cdot10^{-12}$
	32	8	$1.5\cdot10^{-14}$	6	$2.2\cdot10^{-11}$	8	$7.1\cdot10^{-14}$	8	$1.8\cdot10^{-14}$
	64	2	$5.9\cdot10^{-15}$	6	$3.7\cdot10^{-13}$	3	$6.7\cdot10^{-15}$	1	$8.4\cdot10^{-15}$
	128	0	$7.0\cdot10^{-15}$	5	$1.1\cdot10^{-14}$	0	$7.1\cdot10^{-15}$	0	$8.0\cdot10^{-15}$
1,1,1,3	4	8	$4.0\cdot10^{-08}$	4	$5.4\cdot10^{-05}$	7	$7.1\cdot10^{-08}$	8	$5.3\cdot10^{-08}$
	8	8	$1.8\cdot10^{-10}$	5	$2.0\cdot10^{-06}$	8	$3.4\cdot10^{-10}$	8	$2.1\cdot10^{-10}$
	16	8	$7.1\cdot10^{-13}$	5	$6.6\cdot10^{-08}$	8	$1.4\cdot10^{-12}$	8	$8.3\cdot10^{-13}$
	32	8	$2.2\cdot10^{-15}$	5	$2.1\cdot10^{-09}$	4	$1.2\cdot10^{-13}$	2	$1.8\cdot10^{-13}$
	64	-	$6.2\cdot10^{-15}$	5	$6.8\cdot10^{-11}$	3	$1.3\cdot10^{-14}$	2	$6.1\cdot10^{-14}$
	128	1	$3.6\cdot10^{-15}$	5	$2.1\cdot10^{-12}$	2	$4.9\cdot10^{-15}$	1	$3.0\cdot10^{-14}$
2,1,3	4	8	$4.5\cdot10^{-09}$	5	$1.7\cdot10^{-06}$	7	$3.9\cdot10^{-08}$	9	$1.1\cdot10^{-08}$
	8	7	$2.7\cdot10^{-11}$	5	$4.3\cdot10^{-08}$	7	$2.2\cdot10^{-10}$	8	$3.1\cdot10^{-11}$
	16	8	$1.2\cdot10^{-13}$	5	$1.3\cdot10^{-09}$	8	$1.0\cdot10^{-12}$	8	$1.2\cdot10^{-13}$
	32	6	$2.4\cdot10^{-15}$	5	$4.2\cdot10^{-11}$	8	$4.2\cdot10^{-15}$	4	$8.1\cdot10^{-15}$
	64	1	$1.4\cdot10^{-15}$	5	$1.3\cdot10^{-12}$	1	$1.8\cdot10^{-15}$	2	$2.4\cdot10^{-15}$
	128	0	$1.4\cdot10^{-15}$	5	$4.0\cdot10^{-14}$	0	$1.7\cdot10^{-15}$	1	$1.4\cdot10^{-15}$
1,1,4	4	8	$4.0\cdot10^{-08}$	4	$5.4\cdot10^{-05}$	7	$7.3\cdot10^{-08}$	8	$5.6\cdot10^{-08}$
	8	8	$1.8\cdot10^{-10}$	5	$2.0\cdot10^{-06}$	8	$3.7\cdot10^{-10}$	8	$2.2\cdot10^{-10}$
	16	8	$7.1\cdot10^{-13}$	5	$6.6\cdot10^{-08}$	8	$1.6\cdot10^{-12}$	8	$8.4\cdot10^{-13}$
	32	6	$8.9\cdot10^{-15}$	5	$2.1\cdot10^{-09}$	4	$1.2\cdot10^{-13}$	2	$1.8\cdot10^{-13}$
	64	0	$8.0\cdot10^{-15}$	5	$6.8\cdot10^{-11}$	3	$1.4\cdot10^{-14}$	2	$6.1\cdot10^{-14}$
2,4	4	-	$3.8\cdot10^{-09}$	5	$1.7\cdot10^{-06}$	8	$8.9\cdot10^{-09}$	7	$4.3\cdot10^{-09}$
	8	8	$1.5\cdot10^{-11}$	6	$3.1\cdot10^{-08}$	7	$7.6\cdot10^{-11}$	8	$1.6\cdot10^{-11}$
	16	8	$6.0\cdot10^{-14}$	6	$5.2\cdot10^{-10}$	7	$4.9\cdot10^{-13}$	8	$6.9\cdot10^{-14}$
	32	6	$9.7\cdot10^{-16}$	6	$8.4\cdot10^{-12}$	8	$2.6\cdot10^{-15}$	3	$6.7\cdot10^{-15}$
	64	-	$3.2\cdot10^{-15}$	6	$1.3\cdot10^{-13}$	0	$3.7\cdot10^{-15}$	1	$3.3\cdot10^{-15}$

Table 10: Convergence rate of problem 1 for k=4 (Double precision)

