Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 225 (2007) 1409–1426
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A three-dimensional finite volume method based on radial basis functions for the accurate computational modelling of nonlinear diffusion equations

T.J. Moroney *, I.W. Turner

School of Mathematical Sciences, Queensland University of Technology, Gardens Point Campus, GPO Box 2434, Brisbane Q4001, Australia

Received 20 July 2006; received in revised form 12 January 2007; accepted 29 January 2007
Available online 9 February 2007

Abstract

We investigate the effectiveness of a finite volume method incorporating radial basis functions for simulating nonlinear diffusion processes. Past work conducted in two dimensions is extended to produce a three-dimensional discretisation that employs radial basis functions (RBFs) as a means of local interpolation. When combined with Gaussian quadrature integration methods, the resulting finite volume discretisation leads to accurate numerical solutions without the need for very fine meshes, and the additional overheads they entail.

The resulting nonlinear, algebraic system is solved efficiently using a Jacobian-free Newton–Krylov method. By employing the method as an extension of existing shape function-based approaches, the number of nonlinear iterations required to achieve convergence can be reduced while also permitting an effective preconditioning technique.

Results highlight the improved accuracy offered by the new method when applied to three test problems. By successively refining the meshes, we are also able to demonstrate the increased order of the new method, when compared to a traditional shape function-based method. Comparing the resources required for both methods reveals that the new approach can be many times more efficient at producing a solution of a given accuracy.

Keywords: Control volume-finite element; Jacobian-free Newton–Krylov; Unstructured tetrahedral mesh; Gaussian quadrature

1. Introduction

The numerical solution of partial differential equations (PDEs) is an important problem in many fields of scientific and industrial simulation. The finite volume method, originally derived in [33], has proved particularly popular with computational fluid dynamicists for simulating a wide range of important applications and physical processes [1,2,5,28,18,22,14,35,42–44]. The attraction of the method lies in both its conservative nature and ability to be implemented on both structured or unstructured meshes.

* Corresponding author.
E-mail address: t.moroney@qut.edu.au (T.J. Moroney).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.01.029

The basic principle of the finite volume method sees the discretisation of the solution domain into a set of non-overlapping finite volumes and thereafter, the integral representation of the underlying conservation laws are approximated over these volumes using some appropriate numerical strategy. The process usually commences with the generation of a mesh that comprises, for example, hexahedral or tetrahedral elements in three dimensions. In the vertex-centred approach the computed quantities are stored at the vertices of the elements and the vertex values of the dependent variable play an important role in the interpolation methods required to reconstruct fluxes. Traditionally, the midpoint rule has been the favoured method to approximate the surface integrals involving these fluxes. However, as has been noted previously [21,31] this quadrature rule achieves second order accuracy only when the flux evaluations are sufficiently accurate. To this end, a variety of techniques have been explored to reconstruct the fluxes at the face centroids to at least second order accuracy using some set of strategically chosen local nodal values of the dependent variable. Recently, some authors have proposed gradient approximations using either Green–Gauss reconstruction techniques or least squares based methods [2,6,20,22,14,44] to achieve this second order spatial accuracy.

Rather than extend or improve upon this existing body of work, we investigate the possibility of using alternative methods of interpolation that might also yield high-order gradient and flux approximations. The method of radial basis functions (RBFs) is one such method of scattered data interpolation [37]. Its renowned high accuracy, particularly when using the multiquadric basis function [16] has seen it find application in a diverse range of fields. In the review paper [17], applications of multiquadrics to the fields of geodesy, geophysics, surveying, mapping, photogrammetry, remote sensing, signal processing, geography, digital terrain models and hydrology are presented. Indeed, for many years RBFs have even been used to solve PDEs, through the use of collocation methods [12,19,23,27]. However, the collocation approach does not share many of the desirable properties of the finite volume method, such as local conservation and the ability to work with sparse Jacobian matrices. Additionally, the overhead of using RBFs with large three-dimensional point sets can be substantial due to the high cost in determining the RBF coefficients via the solution of a large, dense matrix system.

Past work [29] has shown that the method of radial basis functions can be successfully applied as a means of local gradient interpolation in a two-dimensional finite volume framework. When combined with Gaussian quadrature methods for integration, the precision of the resulting discretisation allows for accurate solutions to be generated using coarse meshes. The use of RBFs as local interpolants, rather than global interpolants, maintains the sparsity of the Jacobian matrix and greatly reduces the computational burden of computing the RBF coefficients. Furthermore, careful solution of the local RBF linear systems using a truncated singular value decomposition ensures accurate gradient estimates for use in the flux reconstruction at the finite volume cell faces.

In this work we extend the two-dimensional framework described in [29] to three dimensions, over unstructured, tetrahedral meshes. The generation and optimisation of such meshes can be costly, and obtaining high accuracy through refined meshes can involve the solution of very large nonlinear systems. Thus we consider the ability of a method to achieve high accuracy on relatively coarse meshes to be a valuable property.

In addition to an accurate discretisation, an effective finite volume method must also employ some means of solving the resultant nonlinear system of equations. Inexact Newton methods implemented using a Jacobian-free, Krylov-based linear solver [3,4,8,9,24,25,34] are growing in popularity as a means for solving these kinds of problems [26]. We implement one such method, and demonstrate how a two-stage solution process, using the method of RBFs as a "corrector" to a traditional shape function-based approach, can be beneficial at improving both the convergence of the inner (linearised) iterations and the outer (nonlinear Newton) iterations.

This paper comprises two main sections. In Section 2 we present the methods of integration and interpolation in a three-dimensional framework, along with the proposed two-stage solution process and associated method of preconditioning. In Section 3 we present the results of numerical experiments that demonstrate the accuracy of the method, and its efficiency compared to a traditional shape function-based approach on a refined mesh. Finally in Section 4 the work is concluded and the key findings summarised. We also hint at the future research directions for this new finite volume discretisation strategy.

2. Finite volume formulation

The finite volume method is built upon the idea of constructing control volumes around every node in the mesh. In this work a vertex-centred scheme over unstructured, tetrahedral meshes is considered. To construct the control volumes, each tetrahedral element is divided into four sub-control volumes. The external faces of each sub-control volume are constructed by connecting the element's centroid, the centroids of its faces, and the midpoints of its edges. This forms a hexahedral sub-control volume, as shown in Fig. 1. Sub-control volumes from neighbouring elements combine to form the resulting control volume, an example of which is illustrated in Fig. 2(a). Fig. 2(b) depicts three adjoining control volumes, one of which shares some of its faces with the domain boundary.

The PDE under consideration in this work is the following nonlinear, steady-state diffusion equation

$$\nabla \cdot \mathbf{D}\nabla u + S = 0, \tag{1}$$

where $\mathbf{D} = \mathrm{diag}(D_{xx}, D_{yy}, D_{zz})$ and D_{xx}, D_{yy}, D_{zz} along with S can be nonlinear functions of u.

