Accurate approximate solution of partial differential equations at off-mesh points

W.H. Enright
University of Toronto

Numerical methods for partial differential equations often determine approximations that are more accurate at the set of discrete meshpoints than they are at the 'off-mesh' points in the domain of interest. These methods are generally most effective if they are allowed to adjust the location of the mesh points to match the local behaviour of the solution. Different methods will typically generate their respective approximations on incompatible, unstructured meshes and it can be difficult to evaluate the quality of a particular solution, or to visualise important properties of a solution. In this paper we will introduce a generic approach which can be used to generate approximate solution values at arbitrary points in the domain of interest for any method that determines approximations to the solution and low order derivatives at meshpoints. This approach is based on associating a set of 'collocation' points with each mesh element and requiring that the local approximation interpolate the meshpoint data and almost satisfy the partial differential equation at the collocation points. The accuracy associated with this interpolation/collocation approach is consistent with the 'meshpoint accuracy' of the underlying method.

The approach that we develop applies to a large class of methods and problems. It uses local information only and is therefore particularly suitable for implementation in a parallel or network computing environment. Numerical examples are given for some second order problems in two and three dimensions.

Categories and Subject Descriptors: G.1.8 [**Numerical Analysis**]: Partial Differential Equations; G.1.1 [**Numerical Analysis**]: Interpolation—*spline and piecewise polynomial interpolation*; G.1.2 [**Numerical Analysis**]: Approximation—*approximation of surfaces and contours*; G.4 [**Mathematics of Computing**]: Mathematical Software

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: collocation, rendering, scientific visualization

1. INTRODUCTION

In many areas of science and engineering one models phenomena of interest by systems of partial differential equations (PDEs). In most applications, one can assume

This research was supported by the Natural Science and Engineering Research Council of Canada and by Communications and Information Technology Ontario.
Address: Department Of Computer Science, University of Toronto, Toronto, Canada, M5S 3G4

the existence of an isolated solution, u, to the particular PDE but it is unlikely that a closed form expression exists. To investigate the behaviour of solutions, effective numerical methods have been developed to approximate the solution and visualisation tools have been developed which can display or 'render' important properties or characteristics of the approximate solutions.

Many numerical methods focus on the computation of accurate approximations at only a discrete set of mesh points in the domain associated with the problem definition. While it is possible to impose constraints on the mesh (and therefore force the method to produce accurate approximations at a pre-determined set of points) these methods are most effective if no constraints are imposed and the mesh is dynamically adjusted to match the local behaviour or difficulty of the problem. With such methods, one often observes problems where the distribution of mesh points is highly non-uniform over the domain of interest. In such cases, if one wishes to visualise properties of the solution on the whole domain, one must first generate extra data values in regions of the domain where there are few mesh points available from the numerical method. The focus of this paper is to develop a new generic approach to determine these off-mesh approximations reliably and efficiently.

There are two approaches that are now widely used for this purpose. The first such approach begins by determining a fine mesh that is adequate for rendering, as a refinement of the coarse mesh that is associated with the approximation generated by the underlying PDE method. The PDE method is then 're-applied' with the fine mesh specified to generate the data values required for rendering the approximate solution. This approach often delivers much more accuracy than is necessary and can be very expensive, particularly if the number of meshpoints associated with the fine mesh is significantly greater than the number of meshpoints associated with the coarse mesh.

The second approach is based on the use of multivariate splines. In this case spline interpolation of the coarse mesh data is used to define the piecewise polynomial which determines the data values required for the fine mesh. This approach can work well in many cases but it is generally more expensive than the approach we develop and it uses global information when refining an element of the coarse mesh.

We assume that the underlying PDE is a two-dimensional, second order problem of the form,

$$Lu = g(x, y, u, u_x, u_y), \tag{1}$$

where g is Lipschitz continuous and L is a given semi-linear differential operator defined by

$$Lu = a_1(x, y)u_{xx} + a_2(x, y)u_{yy} + a_3(x, y)u_{xy}.$$

The approach extends in an obvious way to three-dimensional problems and this is discussed in section 4. It can also be applied directly to first order, higher order, and mixed order problems but we do not consider such extensions in this paper.

We further assume that an underlying reliable numerical method is employed to determine approximations to the solution and first order derivatives on an unstructured triangular or rectangular mesh. At any mesh point, $(\bar{x}, \bar{y})$, the accuracies of these approximations, $\bar{u}, \bar{u}_x$, and $\bar{u}_y$ satisfy

$$u(\bar{x}, \bar{y}) - \bar{u} \;=\; O(h^{p_1}),$$

$$\begin{aligned} u_x(\bar{x},\bar{y}) - \bar{u}_x &= O(h^{p_2}), \\ u_y(\bar{x},\bar{y}) - \bar{u}_y &= O(h^{p_2}), \end{aligned} \tag{2}$$

where h is the maximum mesh spacing (in either the x or y direction).

The mesh points associated with the method's approximate solution will partition the domain of interest into a collection of mesh elements which are either triangles or rectangles. Our approach is to associate with each mesh element, e, a bivariate polynomial $p_{d,e}(x,y)$ of degree d, which can be viewed as approximating $u(x,y)$ for $(x,y) \in e$. Thus the collection of polynomials $p_{d,e}(x,y)$, over all mesh elements, will then define a piecewise polynomial approximation, $p_d(x,y)$, that is well defined for all (x,y) in the domain of interest. In the next section we outline how, for $d \leq 5$, one can determine a polynomial $\bar{p}_{d,e}(x,y)$ that interpolates the data values associated with the mesh points of e and 'almost' satisfies the PDE at a predetermined set of collocation points of e. It will be shown that the resulting piecewise polynomial satisfies

$$\begin{aligned} u(x,y) - \bar{p}_d(x,y) &= O(h^p), \\ u_x(x,y) - \frac{\partial \bar{p}_d(x,y)}{\partial x} &= O(h^{p-1}), \\ u_y(x,y) - \frac{\partial \bar{p}_d(x,y)}{\partial y} &= O(h^{p-1}), \end{aligned} \tag{3}$$

where $p = \min(p_1, p_2+1, d+1)$.

The approach that we have developed was motivated by our work on the numerical solution of ordinary differential equations (ODEs) where we have developed and applied to different classes of Runge-Kutta methods a 'bootstrapping' technique to extend standard discrete solutions to 'continuous' solutions ([Enright et al. 1986; Enright and Muir 1999; Enright and Sivasothinathan 1999]). These 'continuous Runge-Kutta' (CRK) formulas provide an efficient way to produce accurate approximations at off-mesh points in the interval of integration. In addition they can be used effectively in defining new error control and mesh-selection strategies which can lead to more effective implementations of the underlying discrete method. We have investigated ODE methods based on this approach for initial value problems [Enright 1993]; for boundary value problems [Enright and Muir 1996] (see also [Pruess 1986; Pruess and Jin 1996; Pruess and Jin 1998]); for delay differential equation [Enright and Hayashi 1997]; and for algebraic differential equations [Nguyen 1995].

Note that our approach is closely related to the standard approach of approximating tabulated data by interpolating splines (see for example [de Boor 1978] or [Chui 1988]). The difference is that, in this standard approach when one must introduce extra linear constraints to uniquely determine the associated piecewise polynomial, these additional constraints are based on smoothness of the piecewise polynomial (continuity of higher derivatives) at meshpoints rather than being based on almost satisfying the PDE at the collocation points. In the case where there are sufficient interpolating constraints to determine $\bar{p}_d(x,y)$ (as would be the situation for $d=2$ with rectangular mesh elements) our approach reduces to this standard approach. In general, for a given value of d, our approach will generate more accurate but less smooth global approximations. From a rendering point of view, accuracy to within a 'pixel' (at mesh boundaries) is equivalent to continuity and thus this trade-off may not be that relevant.