m	N	COLNEW				BOOTS			
		Mesh data		Cont. aprx.		w/o le		w le	
		CR	GE	CR	GE	CR	GE	CR	GE
1,1,1,1,1,1	8	9	$1.7{\cdot}10^{-12}$	6	$1.6{\cdot}10^{-07}$	8	$1.9{\cdot}10^{-11}$	9	$2.7{\cdot}10^{-12}$
	16	10	$1.8{\cdot}10^{-15}$	6	$2.9{\cdot}10^{-09}$	9	$2.9{\cdot}10^{-14}$	10	$2.5{\cdot}10^{-15}$
	32	10	$1.9{\cdot}10^{-18}$	6	$4.8{\cdot}10^{-11}$	10	$3.5{\cdot}10^{-17}$	10	$2.1{\cdot}10^{-18}$
	64	10	$1.8{\cdot}10^{-21}$	6	$7.7{\cdot}10^{-13}$	10	$3.8{\cdot}10^{-20}$	10	$1.9{\cdot}10^{-21}$
	128	10	$1.8{\cdot}10^{-24}$	6	$1.2{\cdot}10^{-14}$	10	$3.9{\cdot}10^{-23}$	10	$1.8{\cdot}10^{-24}$
	256	10	$1.7{\cdot}10^{-27}$	6	$1.9{\cdot}10^{-16}$	10	$3.9{\cdot}10^{-26}$	10	$1.8{\cdot}10^{-27}$
2,1,1,1,1	8	9	$1.5{\cdot}10^{-12}$	6	$1.6{\cdot}10^{-07}$	9	$4.5{\cdot}10^{-12}$	10	$5.3{\cdot}10^{-12}$
	16	10	$1.7{\cdot}10^{-15}$	6	$2.9{\cdot}10^{-09}$	9	$6.8{\cdot}10^{-15}$	10	$4.3{\cdot}10^{-15}$
	32	10	$1.7{\cdot}10^{-18}$	6	$4.8{\cdot}10^{-11}$	10	$8.1{\cdot}10^{-18}$	10	$3.6{\cdot}10^{-18}$
	64	10	$1.7{\cdot}10^{-21}$	6	$7.7{\cdot}10^{-13}$	10	$8.7{\cdot}10^{-21}$	10	$3.3{\cdot}10^{-21}$
	128	10	$1.7{\cdot}10^{-24}$	6	$1.2{\cdot}10^{-14}$	10	$9.0{\cdot}10^{-24}$	10	$3.1{\cdot}10^{-24}$
	256	10	$1.6{\cdot}10^{-27}$	6	$1.9{\cdot}10^{-16}$	10	$9.0{\cdot}10^{-27}$	10	$3.1{\cdot}10^{-27}$
2,2,2	8	9	$1.4{\cdot}10^{-12}$	7	$1.8{\cdot}10^{-09}$	9	$3.8{\cdot}10^{-12}$	9	$1.4{\cdot}10^{-12}$
	16	10	$1.5{\cdot}10^{-15}$	7	$1.6{\cdot}10^{-11}$	9	$5.3{\cdot}10^{-15}$	10	$1.5{\cdot}10^{-15}$
	32	10	$1.5{\cdot}10^{-18}$	7	$1.4{\cdot}10^{-13}$	10	$6.1{\cdot}10^{-18}$	10	$1.5{\cdot}10^{-18}$
	64	10	$1.5{\cdot}10^{-21}$	7	$1.1{\cdot}10^{-15}$	10	$6.5{\cdot}10^{-21}$	10	$1.5{\cdot}10^{-21}$
	128	10	$1.4{\cdot}10^{-24}$	7	$8.6{\cdot}10^{-18}$	10	$6.6{\cdot}10^{-24}$	10	$1.4{\cdot}10^{-24}$
	256	10	$1.4{\cdot}10^{-27}$	7	$6.1{\cdot}10^{-20}$	10	$6.6{\cdot}10^{-27}$	10	$1.4{\cdot}10^{-27}$
1,1,1,3	8	10	$4.0{\cdot}10^{-13}$	6	$2.6{\cdot}10^{-08}$	10	$1.0{\cdot}10^{-12}$	10	$4.8{\cdot}10^{-13}$
	16	10	$3.9{\cdot}10^{-16}$	6	$4.3{\cdot}10^{-10}$	10	$1.1{\cdot}10^{-15}$	10	$4.2{\cdot}10^{-16}$
	32	10	$3.8{\cdot}10^{-19}$	6	$7.1{\cdot}10^{-12}$	10	$1.2{\cdot}10^{-18}$	10	$3.9{\cdot}10^{-19}$
	64	10	$3.7{\cdot}10^{-22}$	6	$1.1{\cdot}10^{-13}$	10	$1.3{\cdot}10^{-21}$	10	$3.8{\cdot}10^{-22}$
	128	10	$3.6{\cdot}10^{-25}$	6	$1.8{\cdot}10^{-15}$	10	$1.3{\cdot}10^{-24}$	10	$3.7{\cdot}10^{-25}$
	256	10	$3.6{\cdot}10^{-28}$	6	$2.9{\cdot}10^{-17}$	10	$1.3{\cdot}10^{-27}$	10	$3.6{\cdot}10^{-28}$
2,1,3	8	10	$1.4{\cdot}10^{-13}$	7	$3.0{\cdot}10^{-10}$	10	$2.3{\cdot}10^{-13}$	10	$1.4{\cdot}10^{-13}$
	16	10	$1.4{\cdot}10^{-16}$	7	$2.5{\cdot}10^{-12}$	10	$2.4{\cdot}10^{-16}$	10	$1.4{\cdot}10^{-16}$
	32	10	$1.4{\cdot}10^{-19}$	6	$3.3{\cdot}10^{-14}$	10	$2.4{\cdot}10^{-19}$	10	$1.4{\cdot}10^{-19}$
	64	10	$1.4{\cdot}10^{-22}$	6	$6.1{\cdot}10^{-16}$	10	$2.3{\cdot}10^{-22}$	10	$1.4{\cdot}10^{-22}$
	128	10	$1.4{\cdot}10^{-25}$	6	$1.0{\cdot}10^{-17}$	10	$2.3{\cdot}10^{-25}$	10	$1.4{\cdot}10^{-25}$
	256	10	$1.4{\cdot}10^{-28}$	6	$1.7{\cdot}10^{-19}$	10	$2.2{\cdot}10^{-28}$	10	$1.4{\cdot}10^{-28}$
1,1,4	8	10	$4.0{\cdot}10^{-13}$	6	$2.6{\cdot}10^{-08}$	10	$1.6{\cdot}10^{-12}$	10	$7.6{\cdot}10^{-13}$
	16	10	$3.9{\cdot}10^{-16}$	6	$4.3{\cdot}10^{-10}$	10	$1.8{\cdot}10^{-15}$	10	$7.2{\cdot}10^{-16}$
	32	10	$3.8{\cdot}10^{-19}$	6	$7.1{\cdot}10^{-12}$	10	$1.9{\cdot}10^{-18}$	10	$7.0{\cdot}10^{-19}$
	64	10	$3.7{\cdot}10^{-22}$	6	$1.1{\cdot}10^{-13}$	10	$1.9{\cdot}10^{-21}$	10	$6.9{\cdot}10^{-22}$
	128	10	$3.6{\cdot}10^{-25}$	6	$1.8{\cdot}10^{-15}$	10	$1.9{\cdot}10^{-24}$	10	$6.7{\cdot}10^{-25}$
2,4	8	10	$2.8{\cdot}10^{-14}$	7	$3.0{\cdot}10^{-10}$	9	$1.2{\cdot}10^{-13}$	10	$3.2{\cdot}10^{-14}$
	16	10	$2.9{\cdot}10^{-17}$	7	$2.5{\cdot}10^{-12}$	10	$1.3{\cdot}10^{-16}$	10	$3.0{\cdot}10^{-17}$
	32	10	$2.8{\cdot}10^{-20}$	7	$2.0{\cdot}10^{-14}$	10	$1.4{\cdot}10^{-19}$	10	$2.8{\cdot}10^{-20}$
	64	10	$2.7{\cdot}10^{-23}$	7	$1.6{\cdot}10^{-16}$	10	$1.5{\cdot}10^{-22}$	10	$2.7{\cdot}10^{-23}$
	128	10	$2.6{\cdot}10^{-26}$	7	$1.3{\cdot}10^{-18}$	10	$1.5{\cdot}10^{-25}$	10	$2.6{\cdot}10^{-26}$

Table 11: Convergence rate of problem 1 for k=5 (Quadruple precision)