Integration over an arbitrary control volume V_i with boundary surface σ_i enables (1) to be written in control volume form:

$$\underbrace{\int\int_{\sigma_i} \mathbf{D}\nabla u \cdot \hat{\mathbf{n}}\,\mathrm{d}\sigma}_{\text{diffusive flux}} + \underbrace{\int\int\int_{V_i} S\,\mathrm{d}V}_{\text{source}} = 0. \tag{2}$$

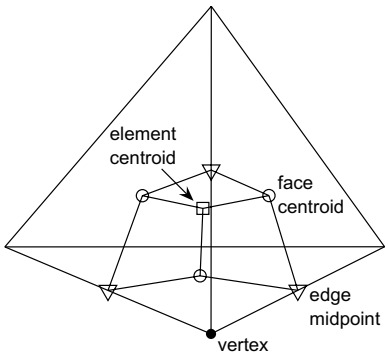

Fig. 1. A sub-control volume.

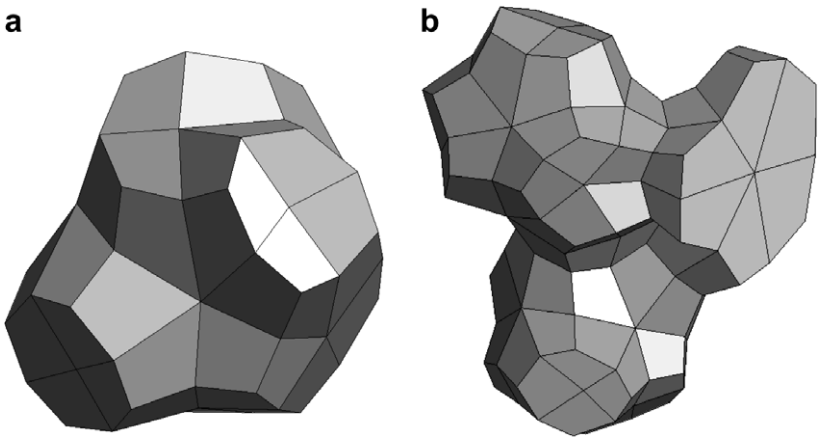

Fig. 2. (a) Single control volume; (b) three adjoining control volumes.

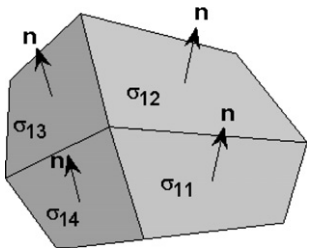

Fig. 3. Control volume faces and normals.

The discretisation of (2) relies on computing approximations to the diffusive flux and source terms using only the quantities $u_i,\ i = 1,\ldots,N$: the values of u at the N mesh nodes $\mathbf{x}_i,\ i = 1,\ldots,N$.

2.1. Integration

Numerical quadrature rules can be used to approximate both the diffusive flux and source terms of (2). In the case of the diffusive flux, the integral over the surface σ_i is expressed as the sum of integrals over each control volume face σ_{ij} (see Fig. 3):

$$\int\int_{\sigma_i} \mathbf{D}\nabla u \cdot \hat{\mathbf{n}}\,\mathrm{d}\sigma = \sum_{j=1}^{N_{f_i}} \int\int_{\sigma_{ij}} \mathbf{D}\nabla u \cdot \hat{\mathbf{n}}\,\mathrm{d}\sigma, \tag{3}$$

where σ_i comprises the N_{f_i} faces $\sigma_{ij},\ j = 1,\ldots,N_{f_i}$. In the case of the source component, a single volume integral approximation is required.

In the classical finite volume theory, a single point quadrature rule is often used for both the diffusive flux and source terms, yielding the approximations

$$\int\int_{\sigma_{ij}} \mathbf{D}\nabla u \cdot \hat{\mathbf{n}}\,\mathrm{d}\sigma \approx [\mathbf{D}\nabla u \cdot \hat{\mathbf{n}}]_{\mathbf{m}_{ij}} A(\sigma_{ij}) \tag{4}$$

and

$$\int\int\int_{V_i} S\,\mathrm{d}V \approx S_i \Delta V_i \tag{5}$$

respectively, where $\mathbf{m}_{ij}$ is the centroid of face σ_{ij}, $A(\sigma_{ij})$ is its area, and ΔV_i is the volume of control volume V_i.

In this work, we also implement four and eight-point Gaussian quadrature rules in place of (4) and (5) respectively. For the diffusive flux surface integrals, each control volume face σ_{ij} is quadrilateral in shape (refer again to Fig. 3), and so by mapping onto the square $[-1,1]^2$, four-point Gaussian quadrature in two dimensions can be applied. In the case of the source volume integral, denote by V_{ij} the jth sub-control volume of control volume V_i which has N_{v_i} sub-control volumes altogether. Each V_{ij} for $j = 1\ldots N_{v_i}$ is hexahedral in shape, and so by mapping onto the cube $[-1,1]^3$, eight-point Gaussian quadrature in three dimensions can be applied. The required mappings can be found in most finite element texts ([7] for example). Here we denote them symbolically by $\mathbf{r}_{ij}$ and $\mathbf{p}_{ij}$, respectively:

$$\begin{aligned} &\sigma_{ij} : \mathbf{x} = \mathbf{r}_{ij}(\xi,\eta), \qquad -1 \leqslant \xi \leqslant 1, \quad -1 \leqslant \eta \leqslant 1, \\ &V_{ij} : \mathbf{x} = \mathbf{p}_{ij}(\xi,\eta,\zeta), \quad -1 \leqslant \xi \leqslant 1, \quad -1 \leqslant \eta \leqslant 1, \ -1 \leqslant \zeta \leqslant 1. \end{aligned}$$

Using these transformations, and the standard Gauss-Legendre quadrature points $t_1 = -\frac{\sqrt{3}}{3}$, $t_2 = \frac{\sqrt{3}}{3}$ and weights $w_1 = w_2 = 1$, we obtain

$$\int\int_{\sigma_{ij}} \mathbf{D}\nabla u \cdot \hat{\mathbf{n}}\,\mathrm{d}\sigma \approx \sum_{\alpha=1}^{2}\sum_{\beta=1}^{2} w_\alpha w_\beta [\mathbf{D}\nabla u \cdot \hat{\mathbf{n}}]_{\mathbf{r}_{ij}(t_\alpha,t_\beta)} \left|\frac{\partial \mathbf{r}_{ij}}{\partial(\xi,\eta)}\right|_{(t_\alpha,t_\beta)} \tag{6}$$

and

$$\int\int\int_{V_{ij}} S\,\mathrm{d}V \approx \sum_{\alpha=1}^{2}\sum_{\beta=1}^{2}\sum_{\gamma=1}^{2} w_\alpha w_\beta w_\gamma [S]_{\mathbf{p}_{ij}(t_\alpha,t_\beta,t_\gamma)} \left|\frac{\partial \mathbf{p}_{ij}}{\partial(\xi,\eta,\zeta)}\right|_{(t_\alpha,t_\beta,t_\gamma)}, \tag{7}$$

where $\left|\frac{\partial \mathbf{r}_{ij}}{\partial(\xi,\eta)}\right|$ and $\left|\frac{\partial \mathbf{p}_{ij}}{\partial(\xi,\eta,\zeta)}\right|$ are the Jacobians of the transformations.

Note that Eqs. (4)–(7) require the evaluation of u and its gradient at specific quadrature points. In the next section we examine methods of interpolation for approximating these values, based only on the values $u_i,\ i = 1,\ldots,N$.