In the next section, we present an overview of our approach for general d. We then develop, in section 3, a detailed analysis for $3 \le d \le 6$ for rectangular meshes and then repeat this detailed analysis for triangular meshes. We conclude section 3 with a summary of the comparable results for three dimensional problems.

In section 4, we present some numerical results (using both exact mesh data and mesh data from two particular numerical methods) to illustrate that the behaviour predicted by our analysis can be observed over the range of accuracies actually used in solving practical problems. We conclude with a discussion of how the piecewise polynomials may prove to be useful in improving the effectiveness of the underlying discrete method as well as in other components of the mathematical modeling process.

2. THE GENERIC APPROACH

For $(\bar{x}_1, \bar{y}_1)$ a mesh point associated with e, let $s = (x - \bar{x}_1)/D_1, t = (y - \bar{y}_1)/D_2$ and consider representing $\bar{p}_{d,e}(x, y)$ by

$$\bar{p}_{d,e}(x, y) = \sum_{i=0}^{d} \sum_{j=0}^{d} c_{ij} s^i t^j, \tag{4}$$

where D_1 and D_2 are constants (to be specified later) that characterise the size of the mesh element e. We determine the $(d + 1)^2$ unknown coefficients, $c = [c_{00}, c_{01}, ...c_{dd}]^T$ by specifying $(d + 1)^2$ independent linear constraints. For each mesh point $(\bar{x}, \bar{y})$ associated with e we require that $\bar{p}_{d,e}(x, y)$ interpolate the data values $\bar{u}$, $\bar{u}_x$ and $\bar{u}_y$, and this leads to the three linear constraints

$$\sum_{i=0}^{d} \sum_{j=0}^{d} c_{ij} \bar{s}^i \bar{t}^j = \bar{u}, \tag{5}$$

$$\sum_{i=0}^{d} \sum_{j=0}^{d} i c_{ij} \bar{s}^{i-1} \bar{t}^j = D_1 \bar{u}_x, \tag{6}$$

$$\sum_{i=0}^{d} \sum_{j=0}^{d} j c_{ij} \bar{s}^i \bar{t}^{j-1} = D_2 \bar{u}_y, \tag{7}$$

where $\bar{s} = (\bar{x} - \bar{x}_1)/D_1$ and $\bar{t} = (\bar{y} - \bar{y}_1)/D_2$.

If each element has k mesh points associated with it we require $m = (d+1)^2 - 3k$ additional independent linear constraints to uniquely determine the coefficients of $\bar{p}_{d,e}(x, y)$. We accomplish this by choosing m distinct collocation points in e and imposing linear constraints that can be interpreted as requiring (1) to be 'almost' satisfied at these points.

From (4) we have

$$\frac{\partial^2 \bar{p}_{d,e}(x, y)}{\partial x^2} = (1/D_1)^2 \sum_{i=0}^{d} \sum_{j=0}^{d} i(i-1) c_{ij} \left(\frac{x - \bar{x}_1}{D_1}\right)^{i-2} \left(\frac{y - \bar{y}_1}{D_2}\right)^j,$$

$$\frac{\partial^2 \bar{p}_{d,e}(x,y)}{\partial y^2} = (1/D_2)^2 \sum_{i=0}^{d}\sum_{j=0}^{d} j(j-1)c_{ij}\left(\frac{x-\bar{x}_1}{D_1}\right)^{i}\left(\frac{y-\bar{y}_1}{D_2}\right)^{j-2}, \quad (8)$$

$$\frac{\partial^2 \bar{p}_{d,e}(x,y)}{\partial x \partial y} = (1/(D_1 D_2)) \sum_{i=0}^{d}\sum_{j=0}^{d} ij c_{ij}\left(\frac{x-\bar{x}_1}{D_1}\right)^{i-1}\left(\frac{y-\bar{y}_1}{D_2}\right)^{j-1}.$$

From (1) we see that $L\bar{p}_{d,e}(x,y)$ is a linear combination of the second order partial derivatives which from (8) implies that, for a fixed value of x and y, $L\bar{p}_{d,e}(x,y)$ is a linear combination of the unknown coefficients, c. The linear constraint that we associate with each collocation point $(\hat{x},\hat{y})$ is then a suitably scaled form of

$$L\bar{p}_{d,e}(\hat{x},\hat{y}) = g(\hat{x},\hat{y},\hat{u},\hat{u}_x,\hat{u}_y), \quad (9)$$

where $\hat{u}, \hat{u}_x$ and $\hat{u}_y$ are $O(h^q)$ approximations to $u(\hat{x},\hat{y}), u_x(\hat{x},\hat{y})$ and $u_y(\hat{x},\hat{y})$ respectively. Note that standard piecewise linear interpolation for rectangular or triangular meshes yields such approximations with $q = 2$ but some methods, such as finite element or collocation methods, may deliver off-mesh approximations at no extra cost with $2 < q < p$. For such methods the off-mesh accuracy, while not necessarily of optimal order, may be sufficiently accurate for rendering and our approach would not be required. Note also that if $h = \max(D_1, D_2)$ then the collocation constraint is better scaled (as $h \to 0$) if we multiply it by h^2.

The system of linear equations that we use to determine the unknown coefficients corresponds to satisfying (5), (6) and (7) for each of the k mesh points and satisfying (9) for each of the m collocation points. This system of linear equations can be written as

$$Wc = b, \quad (10)$$

where W depends only on the mesh points, the collocation points and the definition of L while the right hand side vector b depends on the mesh data associated with the method (assumed to satisfy (2)) and the $O(h^q)$ approximations associated with the collocation points. If we order the equations so that the first $3k$ rows of W correspond to interpolating $\bar{u}, \bar{u}_x$, and $\bar{u}_y$ at mesh points and the last m rows correspond to the collocation constraints (rescaled by multiplying both sides of (9) by h^2) then the vector b has the form

$$b = [\bar{u}_1, \bar{u}_2, \ldots \bar{u}_k, D_1\bar{u}_{x1}, \ldots D_1\bar{u}_{xk}, D_2\bar{u}_{y1}, \ldots D_2\bar{u}_{yk}, h^2\hat{g}_1, \ldots h^2\hat{g}_m]^T, \quad (11)$$

where

$$\hat{g}_i = g(\hat{x}_i, \hat{y}_i, \hat{u}_i, \hat{u}_{xi}, \hat{u}_{yi}),$$

and $(\hat{x}_i, \hat{y}_i) = (\bar{x}_1 + \hat{s}_i D_1, \bar{y}_1 + \hat{t}_i D_2)$. The points $(\hat{s}_i, \hat{t}_i), i = 1, 2 \cdots m$ define the collocation abscissae and, unless otherwise stated, we assume these values are fixed for a given problem and mesh.