m	N	COLNEW				BOOTS			
		Mesh data		Cont. aprx.		w/o le		w le	
		CR	GE	CR	GE	CR	GE	CR	GE
1,1,1,1,1,1	8	11	$3.0\cdot10^{-15}$	7	$3.3\cdot10^{-09}$	12	$1.5\cdot10^{-13}$	11	$3.4\cdot10^{-14}$
	16	12	$8.1\cdot10^{-19}$	7	$2.6\cdot10^{-11}$	11	$7.2\cdot10^{-17}$	12	$7.9\cdot10^{-18}$
	32	12	$2.0\cdot10^{-22}$	7	$2.0\cdot10^{-13}$	12	$2.4\cdot10^{-20}$	12	$1.4\cdot10^{-21}$
	64	12	$5.0\cdot10^{-26}$	7	$1.6\cdot10^{-15}$	12	$6.6\cdot10^{-24}$	13	$2.1\cdot10^{-25}$
	128	12	$1.2\cdot10^{-29}$	7	$1.2\cdot10^{-17}$	12	$1.8\cdot10^{-27}$	11	$1.2\cdot10^{-28}$
	256	6	$1.6\cdot10^{-31}$	7	$9.3\cdot10^{-20}$	5	$6.2\cdot10^{-29}$	1	$7.3\cdot10^{-29}$
2,1,1,1,1	8	11	$3.2\cdot10^{-15}$	7	$3.3\cdot10^{-09}$	13	$2.5\cdot10^{-14}$	13	$1.2\cdot10^{-14}$
	16	12	$8.5\cdot10^{-19}$	7	$2.6\cdot10^{-11}$	10	$1.8\cdot10^{-17}$	12	$3.2\cdot10^{-18}$
	32	12	$2.1\cdot10^{-22}$	7	$2.0\cdot10^{-13}$	11	$8.3\cdot10^{-21}$	12	$6.5\cdot10^{-22}$
	64	12	$5.2\cdot10^{-26}$	7	$1.6\cdot10^{-15}$	12	$2.7\cdot10^{-24}$	13	$1.1\cdot10^{-25}$
	128	12	$1.3\cdot10^{-29}$	7	$1.2\cdot10^{-17}$	12	$7.5\cdot10^{-28}$	10	$1.2\cdot10^{-28}$
	256	7	$7.2\cdot10^{-32}$	7	$9.3\cdot10^{-20}$	4	$6.2\cdot10^{-29}$	1	$7.3\cdot10^{-29}$
2,2,2	8	12	$2.6\cdot10^{-15}$	8	$3.4\cdot10^{-11}$	12	$3.1\cdot10^{-15}$	12	$2.6\cdot10^{-15}$
	16	12	$6.5\cdot10^{-19}$	8	$1.3\cdot10^{-13}$	12	$8.6\cdot10^{-19}$	12	$6.5\cdot10^{-19}$
	32	12	$1.6\cdot10^{-22}$	8	$5.2\cdot10^{-16}$	12	$2.3\cdot10^{-22}$	12	$1.6\cdot10^{-22}$
	64	12	$3.9\cdot10^{-26}$	8	$2.0\cdot10^{-18}$	12	$5.8\cdot10^{-26}$	12	$3.9\cdot10^{-26}$
	128	12	$9.5\cdot10^{-30}$	8	$7.9\cdot10^{-21}$	12	$1.4\cdot10^{-29}$	12	$1.2\cdot10^{-29}$
	256	8	$3.0\cdot10^{-32}$	8	$3.1\cdot10^{-23}$	0	$1.3\cdot10^{-29}$	0	$1.3\cdot10^{-29}$
1,1,1,3	8	10	$8.0\cdot10^{-16}$	6	$2.0\cdot10^{-10}$	12	$8.9\cdot10^{-16}$	11	$1.4\cdot10^{-15}$
	16	12	$2.4\cdot10^{-19}$	6	$2.8\cdot10^{-12}$	12	$2.4\cdot10^{-19}$	12	$3.6\cdot10^{-19}$
	32	12	$6.3\cdot10^{-23}$	7	$2.9\cdot10^{-14}$	12	$6.3\cdot10^{-23}$	12	$8.2\cdot10^{-23}$
	64	12	$1.6\cdot10^{-26}$	7	$2.6\cdot10^{-16}$	12	$1.6\cdot10^{-26}$	12	$1.8\cdot10^{-26}$
	128	12	$4.0\cdot10^{-30}$	7	$2.1\cdot10^{-18}$	9	$3.9\cdot10^{-29}$	9	$4.2\cdot10^{-29}$
	256	5	$1.6\cdot10^{-31}$	7	$1.7\cdot10^{-20}$	0	$3.8\cdot10^{-29}$	0	$3.8\cdot10^{-29}$
2,1,3	8	11	$1.8\cdot10^{-16}$	6	$8.2\cdot10^{-12}$	12	$1.8\cdot10^{-16}$	12	$1.8\cdot10^{-16}$
	16	12	$4.9\cdot10^{-20}$	7	$8.9\cdot10^{-14}$	12	$5.0\cdot10^{-20}$	12	$4.9\cdot10^{-20}$
	32	12	$1.2\cdot10^{-23}$	7	$8.0\cdot10^{-16}$	12	$1.5\cdot10^{-23}$	12	$1.2\cdot10^{-23}$
	64	12	$3.0\cdot10^{-27}$	7	$6.8\cdot10^{-18}$	12	$4.1\cdot10^{-27}$	12	$3.0\cdot10^{-27}$
	128	12	$7.3\cdot10^{-31}$	7	$5.3\cdot10^{-20}$	10	$4.4\cdot10^{-30}$	10	$3.5\cdot10^{-30}$
	256	6	$8.1\cdot10^{-33}$	7	$4.2\cdot10^{-22}$	0	$3.4\cdot10^{-30}$	0	$3.5\cdot10^{-30}$
1,1,4	8	10	$7.9\cdot10^{-16}$	6	$2.0\cdot10^{-10}$	11	$1.7\cdot10^{-15}$	10	$2.2\cdot10^{-15}$
	16	12	$2.4\cdot10^{-19}$	6	$2.8\cdot10^{-12}$	11	$6.3\cdot10^{-19}$	11	$8.3\cdot10^{-19}$
	32	12	$6.2\cdot10^{-23}$	7	$2.9\cdot10^{-14}$	12	$1.8\cdot10^{-22}$	12	$2.5\cdot10^{-22}$
	64	12	$1.5\cdot10^{-26}$	7	$2.6\cdot10^{-16}$	12	$4.8\cdot10^{-26}$	12	$6.7\cdot10^{-26}$
	128	12	$3.9\cdot10^{-30}$	7	$2.1\cdot10^{-18}$	10	$4.9\cdot10^{-29}$	10	$5.4\cdot10^{-29}$
2,4	8	11	$3.7\cdot10^{-17}$	6	$1.9\cdot10^{-12}$	10	$9.3\cdot10^{-17}$	12	$3.8\cdot10^{-17}$
	16	12	$1.0\cdot10^{-20}$	7	$1.4\cdot10^{-14}$	11	$4.2\cdot10^{-20}$	12	$1.0\cdot10^{-20}$
	32	12	$2.6\cdot10^{-24}$	8	$7.3\cdot10^{-17}$	12	$1.4\cdot10^{-23}$	12	$2.6\cdot10^{-24}$
	64	12	$6.4\cdot10^{-28}$	8	$3.3\cdot10^{-19}$	12	$3.8\cdot10^{-27}$	12	$6.4\cdot10^{-28}$
	128	12	$1.6\cdot10^{-31}$	8	$1.4\cdot10^{-21}$	11	$2.6\cdot10^{-30}$	8	$2.5\cdot10^{-30}$

Table 12: Convergence rate of problem 1 for k=6 (Quadruple precision)

m	N	COLNEW				BOOTS			
		Mesh data		Cont. aprx.		w/o le		w le	
		CR	GE	CR	GE	CR	GE	CR	GE
1,1,1,1,1,1	8	14	$3.7\cdot10^{-18}$	8	$3.4\cdot10^{-11}$	12	$2.7\cdot10^{-15}$	14	$1.3\cdot10^{-16}$
	16	14	$2.1\cdot10^{-22}$	8	$1.3\cdot10^{-13}$	13	$2.6\cdot10^{-19}$	14	$6.5\cdot10^{-21}$
	32	14	$1.3\cdot10^{-26}$	8	$5.3\cdot10^{-16}$	14	$2.0\cdot10^{-23}$	15	$2.5\cdot10^{-25}$
	64	14	$7.6\cdot10^{-31}$	8	$2.3\cdot10^{-18}$	13	$1.9\cdot10^{-27}$	4	$1.6\cdot10^{-26}$
	128	4	$5.2\cdot10^{-32}$	8	$1.0\cdot10^{-20}$	1	$8.7\cdot10^{-28}$	2	$4.5\cdot10^{-27}$
	256	0	$5.2\cdot10^{-32}$	8	$4.4\cdot10^{-23}$	2	$2.5\cdot10^{-28}$	1	$2.2\cdot10^{-27}$
2,1,1,1,1	8	14	$3.7\cdot10^{-18}$	8	$3.4\cdot10^{-11}$	12	$1.3\cdot10^{-15}$	14	$2.7\cdot10^{-16}$
	16	14	$2.1\cdot10^{-22}$	8	$1.3\cdot10^{-13}$	13	$1.3\cdot10^{-19}$	14	$1.3\cdot10^{-20}$
	32	14	$1.3\cdot10^{-26}$	8	$5.3\cdot10^{-16}$	14	$1.1\cdot10^{-23}$	15	$4.7\cdot10^{-25}$
	64	14	$8.5\cdot10^{-31}$	8	$2.3\cdot10^{-18}$	12	$2.2\cdot10^{-27}$	5	$1.6\cdot10^{-26}$
	128	1	$4.7\cdot10^{-31}$	8	$1.0\cdot10^{-20}$	2	$8.7\cdot10^{-28}$	2	$4.5\cdot10^{-27}$
	256	0	$4.6\cdot10^{-31}$	8	$4.4\cdot10^{-23}$	2	$2.5\cdot10^{-28}$	1	$2.2\cdot10^{-27}$
2,2,2	8	14	$1.4\cdot10^{-18}$	9	$2.8\cdot10^{-13}$	13	$1.1\cdot10^{-17}$	14	$1.4\cdot10^{-18}$
	16	14	$7.8\cdot10^{-23}$	9	$5.6\cdot10^{-16}$	14	$6.1\cdot10^{-22}$	14	$7.8\cdot10^{-23}$
	32	14	$4.6\cdot10^{-27}$	9	$1.1\cdot10^{-18}$	14	$2.7\cdot10^{-26}$	14	$4.6\cdot10^{-27}$
	64	14	$2.8\cdot10^{-31}$	9	$2.5\cdot10^{-21}$	11	$1.6\cdot10^{-29}$	8	$1.8\cdot10^{-29}$
	128	4	$1.3\cdot10^{-32}$	9	$5.5\cdot10^{-24}$	0	$1.3\cdot10^{-29}$	0	$1.4\cdot10^{-29}$
	256	0	$1.6\cdot10^{-32}$	9	$1.1\cdot10^{-26}$	0	$1.3\cdot10^{-29}$	0	$1.3\cdot10^{-29}$
1,1,1,3	8	14	$2.3\cdot10^{-18}$	6	$1.7\cdot10^{-11}$	14	$2.3\cdot10^{-18}$	14	$2.8\cdot10^{-18}$
	16	14	$1.5\cdot10^{-22}$	7	$1.0\cdot10^{-13}$	14	$1.5\cdot10^{-22}$	14	$1.7\cdot10^{-22}$
	32	14	$9.4\cdot10^{-27}$	8	$4.9\cdot10^{-16}$	14	$1.1\cdot10^{-26}$	14	$1.0\cdot10^{-26}$
	64	14	$5.7\cdot10^{-31}$	8	$2.1\cdot10^{-18}$	7	$1.2\cdot10^{-28}$	3	$1.3\cdot10^{-27}$
	128	5	$1.5\cdot10^{-32}$	8	$8.8\cdot10^{-21}$	1	$6.1\cdot10^{-29}$	2	$3.5\cdot10^{-28}$
	256	0	$1.3\cdot10^{-32}$	8	$3.5\cdot10^{-23}$	1	$3.8\cdot10^{-29}$	1	$1.7\cdot10^{-28}$
2,1,3	8	14	$2.2\cdot10^{-19}$	7	$3.7\cdot10^{-13}$	14	$2.5\cdot10^{-19}$	14	$2.3\cdot10^{-19}$
	16	14	$1.3\cdot10^{-23}$	8	$1.8\cdot10^{-15}$	14	$1.8\cdot10^{-23}$	14	$1.3\cdot10^{-23}$
	32	14	$8.0\cdot10^{-28}$	8	$7.8\cdot10^{-18}$	14	$1.1\cdot10^{-27}$	14	$8.0\cdot10^{-28}$
	64	14	$4.7\cdot10^{-32}$	8	$3.2\cdot10^{-20}$	8	$4.2\cdot10^{-30}$	6	$1.6\cdot10^{-29}$
	128	3	$5.9\cdot10^{-33}$	8	$1.3\cdot10^{-22}$	0	$3.7\cdot10^{-30}$	2	$5.0\cdot10^{-30}$
	256	0	$6.0\cdot10^{-33}$	8	$5.1\cdot10^{-25}$	0	$3.5\cdot10^{-30}$	0	$4.0\cdot10^{-30}$
1,1,4	8	14	$2.3\cdot10^{-18}$	6	$1.7\cdot10^{-11}$	14	$2.3\cdot10^{-18}$	14	$2.7\cdot10^{-18}$
	16	14	$1.5\cdot10^{-22}$	7	$1.0\cdot10^{-13}$	14	$1.5\cdot10^{-22}$	14	$1.8\cdot10^{-22}$
	32	14	$9.2\cdot10^{-27}$	8	$4.9\cdot10^{-16}$	14	$1.1\cdot10^{-26}$	14	$1.1\cdot10^{-26}$
	64	14	$5.8\cdot10^{-31}$	8	$2.1\cdot10^{-18}$	6	$1.2\cdot10^{-28}$	3	$1.3\cdot10^{-27}$
	128	5	$1.5\cdot10^{-32}$	8	$8.8\cdot10^{-21}$	1	$6.1\cdot10^{-29}$	2	$3.5\cdot10^{-28}$
2,4	8	13	$1.9\cdot10^{-19}$	7	$1.4\cdot10^{-13}$	12	$2.6\cdot10^{-19}$	13	$1.9\cdot10^{-19}$
	16	14	$1.3\cdot10^{-23}$	8	$4.3\cdot10^{-16}$	14	$1.9\cdot10^{-23}$	14	$1.3\cdot10^{-23}$
	32	14	$8.2\cdot10^{-28}$	9	$1.0\cdot10^{-18}$	14	$1.1\cdot10^{-27}$	14	$8.2\cdot10^{-28}$
	64	14	$5.1\cdot10^{-32}$	9	$2.2\cdot10^{-21}$	8	$3.3\cdot10^{-30}$	7	$4.9\cdot10^{-30}$
	128	4	$2.9\cdot10^{-33}$	9	$4.6\cdot10^{-24}$	0	$2.5\cdot10^{-30}$	1	$3.2\cdot10^{-30}$