2.2. Interpolation

The method of shape functions can be used in a finite volume discretisation to approximate the required values of u and ∇u for use in (4)–(7) [38]. Over each tetrahedral element, the standard shape function interpolant

$$s(\mathbf{x}) = \sum_{j=1}^{4} u_j N_j(\mathbf{x}) \tag{8}$$

is fitted, resulting in a linear interpolation of the values of u at the element vertices. For the definitions of the shape functions N_j, see [7]. Note that in this case the gradient computed by means of this interpolant,

$$\nabla s(\mathbf{x}) = \sum_{j=1}^{4} u_j \nabla N_j(\mathbf{x}) \tag{9}$$

is constant throughout the element.

Using the finite element shape functions in this way leads to the so-called "control volume-finite element" discretisation, or CV-FE. Although widely used for its simplicity and efficiency, it can fail to achieve second-order accuracy in some cases, particularly for highly anisotropic problems [21,22].

To effectively model problems where the gradient may vary significantly over an element, techniques such as least squares and Green–Gauss reconstruction have been used [2,6,20,22,14,44]. In this work, we investigate an alternative method of interpolation, namely the method of radial basis functions. We will see that this method can be thought of as an extension of the shape function approach, and in fact can be applied on top of the basic CV-FE method, in effect acting as a "corrector" to the CV-FE solution.

The method of radial basis functions, or RBFs, is a method of scattered data interpolation over $\mathbb{R}^m$ that may use an arbitrary number of nodes $\mathbf{x}_j,\ j = 1,2,\ldots n$. An RBF interpolant through these nodes has the following form [37]:

$$s(\mathbf{x}) = \sum_{j=1}^{n} \lambda_j \phi(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{k=0}^{m} c_k q_k(\mathbf{x}), \tag{10}$$

where the function $\phi : \mathbb{R} \to \mathbb{R}$ is a radial basis function [37], and the q_k form the standard basis for the space of m-variate linear polynomials (in three dimensions, $q_0 = 1$, $q_1 = x$, $q_2 = y$, $q_3 = z$). Different choices of ϕ give rise to different RBF interpolants. Some popular choices are the cubic function $\phi(r) = r^3$, the thin plate spline $\phi(r) = r^2\log(r)$, the linear function $\phi(r) = r$ and the multiquadric $\phi(r) = \sqrt{c^2 + r^2}$, which incorporates the parameter c^2 [37]. It is shown in [37] that for these choices of ϕ, the RBF interpolant (10) is infinitely differentiable on any region that excludes the interpolation points. Thus, ∇s is given by

$$\nabla s = \sum_{j=1}^{n} \lambda_j \left(\frac{\mathbf{x}-\mathbf{x}_j}{\|\mathbf{x}-\mathbf{x}_j\|}\right)\phi'(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{k=1}^{m} c_k \mathbf{e}_k, \tag{11}$$

where $\mathbf{e}_k$ is the kth standard basis vector in $\mathbb{R}^m$.

The conditions imposed on the RBF interpolant in order to determine the coefficients λ_j and c_k are the interpolation constraints

$$s(\mathbf{x}_j) = u_j, \quad j = 1, 2, \ldots, n, \tag{12}$$

along with a set of orthogonality constraints

$$\sum_{j=1}^{n} \lambda_j q_k(\mathbf{x}_j) = 0, \quad k = 0, 1, \ldots, m \tag{13}$$

so that the coefficients can be determined by solving the following square matrix system [37]:

$$\begin{pmatrix} \boldsymbol{\Phi} & \mathbf{P} \\ \mathbf{P}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}, \tag{14}$$

where

$$\begin{aligned}
&(\boldsymbol{\Phi})_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad i = 1, \ldots, n, \ j = 1, \ldots, n \\
&(\mathbf{P})_{i,k+1} = q_k(\mathbf{x}_i), \quad i = 1, \ldots, n, \ k = 0, \ldots, m \\
&(\boldsymbol{\lambda})_i = \lambda_i, \quad i = 1, \ldots, n \\
&(\mathbf{c})_{k+1} = c_k, \quad k = 0, \ldots, m \\
&(\mathbf{f})_i = u_i, \quad i = 1, \ldots, n.
\end{aligned}$$

The necessity of solving (14) in order to compute the RBF coefficients makes it costly to fit a single interpolant covering the entire mesh. In fact, this is undesirable anyway, since doing so would result in a dense Jacobian matrix within the Newton–Krylov method (see Section 2.3). Instead, the method of RBFs is best applied locally, fitting one interpolant per element, incorporating a set of local nodes. This is entirely analogous to the way in which shape functions are applied.

Indeed, note that with $n = 4$ in (10), constraints (12) and (13) imply that $\lambda_1 = \lambda_2 = \cdots = \lambda_n = 0$, so that in this case the RBF interpolant reduces to the shape function interpolant (8). Thus, RBF interpolation can be thought of as an extension of shape function interpolation, or equivalently, linear shape functions in $\mathbb{R}^m$ are just special cases of RBFs where the number of nodes is equal to $m + 1$.

For simplicity, we also use a constant number of nodes in our RBF interpolations, though obviously this number is greater than 4. These n nodes are selected in an iterative fashion: by including the element's vertices, the vertices of its nearest neighbours, the vertices of their neighbours, and so on, until the required number of nodes has been selected. In this way, the points at which the interpolant is actually evaluated lie in the interior of the set of nodes, thereby alleviating a well-known problem of RBF interpolations: that of poor accuracy at points near the boundaries [10]. In practice, we find that the value $n = 20$ offers good accuracy without being too costly.

Finally, it is well known that the matrix system (14) tends to be very ill-conditioned [37]. We therefore employ the truncated singular value decomposition [15] to compute its solution. The overhead associated with each singular value decomposition is small, since each matrix involves only a small, local set of nodes. In addition, storing this decomposition allows for efficient processing of multiple right hand sides, as is required within a Jacobian-free Newton–Krylov method (see Section 2.3).

2.3. Solution of nonlinear system

The partial differential Eq. (1) is transformed into its finite volume discretised form through the discretisation process discussed in the previous sections. The result is a nonlinear system of equations

$$\mathbf{F}(\mathbf{u}) = 0 \tag{15}$$

for the finite volume solution vector $\mathbf{u} = (u_1, u_2, \ldots, u_N)^{\mathrm{T}}$. The basic inexact Newton method for solving (15) is the sequence of iterations [24]

$$\mathbf{u}^{(n+1)} = \mathbf{u}^{(n)} + \theta_n \delta\mathbf{u}^{(n)}, \tag{16}$$

where the Newton step $\delta\mathbf{u}^{(n)}$ is obtained iteratively to satisfy

$$\|\mathbf{F} + \mathbf{J}\delta\mathbf{u}^{(n)}\| \leqslant \eta_n \|\mathbf{F}\|. \tag{17}$$

Here $\mathbf{J} = \frac{\partial f_i}{\partial u_j}$ is the Jacobian matrix and η_n is the so-called forcing term [9], designed to avoid solving the linear model beyond the point where reductions are no longer realised in the nonlinear residual $\|\mathbf{F}\|$.