If the matrix W is nonsingular then $\bar{p}_{d,e}(x,y)$ is uniquely defined and we will investigate the error, $u(x,y) - \bar{p}_{d,e}(x,y)$. To do this, we assume that the solution $u(x,y)$ is sufficiently differentiable and consider the polynomial $\tilde{p}_{d,e}(x,y)$ that would result if 'exact data' were used in the definition of the corresponding right hand side vector b in (10). That is, let $\tilde{c}$ (the vector of coefficients defining $\tilde{p}_{d,e}(x,y)$) be the solution of

$$W\tilde{c} = \tilde{b}, \quad (12)$$

where W is as in (10),

$$\tilde{b} = [u(\bar{x}_1, \bar{y}_1), ...u(\bar{x}_k, \bar{y}_k), D_1 u_x(\bar{x}_1, \bar{y}_1), ...D_1 u_x(\bar{x}_k, \bar{y}_k), D_2 u_y(\bar{x}_1, \bar{y}_1), ...D_2 u_y(\bar{x}_k, \bar{y}_k), h^2 g_1, ...h^2 g_m]^T,$$

and

$$g_i = g(\hat{x}_i, \hat{y}_i, u(\hat{x}_i, \hat{y}_i), u_x(\hat{x}_i, \hat{y}_i), u_y(\hat{x}_i, \hat{y}_i)).$$

From standard results in approximation theory we know that, for any $(x, y) \in e$, the error $|u(x, y) - \tilde{p}_{d,e}(x, y)|$ is $O(h^{d+1})$ while $|u_x(x, y) - \frac{\partial \tilde{p}_{d,e}(x,y)}{\partial x}|$ and $|u_y(x, y) - \frac{\partial \tilde{p}_{d,e}(x,y)}{\partial y}|$ are both $O(h^d)$. It follows from (10) and (12) that

$$\tilde{c} - c = W^{-1}(\tilde{b} - b).$$

In the analysis of this section we will use $||\cdot||$ to represent the max-norm for vectors and the associated induced matrix norm for matrices.

Under some mild assumptions on the coefficients $a_1(x, y), a_2(x, y), a_3(x, y)$ (defining L) and on the mesh refinement strategy it can be shown that, if the collocation abscissae can be chosen to avoid a singular matrix W, then $||W^{-1}||$ will be uniformly bounded as $h \to 0$, where $h = \max(D_1, D_2)$. This will be the case, for example, if we assume that the $a_j(x, y)$ are nonzero, continuous and bounded in magnitude; and that the mesh refinement strategy will partition an element e (with associated values h_e, D_1, D_2) by introducing a new meshpoint at the midpoint of each edge defining e. (Note that for rectangles or triangles this replaces e by four new elements, each with $h = h_e/2$ and D_1/D_2 unchanged.)

It is clear (from (2)) that the first k components of $(\tilde{b} - b)$ are $O(h^{p_1})$, the next $2k$ components are $O(h^{p_2+1})$ and (as g is Lipschitz continuous) the last m components are $O(h^{q+2})$. We will then have $||\tilde{c} - c|| = O(h^r)$ where $r = \min(q + 2, p_1, p_2 + 1)$. Furthermore since, for $(x, y) \in e$,

$$|u(x, y) - \bar{p}_{d,e}(x, y)| \leq |u(x, y) - \tilde{p}_{d,e}(x, y)| + |\tilde{p}_{d,e}(x, y) - \bar{p}_{d,e}(x, y)|$$

and

$$|\tilde{p}_{d,e}(x, y) - \bar{p}_{d,e}(x, y)| \leq (d + 1)^2 ||\tilde{c} - \bar{c}||$$

we have $|u(x, y) - \bar{p}_{d,e}(x, y)| = O(h^{r'})$ where $r' = \min(q + 2, d + 1, p_1, p_2 + 1)$. Similarly it can be shown that $|u_x(x, y) - \frac{\partial \bar{p}_{d,e}(x,y)}{\partial x}|$ and $|u_y(x, y) - \frac{\partial \bar{p}_{d,e}(x,y)}{\partial y}|$ are both $O(h^{r'-1})$.

If $q \geq p - 2$, we achieve the optimum order locally, as the associated piecewise polynomial $\bar{p}_d(x, y)$ satisfies (3). On the other hand, if $q < p - 2$, we can repeat the procedure using $\bar{p}_{d,e}(x, y)$ to define $O(h^{q+1})$ accurate collocation point information required to evaluate $\hat{g}_i$. That is, we define $b^{(1)}$ using this more accurate collocation point information and solve the linear system, $Wc^{(1)} = b^{(1)}$, for the more accurate vector $c^{(1)}$ (corresponding to the polynomial $\bar{p}^{(1)}_{d,e}(x, y)$) so that $||\tilde{c} - c^{(1)}|| = O(h^r)$ where $r = \min(q + 3, p)$. Clearly we can repeat the overall process R times, determining $c^{(R-1)}$, until $q + R + 1 \geq p$ at which point we have reached the optimum order.

It is straightforward to show that

$$||c^{(R)} - \tilde{c}|| = O(h^{q+R+2})$$

and that this implies

$$
\begin{aligned}
|u(x,y) - \bar{p}^{(R)}_{d,e}(x,y)| &= O(h^{q+R+2}),\\
|u_x(x,y) - \frac{\partial \bar{p}^{(R)}_{d,e}(x,y)}{\partial x}| &= O(h^{q+R+1}),\\
|u_y(x,y) - \frac{\partial \bar{p}^{(R)}_{d,e}(x,y)}{\partial y}| &= O(h^{q+R+1}).
\end{aligned}
$$

Note that, although the procedure is repeated R times, the matrix W does not change and only one matrix factorization is required. Also, since $q \geq 2$ and for most PDE methods, $p \leq 5$, R will rarely be larger than 2. Nevertheless, it is instructive to confirm the stability of the overall process by investigating the error associated with the vector $c^{(R-1)}$ (corresponding to the application of R 'bootstrapping' steps).

Since g is Lipschitz continuous a standard fixed-point analysis can be used to show that, for sufficiently small h the sequence $c^{(R)}$ converges as R increases and $||c^{(R)} - c^{(R-1)}|| \leq hK||c^{(R-1)} - c^{(R-2)}||$ for some $K \geq 0$ that depends only on d, the Lipschitz constant G, and a bound on $||W^{-1}||$. To see this, we first observe that

$$
\begin{aligned}
\bar{p}^{(R)}_{d,e}(\hat{x}_r,\hat{y}_r) - \bar{p}^{(R-1)}_{d,e}(\hat{x}_r,\hat{y}_r) &= \sum_{i=0}^{d}\sum_{j=0}^{d}(c^{(R)}_{ij} - c^{(R-1)}_{ij})\hat{s}_r^i\hat{t}_r^j,\\
\frac{\partial \bar{p}^{(R)}_{d,e}(\hat{x}_r,\hat{y}_r)}{\partial x} - \frac{\partial \bar{p}^{(R-1)}_{d,e}(\hat{x}_r,\hat{y}_r)}{\partial x} &= (1/h)\sum_{i=0}^{d}\sum_{j=0}^{d} i(c^{(R)}_{ij} - c^{(R-1)}_{ij})\hat{s}_r^{i-1}\hat{t}_r^j,\\
\frac{\partial \bar{p}^{(R)}_{d,e}(\hat{x}_r,\hat{y}_r)}{\partial y} - \frac{\partial \bar{p}^{(R-1)}_{d,e}(\hat{x}_r,\hat{y}_r)}{\partial y} &= (1/h)\sum_{i=0}^{d}\sum_{j=0}^{d} j(c^{(R)}_{ij} - c^{(R-1)}_{ij})\hat{s}_r^{i}\hat{t}_r^{j-1}.
\end{aligned}
$$