Table 13: Convergence rate of problem 1 for k=7 (Quadruple precision)

k	N	COLNEW				BOOTS			
		Mesh data		Cont. aprx.		w/o le		w le	
		CR	GE	CR	GE	CR	GE	CR	GE
3	4	6	$1.1{\cdot}10^{-06}$	4	$2.7{\cdot}10^{-05}$	6	$2.9{\cdot}10^{-06}$	6	$2.1{\cdot}10^{-06}$
	8	6	$1.4{\cdot}10^{-08}$	4	$1.9{\cdot}10^{-06}$	6	$4.1{\cdot}10^{-08}$	6	$2.9{\cdot}10^{-08}$
	16	6	$2.1{\cdot}10^{-10}$	4	$1.2{\cdot}10^{-07}$	6	$6.3{\cdot}10^{-10}$	6	$4.3{\cdot}10^{-10}$
	32	6	$3.2{\cdot}10^{-12}$	4	$7.8{\cdot}10^{-09}$	6	$9.9{\cdot}10^{-12}$	6	$6.7{\cdot}10^{-12}$
	64	6	$4.9{\cdot}10^{-14}$	4	$4.9{\cdot}10^{-10}$	6	$1.5{\cdot}10^{-13}$	6	$1.0{\cdot}10^{-13}$
	128	6	$7.7{\cdot}10^{-16}$	4	$3.1{\cdot}10^{-11}$	6	$2.4{\cdot}10^{-15}$	6	$1.6{\cdot}10^{-15}$
4	4	7	$5.6{\cdot}10^{-09}$	5	$5.0{\cdot}10^{-07}$	7	$1.1{\cdot}10^{-08}$	7	$1.1{\cdot}10^{-08}$
	8	8	$2.4{\cdot}10^{-11}$	5	$1.7{\cdot}10^{-08}$	8	$5.8{\cdot}10^{-11}$	8	$5.8{\cdot}10^{-11}$
	16	8	$9.4{\cdot}10^{-14}$	5	$5.6{\cdot}10^{-10}$	7	$3.4{\cdot}10^{-13}$	7	$3.4{\cdot}10^{-13}$
	32	8	$3.7{\cdot}10^{-16}$	5	$1.8{\cdot}10^{-11}$	7	$2.2{\cdot}10^{-15}$	7	$2.3{\cdot}10^{-15}$
	64	8	$1.4{\cdot}10^{-18}$	5	$5.6{\cdot}10^{-13}$	7	$1.6{\cdot}10^{-17}$	7	$1.6{\cdot}10^{-17}$
	128	8	$5.6{\cdot}10^{-21}$	5	$1.7{\cdot}10^{-14}$	7	$1.2{\cdot}10^{-19}$	7	$1.2{\cdot}10^{-19}$
5	4	9	$1.1{\cdot}10^{-11}$	6	$7.7{\cdot}10^{-09}$	9	$2.3{\cdot}10^{-11}$	9	$1.7{\cdot}10^{-11}$
	8	10	$1.1{\cdot}10^{-14}$	6	$1.3{\cdot}10^{-10}$	10	$2.7{\cdot}10^{-14}$	10	$1.9{\cdot}10^{-14}$
	16	10	$1.0{\cdot}10^{-17}$	6	$2.1{\cdot}10^{-12}$	10	$3.0{\cdot}10^{-17}$	10	$2.1{\cdot}10^{-17}$
	32	10	$1.0{\cdot}10^{-20}$	6	$3.4{\cdot}10^{-14}$	10	$3.4{\cdot}10^{-20}$	10	$2.4{\cdot}10^{-20}$
	64	10	$9.8{\cdot}10^{-24}$	6	$5.4{\cdot}10^{-16}$	10	$4.1{\cdot}10^{-23}$	10	$3.1{\cdot}10^{-23}$
	128	10	$9.6{\cdot}10^{-27}$	6	$8.4{\cdot}10^{-18}$	10	$5.5{\cdot}10^{-26}$	9	$4.5{\cdot}10^{-26}$
6	4	13	$2.9{\cdot}10^{-15}$	7	$9.1{\cdot}10^{-11}$	12	$2.4{\cdot}10^{-14}$	13	$9.6{\cdot}10^{-15}$
	8	13	$3.4{\cdot}10^{-19}$	7	$7.7{\cdot}10^{-13}$	12	$6.8{\cdot}10^{-18}$	13	$1.6{\cdot}10^{-18}$
	16	12	$6.3{\cdot}10^{-23}$	7	$6.3{\cdot}10^{-15}$	12	$1.8{\cdot}10^{-21}$	12	$3.1{\cdot}10^{-22}$
	32	12	$1.4{\cdot}10^{-26}$	7	$5.0{\cdot}10^{-17}$	12	$4.6{\cdot}10^{-25}$	12	$6.9{\cdot}10^{-26}$
	64	12	$3.4{\cdot}10^{-30}$	7	$4.0{\cdot}10^{-19}$	12	$1.2{\cdot}10^{-28}$	12	$1.7{\cdot}10^{-29}$
	128	8	$1.1{\cdot}10^{-32}$	7	$3.0{\cdot}10^{-21}$	5	$3.4{\cdot}10^{-30}$	2	$3.7{\cdot}10^{-30}$
7	4	16	$1.7{\cdot}10^{-18}$	8	$9.4{\cdot}10^{-13}$	13	$3.6{\cdot}10^{-17}$	14	$1.2{\cdot}10^{-17}$
	8	14	$1.1{\cdot}10^{-22}$	8	$4.0{\cdot}10^{-15}$	14	$3.1{\cdot}10^{-21}$	15	$4.8{\cdot}10^{-22}$
	16	14	$9.5{\cdot}10^{-27}$	8	$1.6{\cdot}10^{-17}$	14	$2.3{\cdot}10^{-25}$	15	$1.7{\cdot}10^{-26}$
	32	14	$5.1{\cdot}10^{-31}$	8	$6.4{\cdot}10^{-20}$	11	$1.1{\cdot}10^{-28}$	4	$1.4{\cdot}10^{-27}$
	64	2	$1.3{\cdot}10^{-31}$	8	$2.5{\cdot}10^{-22}$	1	$5.9{\cdot}10^{-29}$	2	$4.8{\cdot}10^{-28}$
	128	0	$1.2{\cdot}10^{-31}$	8	$1.0{\cdot}10^{-24}$	1	$3.0{\cdot}10^{-29}$	2	$1.6{\cdot}10^{-28}$