Whether shape functions or RBFs are employed in the evaluation of $\mathbf{F}$, the use of local nodes in these interpolations ensures that $\mathbf{J}$ will be sparse [30,29]. Krylov subspace methods are the favoured means for finding the solution of the linearised system satisfying (17) under such circumstances [26]. In this work, the GMRES method [40] is used. It utilises the Krylov subspace

$$\mathscr{K}_m = \text{span}\{\mathbf{r}_0, \mathbf{JM}\mathbf{r}_0, (\mathbf{JM})^2\mathbf{r}_0, \ldots, (\mathbf{JM})^{m-1}\mathbf{r}_0\}, \tag{18}$$

where $\mathbf{r}_0$ is the initial residual vector

$$\mathbf{r}_0 = \mathbf{F} + \mathbf{JM}\delta\mathbf{u}_0, \tag{19}$$

and the matrix $\mathbf{M}$ is a right preconditioner.

In the absence of preconditioning, Krylov subspace methods such as GMRES do not require the Jacobian matrix explicitly, but rather only its action on a vector, in the form of the Jacobian-vector product [26,41]. A Jacobian-free implementation of the method can therefore be derived, whereby Jacobian-vector products are approximated using the discrete, directional derivative [26]

$$\mathbf{Jv} \approx \frac{\mathbf{F}(\mathbf{u} + \varepsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\varepsilon}. \tag{20}$$

Computing the shifted vector $\mathbf{F}(\mathbf{u} + \varepsilon\mathbf{v})$ in (20) requires that the nodal values u_i be shifted by the amounts εv_i, and each element's interpolation recomputed. This can be processed efficiently even when RBFs are being used for local interpolations, by means of the saved singular value decompositions discussed in Section 2.2. Referring to (14), we see that changes to u_i affect only the vector $\mathbf{f}$ on the right hand side: the matrix itself depends only on the mesh geometry. Thus, by re-using the stored decompositions we can efficiently process each additional right hand side corresponding to shifted values of $\mathbf{F}$.

2.4. Combining integration and interpolation

In Section 2.1 two methods of numerical integration were presented: one based on the midpoint rule and the other on Gaussian quadrature. In Section 2.2, two methods for estimating the fluxes at integration points were presented: shape functions and radial basis functions. Now the strategy for combining these methods will be considered, such that an accurate solution to (1) is obtained without excessive computational cost.

Table 1 lists the pairings of the integration and interpolation methods used in this work. The midpoint rule integration and shape function interpolation comprise the CV-FE method, which offers higher speed but lower accuracy. The more accurate methods of Gaussian quadrature and radial basis functions are paired to provide higher accuracy at lower speed. For ease of discussion, this RBF-based discretisation method will be referred to as CV-RBF.

The method of CV-RBF is best thought of as a corrector, which is applied after the solution has already been obtained using CV-FE. Thus, as per Table 1, there are two stages involved in obtaining the final solution. Stage one uses CV-FE discretisation, which yields a first approximate solution, after which CV-RBF is applied to obtain the final solution. In this way, the advantages of both methods are realised: the CV-FE method offers a fast route to an approximate solution, and CV-RBF provides the necessary means to improve upon this solution and obtain the final, accurate solution.

Table 1
Optimal pairings of integration and interpolation: two stages of the solution process

Stage	Method	Speed	Accuracy	Integration	Interpolation
1	CV-FE	Higher	Lower	Midpoint rule	Shape functions
2	CV-RBF	Lower	Higher	Gaussian quadrature	Radial basis functions

We conclude this section by analysing the forms of the nonlinear component functions f_i of (15). In the case of CV-FE discretisation they take the form

$$f_i(\mathbf{u}) = \sum_{j=1}^{N_{f_i}} \mathbf{D}(s_i(\mathbf{m}_{ij}))\nabla s_i(\mathbf{m}_{ij}) \cdot \hat{\mathbf{n}} A(\sigma_{ij}) + S(u_i)\Delta V_i \tag{21}$$

while for CV-RBF, they take the form

$$\begin{aligned} f_i(\mathbf{u}) = &\sum_{j=1}^{N_{f_i}} \sum_{\alpha=1}^{2} \sum_{\beta=1}^{2} w_\alpha w_\beta \mathbf{D}(s_i(\mathbf{r}_{ij}(t_\alpha, t_\beta)))\nabla s_i(\mathbf{r}_{ij}(t_\alpha, t_\beta)) \cdot \hat{\mathbf{n}} \left| \frac{\partial \mathbf{r}_{ij}}{\partial(\xi, \eta)} \right|_{(t_\alpha, t_\beta)} \\ &+ \sum_{j=1}^{N_{v_i}} \sum_{\alpha=1}^{2} \sum_{\beta=1}^{2} \sum_{\gamma=1}^{2} w_\alpha w_\beta w_\gamma S(s_i(\mathbf{p}_{ij}(t_\alpha, t_\beta, t_\gamma))) \left| \frac{\partial \mathbf{p}_{ij}}{\partial(\xi, \eta, \zeta)} \right|_{(t_\alpha, t_\beta, t_\gamma)}. \end{aligned} \tag{22}$$

Because of the comparatively high cost of RBF interpolation, it is important to count precisely how many such evaluations are required in a single evaluation of (22). To this end, let E denote the number of elements in the mesh, and let F_b denote the number of element faces that lie on boundaries with non-Dirichlet boundary conditions.

As discussed in Section 2, each tetrahedral element is composed of four sub-control volumes. Each sub-control volume has three external faces (refer again to Fig. 1), but there are only ${}^4C_2 = 6$ distinct external faces per element. In addition to these six control volume faces, any element face that lies on a boundary contributes three more control volume faces through which fluxes must be computed. When using (22), each such computation requires evaluation of $\mathbf{D}(s_i)\nabla s_i$ at four Gauss points.

At first, it would appear that the evaluation of the three components of ∇s_i along with s_i itself would take four times as long as the evaluation of s_i alone. However, numerical experimentation shows that an efficient RBF implementation that re-uses the values of r, $\phi(r)$ and $\phi'(r)$ can compute s_i and ∇s_i in approximately twice the time it takes to evaluate s_i alone. Altogether then, the effective number of RBF evaluations is $2 \times 4 \times (6E + 3F_\mathrm{b})$, or $48E + 24F_\mathrm{b}$ RBF evaluations required to evaluate the first term of (22).

The second term of (22) requires eight evaluations of s_i per sub-control volume. As there are four sub-control volumes per element, this amounts to $8 \times 4E = 32E$ RBF evaluations altogether, making it the cheaper of the two terms to evaluate.

2.5. Preconditioning

A widely-recognised technique for preconditioning in a Jacobian-free Newton–Krylov method is to construct a preconditioner based on a Jacobian matrix from a simplified version of the problem [26]. The two-stage solution approach discussed in the previous section readily permits such a technique: the CV-FE Jacobian can be used as a cheap substitute for the CV-RBF Jacobian for the purposes of preconditioning.

In this work we use a sparse approximate inverse preconditioner based on Frobenius norm minimisation, as described in [39] and the references therein. A matrix $\mathbf{M}$ is constructed, that satisfies

$$\mathbf{M} = \min_{\mathbf{M}^* \in \mathscr{M}} \|\mathbf{I} - \mathbf{J}\mathbf{M}^*\|_F^2 \tag{23}$$

for the set of matrices $\mathscr{M}$ having the same sparsity pattern as the CV-RBF Jacobian matrix. Using the Frobenius norm is the key to this method, for it allows problem (23) to be decomposed into N independent least squares problems [39]:

$$\|\mathbf{I} - \mathbf{J}\mathbf{M}\|_F^2 = \sum_{j=1}^{N} \|\mathbf{e}_j - \mathbf{J}\mathbf{m}_j\|_2^2, \tag{24}$$

where $\mathbf{e}_j$ is the jth standard basis vector in $\mathbb{R}^N$, and $\mathbf{m}_j$ is the jth column of $\mathbf{M}$. Each vector $\mathbf{m}_j$ will have nonzero entries only where the Jacobian sparsity pattern dictates, so each problem (24) is solvable using standard direct decomposition methods.