It follows that

$$
\begin{aligned}
|\bar{p}^{(R)}_{d,e}(\hat{x}_r,\hat{y}_r) - \bar{p}^{(R-1)}_{d,e}(\hat{x}_r,\hat{y}_r)| &\leq \left(\sum_{i=0}^{d}\sum_{j=0}^{d}\hat{s}_r^i\hat{t}_r^j\right)||c^{(R)} - c^{(R-1)}||\\
&\leq (d+1)^2||c^{(R)} - c^{(R-1)}||,\\
|\frac{\partial \bar{p}^{(R)}_{d,e}(\hat{x}_r,\hat{y}_r)}{\partial x} - \frac{\partial \bar{p}^{(R-1)}_{d,e}(\hat{x}_r,\hat{y}_r)}{\partial x}| &\leq (1/h)\left(\sum_{i=0}^{d}\sum_{j=0}^{d} i\hat{s}_r^{i-1}\hat{t}_r^j\right)||c^{(R)} - c^{(R-1)}||\\
&\leq \frac{d^3+d^2}{2h}||c^{(R)} - c^{(R-1)}||\\
|\frac{\partial \bar{p}^{(R)}_{d,e}(\hat{x}_r,\hat{y}_r)}{\partial y} - \frac{\partial \bar{p}^{(R-1)}_{d,e}(\hat{x}_r,\hat{y}_r)}{\partial y}| &\leq (1/h)\left(\sum_{i=0}^{d}\sum_{j=0}^{d} j\hat{s}_r^{i}\hat{t}_r^{j-1}\right)||c^{(R)} - c^{(R-1)}||\\
&\leq \frac{d^3+d^2}{2h}||c^{(R)} - c^{(R-1)}||
\end{aligned}
$$

Now since

$$
b^{(R)} - b^{(R-1)} = [0,\cdots 0, h^2(\hat{g}_1^{(R)} - \hat{g}_1^{(R-1)}),\cdots h^2(\hat{g}_m^{(R)} - \hat{g}_m^{(R-1)})]',
$$

where

$$
\begin{aligned}
|\hat{g}_r^{(R)} - \hat{g}_r^{(R-1)}| \;\le\; & G\,|\bar{p}_{d,e}^{(R-1)}(\hat{x}_r, \hat{y}_r) - \bar{p}_{d,e}^{(R-2)}(\hat{x}_r, \hat{y}_r)| \\
& + G\,|\frac{\partial \bar{p}_{d,e}^{(R-1)}(\hat{x}_r, \hat{y}_r)}{\partial x} - \frac{\partial \bar{p}_{d,e}^{(R-2)}(\hat{x}_r, \hat{y}_r)}{\partial x}| \\
& + G\,|\frac{\partial \bar{p}_{d,e}^{(R-1)}(\hat{x}_r, \hat{y}_r)}{\partial y} - \frac{\partial \bar{p}_{d,e}^{(R-2)}(\hat{x}_r, \hat{y}_r)}{\partial y}|
\end{aligned}
$$

we have

$$||b^{(R)} - b^{(R-1)}|| \le h^2[d^2 + (d^3 + d^2)/h]G||c^{(R-1)} - c^{(R-2)}||.$$

Since we also have

$$||c^{(R)} - c^{(R-1)}|| \le ||W^{-1}||\;||b^{(R)} - b^{(R-1)}||$$

we conclude that

$$||c^{(R)} - c^{(R-1)}|| \le h(h(d+1)^2 + d^3 + d^2)G||W^{-1}||\;||c^{(R-1)} - c^{(R-1)}||,$$

and the result follows.

In our discussion so far we have assumed that the approximating polynomial associated with the mesh element e is a bivariate polynomial of degree d and is represented by the vector c of $(d+1)^2$ coefficients. In some cases, it is more natural to restrict the approximating polynomial to be a bivariate polynomial of total degree d of the form

$$\bar{p}_{d,e}(x,y) = \sum_{i=0}^{d}\sum_{j=0}^{d-i} c_{ij} s^i t^j,$$

and therefore represented by a vector c of $(d+1)(d+2)/2$ coefficients, c_{ij}, that satisfy $0 \le i, j \le d$ and $i + j \le d$. We will refer to such an interpolant as a 'pure' bivariate polynomial. Using this class of polynomials has the advantage of leading to fewer unknowns for the same order of accuracy (as both polynomials will exhibit $O(h^{d+1})$ accuracy) but the number of terms contributing to the coefficient of h^{d+1} in the error expansion will generally be larger. Our approach can be applied to generate either type of approximating polynomial and we will consider examples of both in the remainder of this paper.

3. SPECIFIC EXAMPLES

Our approach can fail only if no set of m collocation abscissae can be found that results in a nonsingular matrix W. From standard results in linear algebra, we know that for rectangular or triangular mesh elements in two dimensions (with $k = 4$ or $k = 3$ respectively) the first $3k$ rows of W will always be linearly independent. In choosing the m collocation abscissae we must avoid the situation where W is inherently rank deficient for any problem. This can happen for example, if more than $d + 1$ collocation points are collinear and lie on a line that is parallel to the x or y axis ($\bar{p}_{d,e}(x, y)$ reduces to a univariate polynomial of degree d along this line with more than $d + 1$ constraints).

To investigate specific examples in detail, we will consider the widely studied two dimensional operator, $Lu = u_{xx} + u_{yy}$, and investigate how the approach can be effectively implemented. Consider the case $d = 3$ and e the rectangular element with corresponding meshpoints $[(\bar{x}_1, \bar{y}_1), (\bar{x}_2, \bar{y}_2), (\bar{x}_3, \bar{y}_3), (\bar{x}_4, \bar{y}_4)]$ defined by $\bar{x}_1 = \bar{x}_4 = X_0$, $\bar{x}_2 = \bar{x}_3 = X_1$, $\bar{y}_1 = \bar{y}_2 = Y_0$, $\bar{y}_3 = \bar{y}_4 = Y_1$ with $D_1 = X_1 - X_0$, $D_2 = Y_1 - Y_0$. If we partition the matrix

$$W = \begin{bmatrix} W_1 \\ \hline W_2 \end{bmatrix},$$

where W_1 corresponds to the first $3k$ rows (associated with interpolating the mesh data of the underlying method) and W_2 corresponds to the last m rows (associated with the collocation constraints), then we have

$$W_1 = \begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&2&0&0&0&3&0&0&0\\
0&0&0&0&1&1&1&1&2&2&2&2&3&3&3&3\\
0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0\\
0&1&2&3&0&1&2&3&0&1&2&3&0&1&2&3\\
0&1&2&3&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}.$$

Note that W_1 is independent of the problem and the mesh spacing and rows 1, 2, 4, 5, 6, 9 and 12 define seven equations which uniquely determine the unknowns $c_{00}, c_{01}, c_{02}, c_{03}, c_{10}, c_{20}, c_{30}$. That is,

$$[c_{00}, c_{01}, c_{02}, c_{03}, c_{10}, c_{20}, c_{30}]^T = \tilde{W}_1^{-1}[\bar{u}_1, \bar{u}_2, \bar{u}_4, D_1\bar{u}_{x1}, D_1\bar{u}_{x2}, D_2\bar{u}_{y1}, D_2\bar{u}_{y4}]^T,$$

with $\tilde{W}_1$ defined by

$$\tilde{W}_1 = \begin{bmatrix}
1&0&0&0&0&0&0\\
1&0&0&0&1&1&1\\
1&1&1&1&0&0&0\\
0&0&0&0&1&0&0\\
0&0&0&0&1&2&3\\
0&1&0&0&0&0&0\\
0&1&2&3&0&0&0
\end{bmatrix}.$$

Before discussing how to select the m collocation abscissae it is instructive to determine the maximum number of independent linear equations one can obtain using only the collocation constraints (9). For the PDE operator under consideration we observe, from (8), that in the linear constraint associated with collocation at $(\hat{x}, \hat{y})$, the coefficients of the unknowns c_{00}, c_{01}, c_{10} and c_{11} are all zero. Furthermore, the coefficient of c_{13} is always equal to $(D_1/D_2)^2$ times the coefficient of c_{31}. Similarly, the coefficients of c_{02} and c_{20}; the coefficients of c_{12} and c_{30}; and the coefficients of c_{03} and c_{21} are constant multiples of each other. Collectively these observations imply that the maximum rank of W_2 is eight. (This follows since W_2 has 16 columns with the first, second, fifth and sixth being zero; the eighth being a multiple of the fourteenth; the third being a multiple of the ninth; the seventh being a multiple of the thirteenth and the fourth being a multiple of the tenth.)