Table 14: Convergence rate of problem 2 (Quadruple precision)

m	N	COLNEW	BOOTS w/o le	BOOTS w le
1,1,1,1,1,1	4	0.104	0.001	0.002
	8	0.181	0.002	0.004
	16	0.338	0.004	0.007
	32	0.575	0.007	0.014
	64	1.158	0.014	0.028
	128	2.321	0.028	0.055
2,1,1,1,1	4	0.095	0.001	0.002
	8	0.187	0.002	0.003
	16	0.324	0.004	0.006
	32	0.607	0.007	0.030
	64	1.195	0.013	0.025
	128	2.177	0.026	0.048
2,2,2	4	0.046	0.000	0.001
	8	0.091	0.001	0.002
	16	0.143	0.001	0.002
	32	0.237	0.002	0.005
	64	0.470	0.003	0.010
	128	0.928	0.007	0.019
1,1,1,3	4	0.047	0.001	0.002
	8	0.086	0.002	0.003
	16	0.166	0.003	0.006
	32	0.312	0.006	0.015
	64	0.619	0.013	0.022
	128	1.240	0.024	0.043
2,1,3	4	0.040	0.001	0.001
	8	0.063	0.001	0.003
	16	0.125	0.003	0.005
	32	0.258	0.006	0.010
	64	0.471	0.012	0.021
	128	0.938	0.023	0.041
1,1,4	4	0.038	0.001	0.002
	8	0.065	0.002	0.003
	16	0.143	0.003	0.006
	32	0.243	0.007	0.012
	64	0.488	0.013	0.022
2,4	4	0.031	0.000	0.001
	8	0.048	0.001	0.001
	16	0.096	0.001	0.002
	32	0.200	0.002	0.005
	64	0.364	0.004	0.009

Table 15: Execution time for convergence rate test of problem 1 for k=4

m	Err Tol	N	Execution Time			Global Error in approximation		
			Col-new	Boots		Colnew	Boots	
				w/o le	w le		w/o le	w le
1,1,1,1,1,1	10^{-4}	384	1.352	0.000	0.000	$5.7\cdot10^{-5}$	$5.7\cdot10^{-5}$	$5.7\cdot10^{-5}$
	10^{-6}	3072	9.500	0.031	0.039	$8.8\cdot10^{-7}$	$8.8\cdot10^{-7}$	$8.8\cdot10^{-7}$

Table 16: Adaptive mesh results of problem 1 for k=1 (Double precision)

m	Err Tol	N	Execution Time			Global Error in approximation		
			Col-new	Boots		Colnew	Boots	
				w/o le	w le		w/o le	w le
1,1,1,1,1,1	10^{-4}	24	0.156	0.000	0.000	$4.2\cdot10^{-5}$	$3.9\cdot10^{-6}$	$3.7\cdot10^{-6}$
	10^{-6}	96	0.492	0.000	0.008	$7.3\cdot10^{-7}$	$1.6\cdot10^{-8}$	$1.3\cdot10^{-8}$
	10^{-8}	384	1.859	0.000	0.008	$1.1\cdot10^{-8}$	$6.5\cdot10^{-11}$	$4.9\cdot10^{-11}$
	10^{-10}	3072	13.43	0.016	0.102	$2.3\cdot10^{-11}$	$3.1\cdot10^{-14}$	$3.2\cdot10^{-14}$
2,1,1,1,1	10^{-4}	24	0.125	0.000	0.000	$3.4\cdot10^{-5}$	$3.9\cdot10^{-6}$	$3.3\cdot10^{-6}$
	10^{-6}	96	0.680	0.000	0.016	$5.2\cdot10^{-7}$	$1.5\cdot10^{-8}$	$1.3\cdot10^{-8}$
	10^{-8}	384	1.594	0.000	0.008	$8.1\cdot10^{-9}$	$6.0\cdot10^{-11}$	$5.0\cdot10^{-11}$
	10^{-10}	3072	11.87	0.016	0.094	$1.6\cdot10^{-11}$	$3.1\cdot10^{-14}$	$3.3\cdot10^{-14}$
2,2,2	10^{-4}	12	0.047	0.000	0.000	$1.9\cdot10^{-5}$	$1.8\cdot10^{-5}$	$1.8\cdot10^{-5}$
	10^{-6}	48	0.180	0.000	0.000	$8.9\cdot10^{-8}$	$7.3\cdot10^{-8}$	$7.3\cdot10^{-8}$
	10^{-8}	96	0.359	0.000	0.000	$5.8\cdot10^{-9}$	$4.6\cdot10^{-9}$	$4.6\cdot10^{-9}$
	10^{-10}	384	1.219	0.000	0.000	$2.3\cdot10^{-11}$	$1.8\cdot10^{-11}$	$1.8\cdot10^{-11}$

Table 17: Adaptive mesh results of problem 1 for k=2 (Double precision)

m	Err Tol	N	Execution Time			Global Error in approximation		
			Col-new	Boots		Colnew	Boots	
				w/o le	w le		w/o le	w le
1,1,1,1,1,1	10^{-4}	6	0.086	0.000	0.008	$1.7\cdot10^{-4}$	$4.0\cdot10^{-5}$	$3.2\cdot10^{-6}$
	10^{-6}	24	0.234	0.000	0.000	$1.1\cdot10^{-6}$	$1.2\cdot10^{-8}$	$4.6\cdot10^{-10}$
	10^{-8}	96	0.781	0.000	0.016	$5.2\cdot10^{-9}$	$3.2\cdot10^{-12}$	$9.8\cdot10^{-14}$
	10^{-10}	192	1.484	0.008	0.031	$3.3\cdot10^{-10}$	$5.7\cdot10^{-14}$	$1.2\cdot10^{-14}$
2,1,1,1,1	10^{-4}	6	0.078	0.000	0.000	$1.6\cdot10^{-4}$	$9.7\cdot10^{-6}$	$1.3\cdot10^{-5}$
	10^{-6}	24	0.258	0.000	0.000	$1.1\cdot10^{-6}$	$3.1\cdot10^{-9}$	$4.1\cdot10^{-9}$
	10^{-8}	96	0.859	0.000	0.016	$5.2\cdot10^{-9}$	$8.3\cdot10^{-13}$	$1.1\cdot10^{-12}$
	10^{-10}	192	1.188	0.016	0.031	$3.3\cdot10^{-10}$	$2.9\cdot10^{-14}$	$3.4\cdot10^{-14}$
2,2,2	10^{-4}	6	0.039	0.000	0.000	$4.3\cdot10^{-6}$	$1.5\cdot10^{-6}$	$1.5\cdot10^{-6}$
	10^{-6}	12	0.070	0.008	0.000	$1.9\cdot10^{-7}$	$2.7\cdot10^{-8}$	$2.6\cdot10^{-8}$
	10^{-8}	24	0.125	0.000	0.000	$7.0\cdot10^{-9}$	$4.8\cdot10^{-10}$	$4.1\cdot10^{-10}$
	10^{-10}	96	0.398	0.000	0.008	$7.8\cdot10^{-12}$	$1.3\cdot10^{-13}$	$1.0\cdot10^{-13}$
1,1,1,3	10^{-4}	6	0.039	0.000	0.000	$1.7\cdot10^{-4}$	$4.2\cdot10^{-6}$	$1.2\cdot10^{-6}$
	10^{-6}	24	0.141	0.000	0.008	$9.3\cdot10^{-7}$	$1.1\cdot10^{-9}$	$2.7\cdot10^{-10}$
	10^{-8}	96	0.500	0.000	0.016	$3.9\cdot10^{-9}$	$2.8\cdot10^{-13}$	$6.7\cdot10^{-14}$
	10^{-10}	192	0.977	0.008	0.031	$2.4\cdot10^{-10}$	$4.9\cdot10^{-15}$	$2.2\cdot10^{-15}$
2,1,3	10^{-4}	6	0.031	0.000	0.000	$9.0\cdot10^{-6}$	$1.2\cdot10^{-6}$	$6.0\cdot10^{-8}$
	10^{-6}	12	0.062	0.000	0.008	$6.2\cdot10^{-7}$	$2.0\cdot10^{-8}$	$8.8\cdot10^{-10}$
	10^{-8}	48	0.219	0.000	0.008	$2.6\cdot10^{-9}$	$5.2\cdot10^{-12}$	$2.1\cdot10^{-13}$
	10^{-10}	192	0.781	0.008	0.023	$1.0\cdot10^{-11}$	$1.8\cdot10^{-15}$	$1.3\cdot10^{-15}$