In practice, the computation of each column of **M** can be too expensive when computed using the CV-RBF Jacobian. Instead, we use the much cheaper CV-FE Jacobian in (24). Using this hybrid approach, where the sparsity pattern is dictated by the CV-RBF Jacobian, and the values themselves by the CV-FE Jacobian, the resultant preconditioner has been found to be effective at preconditioning both stages of the two-stage solution process. This approach is also supported by the notion discussed in Section 2.2 that the method of shape functions can be considered to be a special case of the method of radial basis functions given in (10).

3. Results and discussion

In this section we test the effectiveness of the methods described in the previous sections by applying them to three test problems. In order to exactly measure the accuracy of the numerical solutions, we focus exclusively on problems that have known analytical solutions. Each test problem is based on the steady-state diffusion Eq. (1) on the unit cube. For simplicity in discussing and visualising the solutions, the problem is interpreted as a heat transfer problem. Different choices of **D** and S, along with different boundary conditions, give rise to the three different test problems.

3.1. Test problems

The first two test problems involve the linear, steady-state heat diffusion problem obtained by taking a constant, diagonal **D** along with $S = g_0$ (constant) in (1). The boundaries $x = 0$, $y = 0$ and $z = 0$ are insulated, while the boundaries $x = 1$, $y = 1$ and $z = 1$ are subject to Newtonian cooling with external temperature u_∞ and heat transfer coefficient h. The solution to this problem can found as per [32]:

$$u(x,y,z) = \sum_{n=1}^{\infty}\sum_{m=1}^{\infty}\sum_{k=1}^{\infty} \frac{\alpha(n,m,k)X(\mu_n,x)Y(\lambda_m,y)Z(\gamma_m,z)}{(D_{xx}\mu_n^2 + D_{yy}\lambda_m^2 + D_{zz}\gamma_k^2)N_x(\mu_n)N_y(\lambda_m)N_z(\gamma_k)}, \quad 0 \leqslant x \leqslant 1,\ \ 0 \leqslant y \leqslant 1,\ \ 0 \leqslant z \leqslant 1 \tag{25}$$

where the eigenfunctions are given by

$$X(\mu,x) = \mu\cos(\mu x), \quad Y(\lambda,y) = \lambda\cos(\lambda y), \quad Z(\gamma,z) = \gamma\cos(\gamma z),$$

the normalisation factors are given by

$$N_x(\mu) = \frac{\mu^2}{2}\left(1 + \frac{hD_{xx}}{h + \mu D_{xx}^2}\right),$$

with similar expressions for $N_y(\lambda)$ and $N_z(\gamma)$, and the eigenvalues are found by finding the roots of the transcendental equation

$$\tan(\mu) = \frac{h}{\mu D_{xx}},$$

with similar equations for λ and γ. Finally, $\alpha(n,m,k)$ is given by

$$\alpha(n,m,k) = \int_0^1\int_0^1\int_0^1 X(\mu_n,\xi)Y(\lambda_m,\eta)Z(\gamma_k,\zeta)g_0\,\mathrm{d}\zeta\,\mathrm{d}\eta\,\mathrm{d}\xi.$$

The parameter values used for this problem are listed in Table 2. Note that Test Problem 1 uses $\mathbf{D} = \mathrm{diag}(5,5,5)$, while Test Problem 2 uses $\mathbf{D} = \mathrm{diag}(5,5000,5000)$, with a view to providing an increasing challenge to the numerical solution methods.

Three-dimensional visualisations of the analytic solution for the parameters listed in Table 2 are shown in Fig. 4. Several isosurfaces have been drawn to help illustrate the behaviour of the solution. Fig. 4(a) illustrates the symmetric nature of Test Problem 1, with its equal diffusivities, and Fig. 4(b) the near one-dimensional nature of Test Problem 2, with the diffusion in the y and z directions dominant over that in the x direction.

The third test problem is based on a problem given in [24] and uses nonlinear diffusivity $D_{xx} = D_{yy} = D_{zz} = u^{1.3}$ in (1). The source term is constructed by substituting the imposed solution

Table 2
Physical parameters for Test Problems 1 and 2

Parameter	Description	Value
D_{xx}	Thermal diffusivity in x direction	5 $m^2 s^{-1}$
D_{yy}	Thermal diffusivity in y direction	5 $m^2 s^{-1}$ (Test Problem 1) 5000 $m^2 s^{-1}$ (Test Problem 2)
D_{zz}	Thermal diffusivity in z direction	5 $m^2 s^{-1}$ (Test Problem 1) 5000 $m^2 s^{-1}$ (Test Problem 2)
g_0	Source	10 K $m^{-2} s^{-1}$
h	Heat transfer coefficient	2 W m^{-2} K^{-1}
u_∞	External temperature	20 K

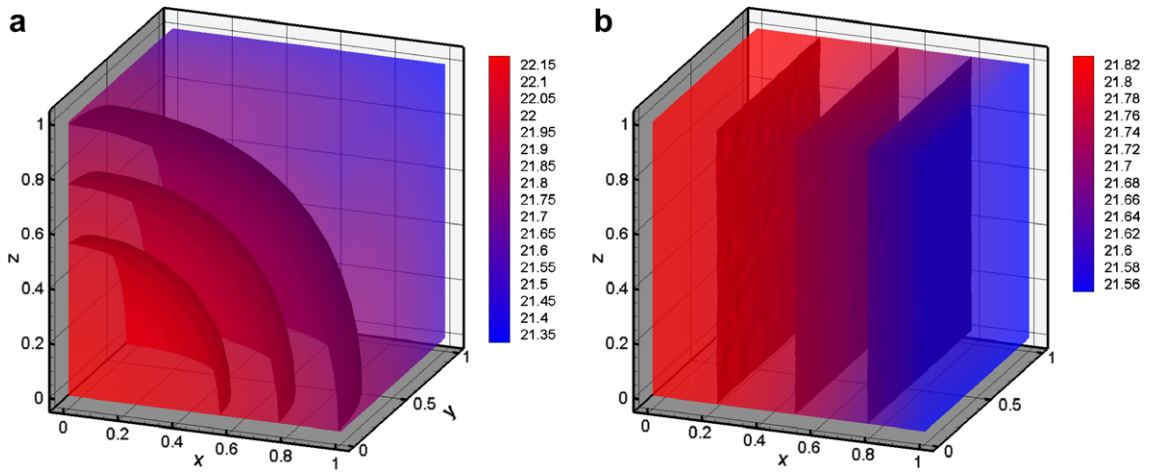

Fig. 4. Analytic solutions of: (a) Test Problem 1; (b) Test Problem 2.

$$u(x,y,z) = 10xyz(1-x)(1-y)(1-z)e^{-(x^2+y^2+z^2)}, \quad 0 \leqslant x \leqslant 1, \quad 0 \leqslant y \leqslant 1, \ 0 \leqslant z \leqslant 1 \tag{26}$$

into (1). On all boundaries, the Dirichlet condition $u = 0$ is prescribed.