On the boundary of e, the bicubic we are prescribing reduces to a univariate cubic polynomial that is uniquely determined from the mesh data. (For example, along the boundary $(X_0, y), Y_0 \leq y \leq Y_1, \bar{p}_{3,e}$ is a cubic in y that interpolates the values $\bar{u}_0, \bar{u}_{y0}, \bar{u}_4, \bar{u}_{y4}$.) Note that this implies that collocation points must be interior to the mesh and the global piecewise polynomial $\bar{p}_3(x, y)$ will be continuous. In our numerical testing we have found that any set of four interior points will result in a nonsingular matrix W and the approach will be successful. Of course, the accuracy of $\bar{p}_{3,e}(x, y)$ depends on the mesh data and on the location of the collocation points. We have not yet investigated how the choice of collocation points affects the magnitude of the error coefficients.

In the case $d = 4$, one must specify 13 collocation points which result in a nonsingular W. A similar argument to the one made above for bicubics can be used to show that the maximum rank of the set of collocation constraints is 16. We have found, through numerical testing, that almost any choice of 13 interior points that avoids obvious excessive collinearities will result in a nonsingular W. In the case $d = 5$ one must specify 24 collocation points. Again, using a similar argument to the one made above for bicubics, the maximum rank of a set of collocation constraints can be shown to be 24. Again we have found that almost any choice of 24 interior points will result in a nonsingular W. For $d = 6$, the maximum rank of a set of collocation constraints can be shown to be 36 while 37 independent collocation constraints are required. This means that our approach will fail for any set of collocation points.

The detailed analysis presented here is for bivariate interpolation on rectangles. We have carried out a similar detailed analysis for total degree d bivariate interpolation on rectangles and for both classes of interpolants on triangles and the results are summarised in Table 1. In carrying out our detailed analysis for $2 \leq d \leq 6$, we were able to show (using in each case an argument similar to that presented earlier for bicubics) that for bivariate interpolation the maximum rank of a set of collocation constraints is d^2 if d is even and $d^2 - 1$ if d is odd. In the case of pure bivariate interpolation it is straightforward to show (by observing the linear dependence of the columns of the corresponding matrix, W_2) that for any $d \geq 2$ the maximum rank of a set of collocation constraints is $d(d+1)/2$. For each value of d Table 1 reports the following measures which can be used to quantify the limitations or applicability of our approach for the particular operator that we are considering:

d - the degree of the interpolant;

n_P - the number of unknown coefficients for pure bivariate interpolation, $(d+1)(d+2)/2$;

n_B - the number of unknown coefficients for bivariate interpolation, $(d+1)^2$;

R_P - the maximum number of independent collocation constraints for pure bivariate interpolation, $d(d+1)/2$;

R_B - the maximum number of independent collocation constraints for bivariate interpolation, $d^2 - (1 - (-1)^d)/2$;

m_P^R - the required number of collocation points for pure bivariate interpolation on rectangles, $(d+1)(d+2)/2 - 12$;

d	n_P	n_B	R_P	R_B	m_P^R	m_B^R	m_P^T	m_B^T
2	6	9	1	4	0	0	0	0
3	10	16	3	8	0	4	1	7
4	15	25	6	16	3	13	6	16
5	21	36	10	24	9	24	12	27
6	28	49	15	36	16	37	19	40

Table 1. Summary of restrictions for second order 2-D model problem

m_B^R - the required number of collocation points for bivariate interpolation on rectangles, $(d+1)^2 - 12$;

m_P^T - the required number of collocation points for pure bivariate interpolation on triangles, $(d+1)(d+2)/2 - 9$;

m_B^T - the required number of collocation points for bivariate interpolation on triangles, $(d+1)^2 - 9$;

As we have noted, our approach fails when the corresponding matrix W is singular and this will happen when the corresponding partitioned component W_2 does not have full rank. From Table 1, it is clear that on rectangles this will be the case for $d \geq 6$ for both classes of interpolants (as both $m_P^R > R_P$ and $m_B^R > R_B$). Similarly, for $d \geq 5$ and triangles, the approach must fail. For all other values of d, for both rectangles and triangles, we found in our numerical testing that the matrix W was always nonsingular (as long as obvious collinearities were avoided when the required number of collocation abscissae were selected).

Note also that Table 1 reveals that, using our approach, (with the restriction that all interpolated data be local to the mesh element) one must interpolate meshpoint derivative information if one wishes second or higher order accuracy at the off-mesh points. That is, for second order problems, if we were to interpolate the solution only at the k meshpoints, we would require an additional $m = (d+1)^2 - k$ collocation points for bivariate interpolation and for $d \geq 2$ (from Table 1) these could not all be linearly independent. A similar observation applies to pure bivariate interpolation.

We have also carried out a detailed analysis of the approach applied to the three dimensional operator, $Lu = u_{xx} + u_{yy} + u_{zz}$, for both generalised rectangles (cubes) and generalised triangles (tetrahedrons) as mesh elements. The results are summarised in Table 2, where the measures reported have the same meaning as in Table 1 (although their respective values as a function of d are different). From Table 2, we observe that the approach must fail for tricubic interpolation on generalised triangles ($m_B^T > R_B$); for triquartic interpolation on generalised rectangles ($m_B^R > R_B$); for pure quartic interpolation on generalised triangles ($m_P^T > R_P$); for triquartic interpolation on generalised triangles ($m_B^T > R_B$); and for all cases involving interpolation by quintics. For tricubics on generalised rectangles we have found, in our numerical testing, that all choices of 32 collocation abscissae that avoided obvious degeneracies resulted in a nonsingular matrix W and a successful application of our approach. Similarly, for triquadratic interpolation on generalised triangles, all choices of 11 collocation abscissae that avoided obvious degeneracies resulted in a nonsingular matrix W and, for pure tricubic interpolation on generalised triangles, all choices of 4 collocation abscissae that avoided obvious degeneracies resulted in a nonsingular matrix W.

d	n_P	n_B	R_P	R_B	m_P^R	m_B^R	m_P^T	m_B^T
2	10	27	1	14	0	0	0	11
3	20	64	4	40	0	32	4	48
4	35	125	10	88	3	93	19	109
5	56	216	20	160	24	184	40	200

Table 2. Summary of restrictions for second order 3-D model problem

For the remaining case of interest, pure triquartic interpolation on generalised rectangles, we were not able to find a suitable set of collocation points. In this case, the observed rank of the matrix W_1 (corresponding to the mesh data interpolation constraints) was deficient and this forces W to be singular for any set of collocation points.

4. NUMERICAL RESULTS

To illustrate that the convergence rates predicted by the analysis and discussion of the previous section remain valid over a range of meshes that are encountered in practice we present numerical results for a two-dimensional test problem and a three-dimensional test problem. The first problem is from [Rice and Boisvert 1985, p.430] with $\alpha = \beta = \pi$. The true solution is $u(x, y) = \cos(\beta y) + \sin\beta(x - y)$ and the corresponding PDE is:

$$u_{xx} + u_{yy} = \cos(\alpha y)u - (1 + \sin(\alpha x))u_x + f(x, y)$$

on the domain

$$0 \leq x \leq 1,\ 0 \leq y \leq 1,$$

with boundary conditions specified on the boundary of the unit square:

$$u(x, y) = \cos(\beta y) + \sin\beta(x - y).$$

Note that $f(x, y)$ is defined to force the true solution to have the desired functional form.