Table 18: Adaptive mesh results of problem 1 for k=3 (Double precision)

m	Err Tol	N	Execution Time Col-new	Execution Time Boots w/o le	Execution Time Boots w le	Global Error in approximation Colnew	Global Error Boots w/o le	Global Error Boots w le
1,1,1,1,1,1	10^{-4}	6	0.125	0.000	0.000	$1.6\cdot10^{-5}$	$8.8\cdot10^{-8}$	$2.0\cdot10^{-8}$
	10^{-6}	12	0.219	0.000	0.008	$6.4\cdot10^{-7}$	$3.5\cdot10^{-10}$	$8.0\cdot10^{-11}$
	10^{-8}	24	0.359	0.008	0.016	$2.3\cdot10^{-8}$	$1.5\cdot10^{-12}$	$3.4\cdot10^{-12}$
	10^{-10}	96	1.203	0.023	0.047	$2.4\cdot10^{-11}$	$2.5\cdot10^{-13}$	$7.9\cdot10^{-13}$
2,1,1,1,1	10^{-4}	6	0.109	0.000	0.000	$1.5\cdot10^{-5}$	$2.7\cdot10^{-8}$	$1.6\cdot10^{-8}$
	10^{-6}	12	0.180	0.008	0.000	$6.3\cdot10^{-7}$	$1.2\cdot10^{-10}$	$6.5\cdot10^{-11}$
	10^{-8}	24	0.281	0.008	0.008	$2.3\cdot10^{-8}$	$6.7\cdot10^{-13}$	$3.3\cdot10^{-12}$
	10^{-10}	96	0.938	0.023	0.039	$2.4\cdot10^{-11}$	$2.5\cdot10^{-13}$	$8.0\cdot10^{-13}$
2,2,2	10^{-4}	6	0.055	0.008	0.000	$3.2\cdot10^{-7}$	$3.6\cdot10^{-8}$	$1.1\cdot10^{-8}$
	10^{-6}	6	0.055	0.000	0.008	$3.2\cdot10^{-7}$	$3.6\cdot10^{-8}$	$1.1\cdot10^{-8}$
	10^{-8}	12	0.094	0.000	0.000	$6.7\cdot10^{-9}$	$1.6\cdot10^{-10}$	$4.4\cdot10^{-11}$
	10^{-10}	24	0.156	0.000	0.000	$1.2\cdot10^{-10}$	$7.1\cdot10^{-13}$	$1.9\cdot10^{-13}$
1,1,1,3	10^{-4}	6	0.055	0.000	0.008	$7.9\cdot10^{-6}$	$3.3\cdot10^{-9}$	$2.2\cdot10^{-9}$
	10^{-6}	12	0.102	0.000	0.008	$2.7\cdot10^{-7}$	$1.4\cdot10^{-11}$	$7.8\cdot10^{-12}$
	10^{-8}	24	0.195	0.008	0.008	$8.9\cdot10^{-9}$	$1.5\cdot10^{-13}$	$2.7\cdot10^{-13}$
	10^{-10}	48	0.375	0.008	0.016	$2.8\cdot10^{-10}$	$1.2\cdot10^{-13}$	$1.3\cdot10^{-13}$
2,1,3	10^{-4}	6	0.039	0.008	0.000	$1.8\cdot10^{-7}$	$2.0\cdot10^{-9}$	$3.3\cdot10^{-10}$
	10^{-6}	6	0.047	0.000	0.008	$1.8\cdot10^{-7}$	$2.0\cdot10^{-9}$	$3.3\cdot10^{-10}$
	10^{-8}	12	0.094	0.008	0.008	$5.6\cdot10^{-9}$	$9.8\cdot10^{-12}$	$1.2\cdot10^{-12}$
	10^{-10}	48	0.281	0.008	0.016	$5.5\cdot10^{-12}$	$7.3\cdot10^{-15}$	$4.2\cdot10^{-15}$
1,1,4	10^{-4}	6	0.039	0.000	0.000	$7.9\cdot10^{-6}$	$3.5\cdot10^{-9}$	$2.3\cdot10^{-9}$
	10^{-6}	12	0.086	0.000	0.008	$2.7\cdot10^{-7}$	$1.5\cdot10^{-11}$	$7.8\cdot10^{-12}$
	10^{-8}	24	0.156	0.008	0.008	$8.9\cdot10^{-9}$	$1.5\cdot10^{-13}$	$2.7\cdot10^{-13}$
	10^{-10}	48	0.359	0.016	0.016	$2.8\cdot10^{-10}$	$1.2\cdot10^{-13}$	$1.3\cdot10^{-13}$
4,2	10^{-4}	6	0.031	0.000	0.000	$1.6\cdot10^{-7}$	$5.6\cdot10^{-10}$	$1.6\cdot10^{-10}$
	10^{-6}	6	0.031	0.000	0.000	$1.6\cdot10^{-7}$	$5.6\cdot10^{-10}$	$1.6\cdot10^{-10}$
	10^{-8}	12	0.062	0.000	0.008	$2.8\cdot10^{-9}$	$4.1\cdot10^{-12}$	$6.1\cdot10^{-13}$
	10^{-10}	24	0.141	0.000	0.008	$4.7\cdot10^{-11}$	$2.2\cdot10^{-14}$	$9.7\cdot10^{-15}$

Table 19: Adaptive mesh results of problem 1 for k=4 (Double precision)