The zero condition on the boundaries makes three-dimensional visualisation of this solution awkward. Fig. 5 depicts a single contour of the solution at $z = 0.5$. The physical nature of the solution is that of a hot interior with maximum temperature near the point $(0.4, 0.4, 0.4)$, with the temperature cooling further away from this point and zero on the boundaries.

3.2. Meshes

Two examples of the unstructured, tetrahedral meshes used for these test problems are exhibited in Fig. 6. They were generated using the mesh generator *Gmsh* [13] over a unit cube. A uniform edge length was requested throughout the mesh, but naturally as these are unstructured meshes there is some variation in edge lengths throughout the meshes.

3.3. Comparative accuracy of solutions

For the results presented in this section, the unstructured mesh comprising 1567 nodes and 7081 elements illustrated in Fig. 6(a) was used to solve each test problem. In computing the solution errors, the error measure $\frac{\|\mathbf{u}^{(e)}-\mathbf{u}^{(a)}\|_2}{\|\mathbf{u}^{(e)}\|_2}$ was used, where superscript (e) symbolises the exact, analytic solution and superscript (a) the approximate, numerical solution.

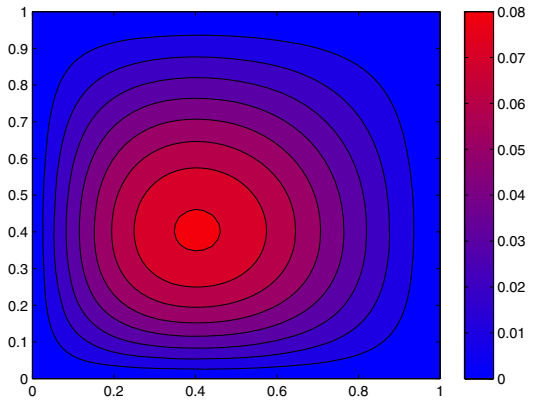

Fig. 5. Analytic solution of Test Problem 3. Contour at $z = 0.5$.

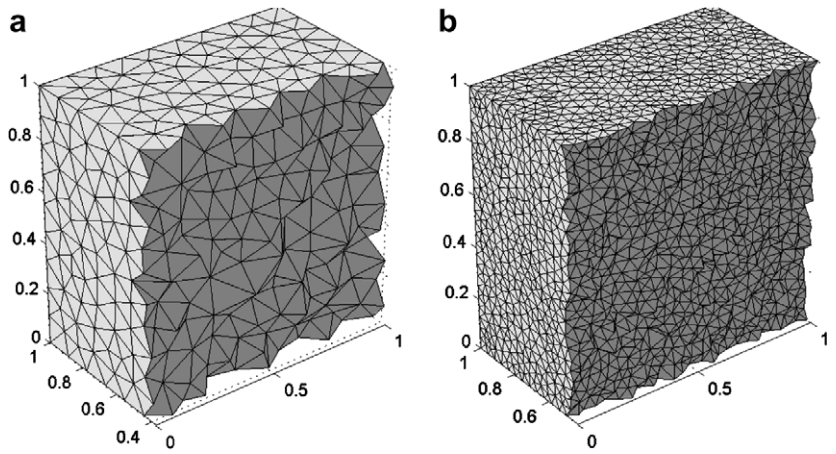

Fig. 6. Unstructured tetrahedral meshes with: (a) 7081 elements; (b) 103,008 elements. Visualisations were produced using *DistMesh* [36].

Table 3
Error in solution for Test Problems 1, 2 and 3

	Test Problem 1	Test Problem 2	Test Problem 3
CV-FE	4.40E − 05	9.33E − 04	2.90E − 02
CV-RBF	1.30E − 06	5.81E − 05	6.55E − 04

The multiquadric radial basis function was used throughout, for its high accuracy, as observed in previous work such as [11,17,29]. The "default" value of its parameter, $c^2 = 1$, was found by numerical experiment to be acceptable. Mindful of the computational costs associated with using too many nodes per interpolation, we chose to use $n = 20$ nodes for every element's RBF interpolation.

The results of solving each test problem using both CV-FE and CV-RBF are given in Table 3. We see that in each case the CV-RBF solution is between one and two orders of magnitude more accurate than that achieved with CV-FE. The best accuracy is achieved for both methods on the isotropic Test Problem 1. Test Problem 2, with its much stronger diffusion in the y and z directions, poses more difficulty, and the accuracy of both methods is reduced. For the nonlinear Test Problem 3, the accuracy is further reduced, however the improvement offered by CV-RBF over CV-FE is greatest for this problem.

The contour plots in Fig. 7 illustrate the two numerical solutions for Test Problem 2. The contours are shown in the plane $z = 0.5$ and the analytic solution contours are also drawn for comparison. The CV-FE

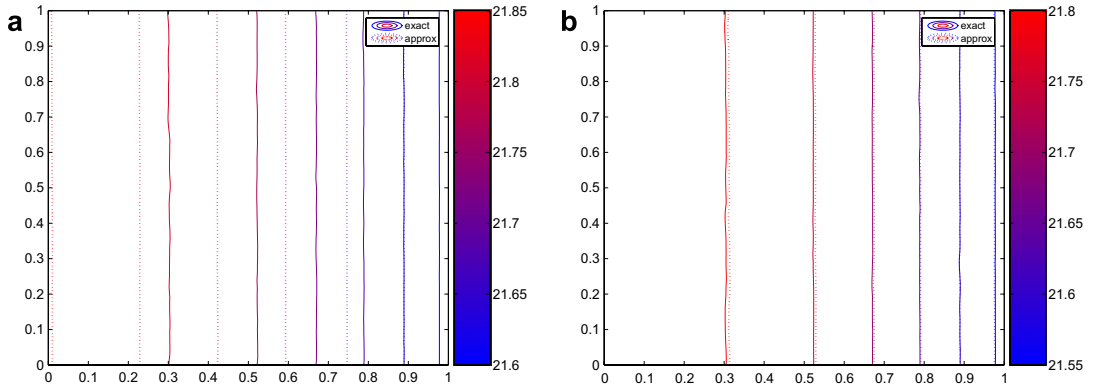

Fig. 7. Solution contours at $z = 0.5$ for Test Problem 1 with $\mathbf{D} = \mathrm{diag}(5, 5000, 5000)$: (a) CV-FE; (b) CV-RBF.

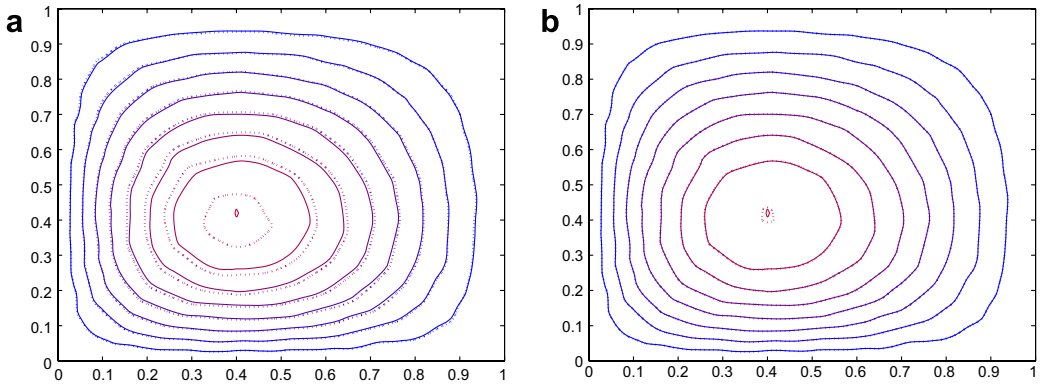

Fig. 8. Solution contours at $z = 0.5$ for Test Problem 2: (a) CV-FE; (b) CV-RBF.

solution shown in Fig. 7(a) has failed to reproduce the correct temperature variations in the x direction. This failure is explained by the large anisotropic ratio present in this problem: the diffusivities in the y and z directions are one thousand times greater than the diffusivity in the x direction, and the CV-FE method has failed to adequately capture the much subtler diffusion in the x direction.