The second test problem is defined to have the true solution $u(x, y, z) = (x^2 - x)(y^2 - y)(z^2 + 1)$ by the PDE:

$$u_{xx} + u_{yy} + u_{zz} = u_z + 5u + f(x, y, z)$$

on the domain

$$0 \leq x \leq 1,\ 0 \leq y \leq 1,\ 0 \leq z \leq 1$$

with boundary conditions specified on the boundary of the unit cube:

$$u(x, y, z) = (x^2 - x)(y^2 - y)(z^2 + 1).$$

(As for the first problem, $f(x, y, z)$ is defined to force the true solution to have the desired functional form.)

It must be emphasised that the results reported in this section are for simple test problems where a uniform mesh is appropriate. These results are presented primarily to illustrate that observed performance can be consistent with the thoretical analysis over a reasonable range of accuracy requirements. Detailed evaluation of performance and/or a comparison with other approaches (over a range of problems, methods and meshes) remains to be done. It is clear that the overall effectiveness of our approach (or any approach for that matter) is limited by the ability of the underlying PDE method to produce accurate mesh data on a suitable (from an approximation theory viewpoint) coarse mesh. This implies that such investigations of performance will require significant care and effort.

For both test problems, we have investigated the performance of our approach using a sequence of uniform meshes where the mesh data was generated using the exact solution and the initial off-mesh information was generated using piecewise linear interpolation. For the first problem we also repeated the tests using data generated by the methods HODIE and INTERIOR_COLLOCATION of the ELLPACK package. All results were obtained using IEEE double precision arithmetic. The results are summarised in Tables 3 - 6 where the maximum observed off-mesh error for the solution and first order derivatives is reported ($ErrU$ and $ErrU_x$, $ErrU_y$, $ErrU_z$ respectively). These maximum errors are determined for the first problem by sampling the error at 100 points per mesh element and for the second problem by sampling the error at 125 points per mesh element. In each case, we report statistics for the interpolants corresponding to the application of one and two bootstrap steps (that is, for $R = 1$ and $R = 2$). In cases where the mesh data is not exact, we report the discrete meshpoint errors. We also report the average 'observed order', O_Ord, which is the average value of the base-2 logarithm of the ratio of successive entries in each column of the table. (Note that since each entry corresponds to reducing h by a factor of 2, this ratio should approach 2^p as $h \to 0$.)

In all cases, the tabulated results confirm that the expected rate of convergence for the interpolants $\bar{p}_{d,e}$ and $\bar{p}^{(1)}_{d,e}$ were achieved, even for the relatively coarse meshes that were used. The results also show that, although for the coarse meshes the second bootstrap step does not always improve the accuracy, as the mesh is refined $\bar{p}^{(1)}_{d,e}$ is consistently more accurate than $\bar{p}_{d,e}$ for both problems and for all values of d.

Table 6 shows that for the three dimensional test problem tricubic interpolation can be significantly more accurate than pure tricubic interpolation. More testing and/or analysis is necessary to determine whether this can be expected in general.

5. CONCLUSIONS AND DISCUSSION

We have introduced an inexpensive generic technique for determining accurate approximations to the solution and low order derivatives of a PDE at any point in the problem domain from the discrete approximation associated with a standard numerical method. We have presented this technique as a post processing procedure which can be applied after the underlying numerical method has successfully determined a suitable discrete solution on a mesh (which may have been adaptively determined as the method generated the discrete solution). An important area where this technique can be directly applied is in scientific visualisation where standard renderers require that the function to be displayed be known on a very fine mesh (consistent with the resolution of the display device). In cases where the function to be displayed is the solution of a PDE, it is rarely the case that the mesh appropriate for the renderer is consistent with the mesh associated with the underlying PDE method, and one must generate approximations to the solution for the

Mesh	R = 1			R = 2		
	$ErrU$	$ErrU_x$	$ErrU_y$	$ErrU$	$ErrU_x$	$ErrU_y$
Bicubic interpolation:						
2×2	$4.1\cdot10^{-2}$	$5.1\cdot10^{-1}$	$4.7\cdot10^{-1}$	$2.2\cdot10^{-2}$	$2.5\cdot10^{-1}$	$2.6\cdot10^{-1}$
4×4	$3.5\cdot10^{-3}$	$7.1\cdot10^{-2}$	$6.6\cdot10^{-2}$	$1.8\cdot10^{-3}$	$3.5\cdot10^{-2}$	$3.5\cdot10^{-2}$
8×8	$2.3\cdot10^{-4}$	$9.3\cdot10^{-3}$	$9.1\cdot10^{-3}$	$1.2\cdot10^{-4}$	$4.5\cdot10^{-3}$	$4.4\cdot10^{-3}$
16×16	$1.5\cdot10^{-5}$	$1.2\cdot10^{-3}$	$1.2\cdot10^{-3}$	$7.7\cdot10^{-6}$	$5.6\cdot10^{-4}$	$5.6\cdot10^{-4}$
O_Ord	3.8	2.9	2.9	3.8	2.9	2.9
Pure biquartic interpolation:						
2×2	$4.5\cdot10^{-1}$	$3.7\cdot10^{0}$	$3.7\cdot10^{0}$	$7.0\cdot10^{-1}$	$6.6\cdot10^{0}$	$6.6\cdot10^{0}$
4×4	$1.6\cdot10^{-2}$	$2.4\cdot10^{-1}$	$2.4\cdot10^{-1}$	$2.0\cdot10^{-2}$	$3.7\cdot10^{-1}$	$3.7\cdot10^{-1}$
8×8	$5.3\cdot10^{-4}$	$1.4\cdot10^{-2}$	$1.4\cdot10^{-2}$	$5.8\cdot10^{-4}$	$2.1\cdot10^{-2}$	$2.1\cdot10^{-2}$
16×16	$2.1\cdot10^{-5}$	$8.2\cdot10^{-4}$	$6.8\cdot10^{-4}$	$1.7\cdot10^{-5}$	$1.2\cdot10^{-3}$	$1.2\cdot10^{-3}$
O_Ord	4.8	4.0	4.1	5.1	4.1	4.1
Biquartic interpolation:						
2×2	$1.0\cdot10^{-1}$	$5.4\cdot10^{-1}$	$6.3\cdot10^{-1}$	$1.0\cdot10^{-1}$	$6.6\cdot10^{-1}$	$6.5\cdot10^{-1}$
4×4	$4.6\cdot10^{-3}$	$5.2\cdot10^{-2}$	$6.4\cdot10^{-2}$	$3.0\cdot10^{-3}$	$3.7\cdot10^{-2}$	$3.8\cdot10^{-2}$
8×8	$2.6\cdot10^{-4}$	$7.0\cdot10^{-3}$	$7.9\cdot10^{-3}$	$8.1\cdot10^{-5}$	$2.1\cdot10^{-3}$	$2.1\cdot10^{-3}$
16×16	$1.6\cdot10^{-5}$	$9.1\cdot10^{-4}$	$9.7\cdot10^{-4}$	$2.3\cdot10^{-6}$	$1.2\cdot10^{-4}$	$1.2\cdot10^{-4}$
O_Ord	4.2	3.1	3.1	5.1	4.1	4.1
Pure biquintic interpolation:						
2×2	$2.1\cdot10^{-1}$	$1.7\cdot10^{0}$	$1.8\cdot10^{0}$	$7.0\cdot10^{-2}$	$5.8\cdot10^{-1}$	$5.9\cdot10^{-1}$
4×4	$3.6\cdot10^{-3}$	$7.2\cdot10^{-2}$	$7.8\cdot10^{-2}$	$2.2\cdot10^{-3}$	$3.5\cdot10^{-2}$	$3.6\cdot10^{-2}$
8×8	$2.3\cdot10^{-4}$	$7.8\cdot10^{-3}$	$7.8\cdot10^{-3}$	$4.0\cdot10^{-5}$	$1.3\cdot10^{-3}$	$1.3\cdot10^{-3}$
16×16	$1.5\cdot10^{-5}$	$9.5\cdot10^{-4}$	$9.6\cdot10^{-4}$	$6.7\cdot10^{-7}$	$4.8\cdot10^{-5}$	$4.5\cdot10^{-5}$
O_Ord	5.3	3.6	3.6	5.6	4.5	4.6
Biquintic interpolation:						
2×2	$9.1\cdot10^{-1}$	$6.8\cdot10^{0}$	$6.1\cdot10^{0}$	$1.2\cdot10^{1}$	$9.5\cdot10^{1}$	$8.4\cdot10^{1}$
4×4	$1.5\cdot10^{-2}$	$2.3\cdot10^{-1}$	$2.0\cdot10^{-1}$	$1.4\cdot10^{-1}$	$2.1\cdot10^{0}$	$1.9\cdot10^{0}$
8×8	$2.7\cdot10^{-4}$	$1.2\cdot10^{-2}$	$1.1\cdot10^{-2}$	$1.3\cdot10^{-3}$	$3.9\cdot10^{-2}$	$3.5\cdot10^{-2}$
16×16	$1.5\cdot10^{-5}$	$1.1\cdot10^{-3}$	$1.0\cdot10^{-3}$	$1.1\cdot10^{-5}$	$7.0\cdot10^{-4}$	$6.3\cdot10^{-4}$
O_Ord	5.3	4.2	4.2	5.6	5.7	5.7