m	Err Tol	N	Execution Time Col-new	Boots w/o le	Boots w le	Global Error in approximation Colnew	Boots w/o le	Boots w le
1,1,1,1,1,1	10^{-4}	6	0.195	0.008	0.008	$8.2 \cdot 10^{-7}$	$2.4 \cdot 10^{-10}$	$1.9 \cdot 10^{-10}$
	10^{-6}	6	0.203	0.008	0.008	$8.2 \cdot 10^{-7}$	$2.4 \cdot 10^{-10}$	$1.9 \cdot 10^{-10}$
	10^{-8}	12	0.320	0.008	0.016	$1.5 \cdot 10^{-8}$	$1.3 \cdot 10^{-11}$	$1.1 \cdot 10^{-10}$
	10^{-10}	24	0.508	0.016	0.031	$2.6 \cdot 10^{-10}$	$6.7 \cdot 10^{-12}$	$5.8 \cdot 10^{-11}$
2,1,1,1,1	10^{-4}	6	0.156	0.000	0.008	$8.2 \cdot 10^{-7}$	$6.0 \cdot 10^{-11}$	$1.9 \cdot 10^{-10}$
	10^{-6}	6	0.172	0.000	0.008	$8.2 \cdot 10^{-7}$	$6.0 \cdot 10^{-11}$	$1.9 \cdot 10^{-10}$
	10^{-8}	12	0.250	0.008	0.008	$1.5 \cdot 10^{-8}$	$1.3 \cdot 10^{-11}$	$1.1 \cdot 10^{-10}$
	10^{-10}	24	0.422	0.016	0.023	$2.6 \cdot 10^{-10}$	$6.7 \cdot 10^{-12}$	$5.8 \cdot 10^{-11}$
2,2,2	10^{-4}	6	0.070	0.008	0.008	$1.2 \cdot 10^{-8}$	$5.2 \cdot 10^{-11}$	$2.3 \cdot 10^{-11}$
	10^{-6}	6	0.078	0.000	0.000	$1.2 \cdot 10^{-8}$	$5.2 \cdot 10^{-11}$	$2.3 \cdot 10^{-11}$
	10^{-8}	6	0.078	0.008	0.008	$1.2 \cdot 10^{-8}$	$5.2 \cdot 10^{-11}$	$2.3 \cdot 10^{-11}$
	10^{-10}	12	0.172	0.000	0.008	$1.2 \cdot 10^{-10}$	$1.6 \cdot 10^{-13}$	$4.7 \cdot 10^{-13}$
1,1,1,3	10^{-4}	6	0.078	0.000	0.008	$1.4 \cdot 10^{-7}$	$1.7 \cdot 10^{-11}$	$2.5 \cdot 10^{-11}$
	10^{-6}	6	0.070	0.008	0.008	$1.4 \cdot 10^{-7}$	$1.7 \cdot 10^{-11}$	$2.5 \cdot 10^{-11}$
	10^{-8}	12	0.141	0.008	0.008	$2.4 \cdot 10^{-9}$	$1.1 \cdot 10^{-12}$	$1.3 \cdot 10^{-11}$
	10^{-10}	24	0.273	0.016	0.016	$3.9 \cdot 10^{-11}$	$5.4 \cdot 10^{-13}$	$4.7 \cdot 10^{-12}$
2,1,3	10^{-4}	6	0.062	0.000	0.008	$2.2 \cdot 10^{-9}$	$3.9 \cdot 10^{-12}$	$2.2 \cdot 10^{-12}$
	10^{-6}	6	0.055	0.008	0.008	$2.2 \cdot 10^{-9}$	$3.9 \cdot 10^{-12}$	$2.2 \cdot 10^{-12}$
	10^{-8}	6	0.055	0.008	0.008	$2.2 \cdot 10^{-9}$	$3.9 \cdot 10^{-12}$	$2.2 \cdot 10^{-12}$
	10^{-10}	12	0.109	0.008	0.008	$1.8 \cdot 10^{-11}$	$9.6 \cdot 10^{-14}$	$2.0 \cdot 10^{-13}$
1,1,4	10^{-4}	6	0.047	0.008	0.000	$1.4 \cdot 10^{-7}$	$2.6 \cdot 10^{-11}$	$2.5 \cdot 10^{-11}$
	10^{-6}	6	0.062	0.008	0.000	$1.4 \cdot 10^{-7}$	$2.6 \cdot 10^{-11}$	$2.5 \cdot 10^{-11}$
	10^{-8}	12	0.109	0.000	0.008	$2.4 \cdot 10^{-9}$	$1.1 \cdot 10^{-12}$	$1.3 \cdot 10^{-11}$
	10^{-10}	24	0.211	0.016	0.016	$3.9 \cdot 10^{-11}$	$5.4 \cdot 10^{-13}$	$4.7 \cdot 10^{-12}$
4,2	10^{-4}	6	0.039	0.000	0.008	$2.2 \cdot 10^{-9}$	$1.8 \cdot 10^{-12}$	$8.9 \cdot 10^{-13}$
	10^{-6}	6	0.047	0.000	0.000	$2.2 \cdot 10^{-9}$	$1.8 \cdot 10^{-12}$	$8.9 \cdot 10^{-13}$
	10^{-8}	6	0.047	0.000	0.000	$2.2 \cdot 10^{-9}$	$1.8 \cdot 10^{-12}$	$8.9 \cdot 10^{-13}$
	10^{-10}	12	0.086	0.008	0.008	$1.8 \cdot 10^{-11}$	$9.6 \cdot 10^{-14}$	$2.0 \cdot 10^{-13}$

Table 20: Adaptive mesh results of problem 1 for k=5 (Double precision)

m	Err Tol	N	Execution Time Col-new	Boots w/o le	Boots w le	Global Error in approximation Colnew	Boots w/o le	Boots w le
1,1,1,1,1,1	10^{-4}	6	2.633	0.062	0.086	$2.4\cdot10^{-8}$	$2.9\cdot10^{-12}$	$8.9\cdot10^{-13}$
	10^{-6}	6	2.961	0.062	0.094	$2.4\cdot10^{-8}$	$2.9\cdot10^{-12}$	$8.9\cdot10^{-13}$
	10^{-8}	12	4.883	0.133	0.180	$1.9\cdot10^{-10}$	$1.9\cdot10^{-15}$	$2.7\cdot10^{-16}$
	10^{-10}	12	4.719	0.133	0.180	$1.9\cdot10^{-10}$	$1.9\cdot10^{-15}$	$2.7\cdot10^{-16}$
2,1,1,1,1	10^{-4}	6	2.672	0.078	0.094	$2.4\cdot10^{-8}$	$7.6\cdot10^{-13}$	$3.5\cdot10^{-13}$
	10^{-6}	6	2.398	0.062	0.094	$2.4\cdot10^{-8}$	$7.6\cdot10^{-13}$	$3.5\cdot10^{-13}$
	10^{-8}	12	3.656	0.133	0.195	$1.9\cdot10^{-10}$	$3.2\cdot10^{-16}$	$9.3\cdot10^{-17}$
	10^{-10}	12	3.945	0.141	0.188	$1.9\cdot10^{-10}$	$3.2\cdot10^{-16}$	$9.3\cdot10^{-17}$
2,2,2	10^{-4}	6	0.805	0.039	0.062	$3.2\cdot10^{-10}$	$9.1\cdot10^{-14}$	$7.9\cdot10^{-14}$
	10^{-6}	6	0.805	0.039	0.062	$3.2\cdot10^{-10}$	$9.1\cdot10^{-14}$	$7.9\cdot10^{-14}$
	10^{-8}	6	0.797	0.039	0.062	$3.2\cdot10^{-10}$	$9.1\cdot10^{-14}$	$7.9\cdot10^{-14}$
	10^{-10}	12	1.445	0.078	0.125	$1.3\cdot10^{-12}$	$2.6\cdot10^{-17}$	$2.0\cdot10^{-17}$
1,1,1,3	10^{-4}	6	0.930	0.078	0.102	$9.5\cdot10^{-10}$	$2.6\cdot10^{-14}$	$3.6\cdot10^{-14}$
	10^{-6}	6	0.875	0.070	0.094	$9.5\cdot10^{-10}$	$2.6\cdot10^{-14}$	$3.6\cdot10^{-14}$
	10^{-8}	6	0.875	0.070	0.094	$9.5\cdot10^{-10}$	$2.6\cdot10^{-14}$	$3.6\cdot10^{-14}$
	10^{-10}	12	2.016	0.141	0.195	$1.7\cdot10^{-11}$	$7.3\cdot10^{-18}$	$1.1\cdot10^{-17}$
2,1,3	10^{-4}	6	0.766	0.188	0.094	$4.8\cdot10^{-11}$	$5.0\cdot10^{-15}$	$5.8\cdot10^{-15}$
	10^{-6}	6	0.922	0.289	0.219	$4.8\cdot10^{-11}$	$5.0\cdot10^{-15}$	$5.8\cdot10^{-15}$
	10^{-8}	6	0.586	0.078	0.102	$4.8\cdot10^{-11}$	$5.0\cdot10^{-15}$	$5.8\cdot10^{-15}$
	10^{-10}	6	0.703	0.070	0.102	$4.8\cdot10^{-11}$	$5.0\cdot10^{-15}$	$5.8\cdot10^{-15}$
1,1,4	10^{-4}	6	0.484	0.078	0.102	$9.6\cdot10^{-10}$	$3.8\cdot10^{-14}$	$4.9\cdot10^{-14}$
	10^{-6}	6	0.594	0.078	0.102	$9.6\cdot10^{-10}$	$3.8\cdot10^{-14}$	$4.9\cdot10^{-14}$
	10^{-8}	6	0.625	0.070	0.102	$9.6\cdot10^{-10}$	$3.8\cdot10^{-14}$	$4.9\cdot10^{-14}$
	10^{-10}	12	1.258	0.148	0.219	$1.7\cdot10^{-11}$	$1.7\cdot10^{-17}$	$2.3\cdot10^{-17}$
4,2	10^{-4}	6	0.336	0.047	0.062	$1.2\cdot10^{-11}$	$1.9\cdot10^{-15}$	$1.1\cdot10^{-15}$
	10^{-6}	6	0.414	0.047	0.070	$1.2\cdot10^{-11}$	$1.9\cdot10^{-15}$	$1.1\cdot10^{-15}$
	10^{-8}	6	0.398	0.047	0.070	$1.2\cdot10^{-11}$	$1.9\cdot10^{-15}$	$1.1\cdot10^{-15}$
	10^{-10}	6	0.398	0.047	0.070	$1.2\cdot10^{-11}$	$1.9\cdot10^{-15}$	$1.1\cdot10^{-15}$