Using CV-RBF results in the much better solution shown in Fig. 7(b). The more accurate discretisation has, by and large, captured the correct temperature variation in the x direction. The approximate solution contours are now just slightly offset from their correct positions, and there are none of the spurious or missing contours that were evident in Fig. 7(a).

In Fig. 8 the solution contours obtained for Test Problem 3 are shown. From Fig. 8(a) it is evident that the main weakness of the CV-FE numerical solution is its failure to adequately capture the correct behaviour in the interior, where the temperature is highest and the diffusion is strongest. In Fig. 8(b) the corresponding CV-RBF solution is almost a perfect visual match for the analytic solution. We note that the apparent oscillations in the solution contours are an artefact of the visualisation process and are not present in the actual solutions.

3.4. Estimated order of methods

In this section we estimate and compare the orders of accuracy achieved by CV-FE and CV-RBF by plotting the error in the solution against the average control volume face edge length and observing how this error decreases as the mesh is refined. As we are using unstructured meshes we expect some degree of variation in the results, caused by the variation in edge lengths present in each mesh.

The error measurement used for these tests is derived from the norm

$$\left\|\mathbf{u}^{(e)} - \mathbf{u}^{(a)}\right\|_p, \tag{27}$$

where as before superscript (e) symbolises the exact, analytic solution and superscript (a) the approximate, numerical solution, and p is 1, 2, or ∞.

As the mesh is refined, the number of nodes in the mesh, N, increases. For $p = 1$ and $p = 2$, the error measurement (27) involves a summation over N, which must be compensated for if the measurements are to be legitimately compared. (The infinity norm is immune to this problem, as it is simply the maximum error in the mesh.) The appropriate scaling factors, based on the definitions of the one and two-norms, are $\frac{1}{N}$ and $\frac{1}{\sqrt{N}}$ respectively. Thus, the final, comparable measures, are:

$$\text{error} = \frac{1}{N}\sum_{i=1}^{N}\left|u_i^{(e)} - u_i^{(a)}\right|, \tag{28}$$

$$\text{error} = \frac{1}{\sqrt{N}}\sqrt{\sum_{i=1}^{N}\left(u_i^{(e)} - u_i^{(a)}\right)^2} \tag{29}$$

and

$$\text{error} = \max_{i=1\ldots N}\left|u_i^{(e)} - u_i^{(a)}\right| \tag{30}$$

for the one, two and infinity-norms, respectively.

For each test problem, the solution is computed using CV-FE discretisation and using CV-RBF discretisation with $n = 20$ nodes per interpolation. The order, q, of each method is estimated by fitting the curve

$$\text{error} = \text{constant} \times h^q \tag{31}$$

using logarithmic regression on the (h, error) ordered pairs, where h is the average control volume face edge length, and error is measured as per Eqs. (28)–(30).

Fig. 9 shows logarithmic plots of these ordered pairs, along with the fitted curves (31), which appear linear on the log-log scale. The plots show that in all cases there is a reduction in solution error as the mesh is refined. Furthermore, there is a close agreement between the observed errors and the fitted curves for all three test problems and all six error measurements.

Table 4 lists the estimated orders of accuracy computed for each test problem, under the three different error measures, for both CV-FE and CV-RBF. The observed orders are largely consistent across all three error measures, with perhaps only slight disagreement in the infinity-norm against the other two measures for the CV-RBF method. This disparity could be attributed to the greater spread of infinity-norm values (marked with stars in Fig. 9) as compared to the other measures, resulting in a less accurate regression fit.

The order of the CV-FE method is consistently estimated across all three error measures. Of particular note is the CV-FE method's failure to achieve second order accuracy for Test Problem 2. This result is consistent with those reported elsewhere [21,22], where the CV-FE method fails to accurately model problems with very high anisotropic ratios. The CV-RBF method has no such difficulty for this problem, and in fact it records its highest order of accuracy in this case, estimated as fourth order. For Test Problems 1 and 3, the CV-FE method does exhibit second order accuracy, while the CV-RBF method is consistently measured as having between third and fourth order accuracy.

3.5. Linear and nonlinear iterations

In Section 2.4, it was discussed how the CV-RBF method was best applied as part of a two-stage solution process, with CV-FE iterations comprising the first stage. In this section, we investigate how this approach can reduce the number of CV-RBF iterations required for convergence. We also investigate the effect of the preconditioner discussed in Section 2.5, whereby combined CV-FE/CV-RBF information is used to construct a preconditioner to use throughout both stages of the solution process.

For these tests, we examine the number of CV-FE and CV-RBF iterations that are required to solve Test Problem 3 on the mesh depicted in Fig. 6(a). It should be noted that with $n = 20$ nodes per interpolation, each