Table 3. Results for problem 1 on rectangles with exact mesh data

Mesh	R = 1			R = 2		
	$ErrU$	$ErrU_x$	$ErrU_y$	$ErrU$	$Erru_x$	$Erru_y$
Discrete Meshpoint Error:						
4×4	$3.2\cdot10^{-5}$	$2.7\cdot10^{-1}$	$4.8\cdot10^{-1}$			
8×8	$5.3\cdot10^{-7}$	$2.9\cdot10^{-3}$	$5.2\cdot10^{-3}$			
16×16	$2.6\cdot10^{-9}$	$3.2\cdot10^{-5}$	$1.8\cdot10^{-4}$			
O_Ord	6.8	6.5	5.7			
Bicubic interpolation:						
4×4	$1.9\cdot10^{-2}$	$2.7\cdot10^{-1}$	$4.8\cdot10^{-1}$	$1.9\cdot10^{-2}$	$2.7\cdot10^{-1}$	$4.8\cdot10^{-1}$
8×8	$2.5\cdot10^{-4}$	$1.1\cdot10^{-2}$	$8.8\cdot10^{-3}$	$2.2\cdot10^{-4}$	$6.3\cdot10^{-3}$	$5.6\cdot10^{-3}$
16×16	$1.5\cdot10^{-5}$	$1.2\cdot10^{-3}$	$1.1\cdot10^{-3}$	$9.3\cdot10^{-6}$	$6.2\cdot10^{-4}$	$5.6\cdot10^{-4}$
O_Ord	5.1	3.9	4.4	5.5	4.4	4.9
Biquartic interpolation:						
4×4	$3.4\cdot10^{-2}$	$4.5\cdot10^{-1}$	$4.8\cdot10^{-1}$	$4.1\cdot10^{-2}$	$4.4\cdot10^{-1}$	$5.3\cdot10^{-1}$
8×8	$2.8\cdot10^{-4}$	$6.9\cdot10^{-3}$	$7.8\cdot10^{-3}$	$2.5\cdot10^{-4}$	$5.8\cdot10^{-3}$	$6.2\cdot10^{-3}$
16×16	$1.6\cdot10^{-5}$	$9.1\cdot10^{-4}$	$9.7\cdot10^{-4}$	$4.1\cdot10^{-6}$	$2.0\cdot10^{-4}$	$1.9\cdot10^{-4}$
O_Ord	5.5	4.5	4.5	6.6	5.6	5.7
Biquintic interpolation:						
4×4	$3.6\cdot10^{-2}$	$4.3\cdot10^{-1}$	$4.8\cdot10^{-1}$	$1.4\cdot10^{-1}$	$2.0\cdot10^{0}$	$1.8\cdot10^{0}$
8×8	$3.2\cdot10^{-4}$	$1.5\cdot10^{-2}$	$1.1\cdot10^{-2}$	$1.2\cdot10^{-3}$	$3.7\cdot10^{-2}$	$3.4\cdot10^{-2}$
16×16	$1.5\cdot10^{-5}$	$1.1\cdot10^{-3}$	$1.0\cdot10^{-3}$	$1.1\cdot10^{-5}$	$6.9\cdot10^{-4}$	$6.6\cdot10^{-4}$
O_Ord	5.6	4.3	4.5	6.8	5.8	5.7

Table 4. Results for problem 1 with mesh data from HODIE of ELLPACK

Mesh	R = 1			R = 2		
	$ErrU$	$ErrU_x$	$ErrU_y$	$ErrU$	$Erru_x$	$Erru_y$
Discrete Meshpoint Error:						
4×4	$4.9\cdot10^{-4}$	$3.2\cdot10^{-3}$	$4.0\cdot10^{-3}$			
8×8	$3.3\cdot10^{-5}$	$2.0\cdot10^{-4}$	$2.4\cdot10^{-4}$			
16×16	$2.1\cdot10^{-6}$	$2.1\cdot10^{-5}$	$2.9\cdot10^{-5}$			
O_Ord	3.9	3.6	3.6			
Bicubic interpolation:						
4×4	$3.7\cdot10^{-3}$	$6.9\cdot10^{-2}$	$6.6\cdot10^{-2}$	$1.7\cdot10^{-3}$	$3.5\cdot10^{-2}$	$3.7\cdot10^{-2}$
8×8	$2.5\cdot10^{-4}$	$9.3\cdot10^{-3}$	$9.1\cdot10^{-3}$	$1.2\cdot10^{-4}$	$4.5\cdot10^{-3}$	$4.6\cdot10^{-3}$
16×16	$1.6\cdot10^{-5}$	$1.2\cdot10^{-3}$	$1.2\cdot10^{-3}$	$7.5\cdot10^{-6}$	$5.6\cdot10^{-4}$	$5.7\cdot10^{-4}$
O_Ord	3.9	2.9	2.9	3.9	3.0	3.0
Biquartic interpolation:						
4×4	$4.9\cdot10^{-3}$	$4.9\cdot10^{-2}$	$6.2\cdot10^{-2}$	$3.2\cdot10^{-3}$	$4.0\cdot10^{-2}$	$3.9\cdot10^{-2}$
8×8	$2.8\cdot10^{-4}$	$6.8\cdot10^{-3}$	$7.8\cdot10^{-3}$	$1.0\cdot10^{-4}$	$2.2\cdot10^{-3}$	$2.1\cdot10^{-3}$
16×16	$1.7\cdot10^{-5}$	$9.0\cdot10^{-4}$	$9.7\cdot10^{-4}$	$3.9\cdot10^{-6}$	$1.3\cdot10^{-4}$	$1.2\cdot10^{-4}$
O_Ord	4.1	2.9	3.0	4.8	4.1	4.2
Biquintic interpolation:						
4×4	$1.5\cdot10^{-2}$	$2.3\cdot10^{-1}$	$2.0\cdot10^{-1}$	$1.4\cdot10^{-1}$	$2.1\cdot10^{0}$	$1.9\cdot10^{0}$
8×8	$2.8\cdot10^{-4}$	$1.2\cdot10^{-2}$	$1.1\cdot10^{-2}$	$1.3\cdot10^{-3}$	$3.9\cdot10^{-2}$	$3.5\cdot10^{-2}$
16×16	$1.6\cdot10^{-5}$	$1.0\cdot10^{-3}$	$1.1\cdot10^{-3}$	$1.2\cdot10^{-5}$	$7.4\cdot10^{-4}$	$6.5\cdot10^{-4}$
O_Ord	4.9	3.9	3.8	6.8	5.7	5.8