Table 21: Adaptive mesh results of problem 1 for k=6 (Quadruple precision)

m	Err Tol	N	Execution Time Col-new	Boots w/o le	Boots w le	Global Error in approximation Colnew	Boots w/o le	Boots w le
1,1,1,1,1,1	10^{-4}	6	3.758	0.117	0.141	$3.1 \cdot 10^{-10}$	$1.1 \cdot 10^{-13}$	$7.8 \cdot 10^{-15}$
	10^{-6}	6	5.078	0.133	0.156	$3.1 \cdot 10^{-10}$	$1.1 \cdot 10^{-13}$	$7.8 \cdot 10^{-15}$
	10^{-8}	6	5.141	0.117	0.148	$3.1 \cdot 10^{-10}$	$1.1 \cdot 10^{-13}$	$7.8 \cdot 10^{-15}$
	10^{-10}	12	6.977	0.227	0.305	$1.3 \cdot 10^{-12}$	$1.3 \cdot 10^{-17}$	$4.2 \cdot 10^{-19}$
2,1,1,1,1	10^{-4}	6	3.883	0.203	0.148	$3.1 \cdot 10^{-10}$	$5.0 \cdot 10^{-14}$	$1.5 \cdot 10^{-14}$
	10^{-6}	6	5.016	0.117	0.359	$3.1 \cdot 10^{-10}$	$5.0 \cdot 10^{-14}$	$1.5 \cdot 10^{-14}$
	10^{-8}	6	5.227	0.117	0.156	$3.1 \cdot 10^{-10}$	$5.0 \cdot 10^{-14}$	$1.5 \cdot 10^{-14}$
	10^{-10}	12	6.359	0.227	0.312	$1.3 \cdot 10^{-12}$	$6.3 \cdot 10^{-18}$	$8.2 \cdot 10^{-19}$
2,2,2	10^{-4}	6	0.953	0.078	0.102	$3.4 \cdot 10^{-12}$	$6.0 \cdot 10^{-16}$	$8.2 \cdot 10^{-17}$
	10^{-6}	6	1.094	0.078	0.109	$3.4 \cdot 10^{-12}$	$6.0 \cdot 10^{-16}$	$8.2 \cdot 10^{-17}$
	10^{-8}	6	1.109	0.070	0.195	$3.4 \cdot 10^{-12}$	$6.0 \cdot 10^{-16}$	$8.2 \cdot 10^{-17}$
	10^{-10}	6	1.641	0.156	0.109	$3.4 \cdot 10^{-12}$	$6.0 \cdot 10^{-16}$	$8.2 \cdot 10^{-17}$
1,1,1,3	10^{-4}	6	1.258	0.125	0.188	$1.2 \cdot 10^{-10}$	$1.2 \cdot 10^{-16}$	$1.5 \cdot 10^{-16}$
	10^{-6}	6	1.820	0.125	0.242	$1.2 \cdot 10^{-10}$	$1.2 \cdot 10^{-16}$	$1.5 \cdot 10^{-16}$
	10^{-8}	6	1.688	0.117	0.164	$1.2 \cdot 10^{-10}$	$1.2 \cdot 10^{-16}$	$1.5 \cdot 10^{-16}$
	10^{-10}	6	1.719	0.320	0.305	$1.2 \cdot 10^{-10}$	$1.2 \cdot 10^{-16}$	$1.5 \cdot 10^{-16}$
2,1,3	10^{-4}	6	0.852	0.125	0.164	$3.2 \cdot 10^{-12}$	$1.3 \cdot 10^{-17}$	$1.3 \cdot 10^{-17}$
	10^{-6}	6	0.797	0.125	0.164	$3.2 \cdot 10^{-12}$	$1.3 \cdot 10^{-17}$	$1.3 \cdot 10^{-17}$
	10^{-8}	6	0.836	0.133	0.180	$3.2 \cdot 10^{-12}$	$1.3 \cdot 10^{-17}$	$1.3 \cdot 10^{-17}$
	10^{-10}	6	0.938	0.125	0.156	$3.2 \cdot 10^{-12}$	$1.3 \cdot 10^{-17}$	$1.3 \cdot 10^{-17}$
1,1,4	10^{-4}	6	0.656	0.125	0.172	$1.2 \cdot 10^{-10}$	$1.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-16}$
	10^{-6}	6	0.812	0.133	0.172	$1.2 \cdot 10^{-10}$	$1.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-16}$
	10^{-8}	6	0.828	0.125	0.172	$1.2 \cdot 10^{-10}$	$1.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-16}$
	10^{-10}	6	0.812	0.125	0.172	$1.2 \cdot 10^{-10}$	$1.2 \cdot 10^{-16}$	$1.4 \cdot 10^{-16}$
4,2	10^{-4}	6	0.438	0.086	0.125	$1.4 \cdot 10^{-12}$	$1.1 \cdot 10^{-17}$	$9.0 \cdot 10^{-18}$
	10^{-6}	6	0.539	0.086	0.125	$1.4 \cdot 10^{-12}$	$1.1 \cdot 10^{-17}$	$9.0 \cdot 10^{-18}$
	10^{-8}	6	0.539	0.094	0.125	$1.4 \cdot 10^{-12}$	$1.1 \cdot 10^{-17}$	$9.0 \cdot 10^{-18}$
	10^{-10}	6	0.531	0.086	0.164	$1.4 \cdot 10^{-12}$	$1.1 \cdot 10^{-17}$	$9.0 \cdot 10^{-18}$

Table 22: Adaptive mesh results of problem 1 for k=7 (Quadruple precision)

References

[1] U.M. Ascher and G. Bader, A new basis implementation for a mixed order boundary value ODE solver, SIAM J. Sci. Stat. Comp., 8, 1987, pp.483-500.

[2] U.M. 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.

[3] U.M. Ascher, J. Christiansen and R.D. Russell, Collocation software for boundary value ODEs, ACM Trans. Math. Soft., 7, 1981, pp.209-222.

[4] U.M. Ascher, R.M.M. Mattheij and R.D. Russell, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, Classics in Applied Mathematics Series, SIAM, Philadelphia, 1995.

[5] U.M. Ascher, S. Pruess and R.D. Russell, On spline basis selection for solving differential equations, SIAM J. Numer. Anal., 20, 1983, pp.121-142.

[6] C. De Boor and B. Swartz, Collocation at Gaussian points, SIAM J. Numer. Anal., 10, 1973, pp.582-606.

[7] W.H. Enright, K.R. Jackson, S.P. Nørsett and P.G. Thomsen, Interpolants for Runge-Kutta formulas, ACM Trans. Math. Soft., 12, 1986, pp.193-218.

[8] W.H. Enright and P.H. Muir, Superconvergent interpolants for the collocation solution of BVODEs, to appear in SIAM J. Sci. Comp.

[9] S. Karlin and J. Karon, On Hermite-Birkhoff interpolation, J. Approx. Theory, 6, 1972, pp.90-114.

[10] P. Muir and B. Owren, Order barriers and characterizations for continuous mono-implicit Runge-Kutta schemes, Math. Comp., 61, 1993, pp.675-699.

[11] S. Pruess, Interpolation schemes for collocation solutions of two point boundary value problems, SIAM J. Sci. Stat. Comp., 7, 1986, pp.322-333.

[12] S. Pruess and H. Jin, A stable high-order interpolation scheme for superconvergent data, SIAM J. Sci. Comp., 17, 1996, pp.714-724.

[13] S. Pruess and H. Jin, Uniformly superconvergent approximations for linear two-point boundary value problems, SIAM J. Numer. Anal., 35, 1998, pp.363-375.

[14] R. Sivasothinathan, A Bootstrap Approach for Constructing Superconvergent Interpolants, M.Sc. Thesis, University of Toronto, 1998.