Table 5. Results for problem 1 with mesh data from INTERIOR_COLLOCATION of ELLPACK

Mesh	$ErrU$	$ErrU_x$	$ErrU_y$	$ErrU_z$
	Pure tricubic interpolation:			
2x2x2	$2.1\cdot10^{-2}$	$1.4\cdot10^{-1}$	$1.4\cdot10^{-1}$	$4.9\cdot10^{-2}$
4x4x4	$1.9\cdot10^{-3}$	$2.5\cdot10^{-2}$	$2.5\cdot10^{-2}$	$1.1\cdot10^{-2}$
8x8x8	$1.4\cdot10^{-4}$	$3.7\cdot10^{-3}$	$3.7\cdot10^{-3}$	$1.7\cdot10^{-3}$
O_Ord	3.6	2.6	2.6	2.4
	Tricubic interpolation with R = 1:			
2x2x2	$4.7\cdot10^{-3}$	$4.2\cdot10^{-2}$	$4.3\cdot10^{-2}$	$2.0\cdot10^{-2}$
4x4x4	$3.4\cdot10^{-4}$	$5.8\cdot10^{-3}$	$5.8\cdot10^{-3}$	$2.6\cdot10^{-3}$
8x8x8	$2.2\cdot10^{-5}$	$7.4\cdot10^{-4}$	$7.4\cdot10^{-4}$	$3.4\cdot10^{-4}$
O_Ord	3.9	2.9	2.9	2.9
	Tricubic interpolation with R = 2:			
2x2x2	$1.7\cdot10^{-3}$	$1.4\cdot10^{-2}$	$1.7\cdot10^{-2}$	$1.3\cdot10^{-2}$
4x4x4	$2.7\cdot10^{-5}$	$4.4\cdot10^{-4}$	$5.3\cdot10^{-4}$	$4.0\cdot10^{-4}$
8x8x8	$4.4\cdot10^{-7}$	$1.5\cdot10^{-5}$	$1.7\cdot10^{-5}$	$1.3\cdot10^{-5}$
O_Ord	6.0	4.9	5.0	5.0

Table 6. Results for problem 2 on generalised rectangles with exact mesh data

'renderer mesh' by a technique such as ours. Similarly, in applications that involve the computation of level surfaces or contour plots of functions, the technique can be directly applied.

One can also consider using the interpolating strategies that we have developed to help improve some of the heuristic strategies of the underlying numerical method. That is, rather than consider interpolation as a post-processing step, we allow the underlying method to use the interpolant (possibly associated with a preliminary intermediate mesh element) in its own internal computations. In the case of ODEs, we have found that this access to a continuous intermediate approximate solution can be very useful to the underlying discrete method for such things as stepsize selection, error estimation and discontinuity detection. In the case of PDEs, strategies related to multigrid calculations, discontinuity detection and mesh refinement are obvious examples where access to these interpolants may lead to more effective heuristics.

Throughout this paper, we have assumed that the underlying numerical method generates discrete approximations to the solution and low order derivatives at the meshpoints. We have done this for several reasons. In particular, it is the obvious generalisation to PDEs of the natural assumption that can be made in ODEs (where discrete methods for an m^{th}-order ODE generate approximations to the solution and the first $(m-1)$ derivatives at the meshpoints). It is also often the case that, in order to compute $O(h^{p_1})$ approximations to the solution at the meshpoints, the underlying method must also generate approximations to these lower order derivatives (although they may not be explicitly identified or available to the user). Nevertheless, there are methods for which the assumption (2) is difficult to satisfy for p_2 close to p_1. In these cases one can consider replacing the derivative interpolation constraints with additional solution interpolation constraints associated with 'nearby' meshpoints. For example for rectangular meshes one can associate with each meshpoint the nearest meshpoint exterior to e in each of the coordinate directions. (An obvious adjustment would be made for elements on the boundary.) This would change the definition of the component matrix W_1 and require modest changes in the analysis but most of the results would remain true. The errors would depend on the ratios of adjacent mesh widths, and would degrade for non-uniform meshes but the interpolants would still be defined in terms of relatively local information and similar numerical performance is expected.

ACKNOWLEDGMENTS

The author would like to thank Professor P. Muir for several stimulating discussions and for his comments which influenced this work. He would also like to thank Professors C. Christara and G. Fairweather for their helpful suggestions and comments.

REFERENCES

CHUI, C. 1988. *Multivariate Splines.* CBMS-NSF Regional Conf. Series in App. Math. SIAM, Philadelphia.

DE BOOR, C. 1978. *A Practical Guide to Splines*, Volume 27 of *Applied Mathematical Sciences.* Springer-Verlag, New York.

ENRIGHT, W. 1993. The relative efficiency of alternative defect control schemes for high order Runge-Kutta formulas. *SIAM J. Numer. Anal. 30*, 5, 1419–1445.

ENRIGHT, W. AND HAYASHI, H. 1997. A delay differential equation solver based on a continuous Runge-Kutta method with defect control. *Numerical Algorithms 16*, 349–364.

ENRIGHT, W., JACKSON, K., NØRSETT, S., AND THOMSEN, P. 1986. Interpolants for Runge-Kutta formulas. *ACM Trans. Math. Soft. 12*, 193–218.

ENRIGHT, W. AND MUIR, P. 1996. A Runge-Kutta type boundary value ODE solver with defect control. *SIAM J. Sci. Comput. 17*, 479–497.

ENRIGHT, W. AND MUIR, P. 1999. Superconvergent interpolants for the collocation solution of BVODEs. *SIAM J. Sci. Comput.* to appear.

ENRIGHT, W. AND SIVASOTHINATHAN, R. 1999. Superconvergent interpolants for collocation methods applied to mixed order BVODEs. submitted to ACM Trans. Math Soft.

NGUYEN, H. 1995. *Interpolation and error control schemes for algebraic differential equations using continuous implicit Runge-Kutta methods.* Ph. D. thesis, University of Toronto. (also appeared as Dept. Comp. Sc. Tech. Rep. No. 298/95).

PRUESS, S. 1986. Interpolation schemes for collocation solutions of two point boundary value problems. *SIAM J. Sci. Stat. Comput. 7*, 322–333.

PRUESS, S. AND JIN, H. 1996. A stable high-order interpolation scheme for superconvergent data. *SIAM J. Sci. Comput. 17*, 714–724.

PRUESS, S. AND JIN, H. 1998. Uniformly superconvergent approximations for linear two-point boundary value problems. *SIAM J. Numer. Anal. 35*, 2, 363–375.

RICE, J. AND BOISVERT, R. 1985. *Solving Elliptic Problems using ELLPACK*, Volume 2 of *Springer Computational Mathematics.* Springer-Verlag, New York.