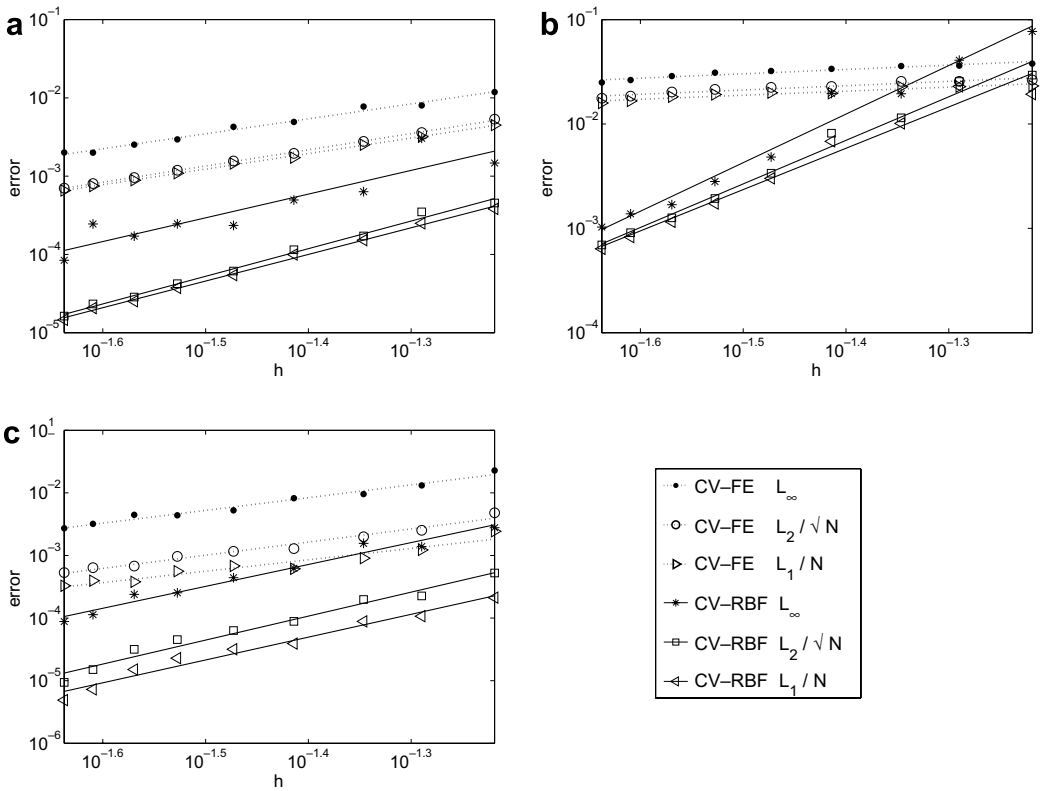

Fig. 9. Six kinds of error versus average control volume edge length (h) for: (a) Test Problem 1; (b) Test Problem 2; (c) Test Problem 3. Markers show CV-FE and CV-RBF errors, using scaled one-norm, scaled two-norm and infinity-norm measures.

Table 4
Estimated orders of accuracy for solving Test Problems 1, 2 and 3

Problem	Method	L_1/N	$L_2/\sqrt{N}$	L_∞
Test Problem 1	CV-FE	2.0	2.1	1.9
	CV-RBF	3.4	3.5	3.0
Test Problem 2	CV-FE	0.4	0.4	0.4
	CV-RBF	4.0	4.2	4.7
Test Problem 3	CV-FE	1.8	2.1	2.0
	CV-RBF	3.6	3.8	3.5

CV-RBF iteration can be an order of magnitude slower than each CV-FE iteration. Thus any approach that can achieve a reduction in the number of CV-RBF iterations is to be highly regarded.

Fig. 10 depicts the convergence curves for Test Problem 3 when solved using the two-stage method with, and without, preconditioning. Several features are apparent. First, the use of a forcing term (Section 2.3) has not completely eliminated oversolving from the process, but oversolving is mostly confined to the much cheaper CV-FE iterations. Second, the switch from CV-FE to CV-RBF is accompanied by a sudden increase in the residual norm, as the final CV-FE solution is re-assessed under the more accurate CV-RBF discretisation.

Third, by comparing Fig. 10(a) and (b), it is evident that the preconditioner has been effective in reducing the number of CV-FE and CV-RBF iterations required for convergence. For this nonlinear problem, the preconditioner was computed once, as per Section 2.5, at the beginning of the solution process and held constant throughout. Its effect was to more than halve the total number of iterations required, but more important, to reduce the number of CV-RBF iterations by almost a factor of four.

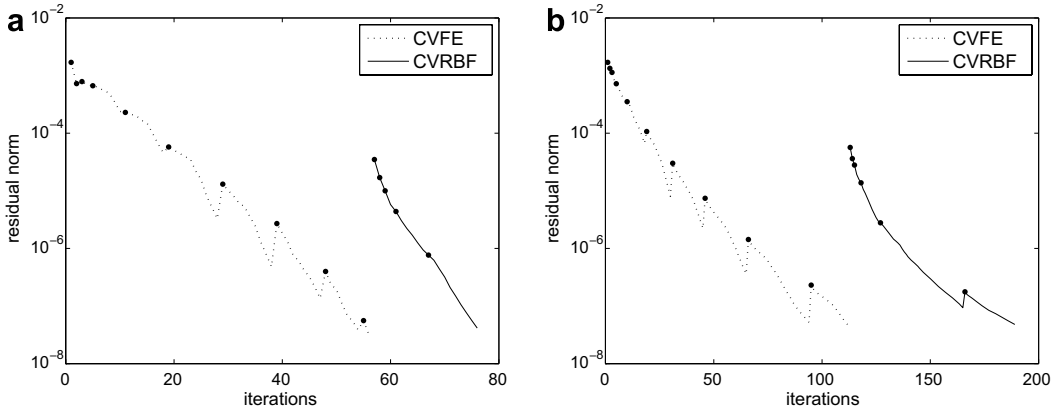

Fig. 10. GMRES residual versus number of iterations for Test Problem 3 with: (a) preconditioning; (b) no preconditioning. Dots signify the beginning of each nonlinear iteration.

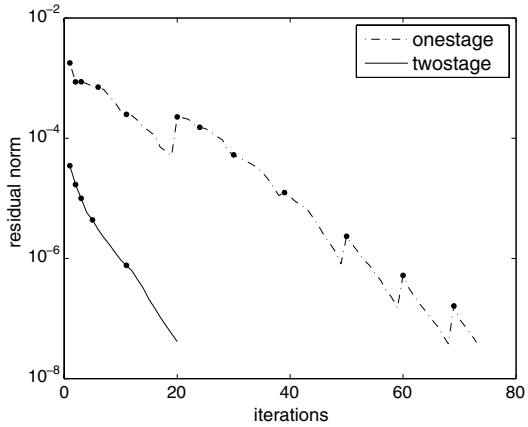

Fig. 11. GMRES residual versus number of CV-RBF iterations for Test Problem 2. The one-stage method solves the entire problem, whereas the two-stage method solves only from the CV-FE solution.

What is not apparent from Fig. 10 is whether the two-stage process itself contributes to reducing the number of CV-RBF iterations. In fact, the jump in the residual norm at the transition from stage one to stage two perhaps suggests that the impact of the CV-FE iterations is minimal. However, Fig. 11 shows that this is not the case.

Fig. 11 compares the number of CV-RBF iterations required for convergence, when used as either a one-stage or two-stage process. The two-stage curve is as per Fig. 10(a), whereas the one-stage curve corresponds to solving the entire problem using only CV-RBF, without the benefit of the initial CV-FE iterations. The figure shows how it takes between three and four times as many CV-RBF iterations to solve this problem when the initial CV-FE stage is not employed.

In the final section of these numerical experiments we compare the total running time of a CV-FE-only method, to the two-stage CV-FE/CV-RBF method, such that they obtain comparable accuracy for a given problem.

3.6. Comparative resource requirements

In previous sections it has been demonstrated that the accuracy achieved by the two-stage CV-FE/CV-RBF solution method is superior to that achieved by the CV-FE method alone. In producing this additional accu-

Table 5
Mesh sizes for the CV-FE method to achieve comparable accuracy to the CV-RBF method for Test Problem 3

Method	Nodes	Elements
CV-RBF	1567	7081
CV-FE	19,367	103,008

Table 6
Relative resource requirements for the CV-FE and CV-RBF methods (comparable accuracy) for Test Problem 3

Method	Time	Memory
CV-RBF	1.0	1.0
CV-FE	7.4	9.0

racy however, the resource requirements are much more demanding. These additional resource requirements must be taken into account when deciding if the method is beneficial overall.

An important question then, is whether the extra accuracy offered by the CV-RBF correction applied in stage two compensates for the extra resources required to compute it. In this section, we examine how the two-stage method fares in terms of efficiency, by comparing its resource requirements to those of a CV-FE-only method over a refined mesh, such that the accuracy achieved by the two methods is equivalent.

In Section 3.3 we found that the numerical solution to Test Problem 3 was more than an order of magnitude more accurate when using CV-RBF. The mesh used for these tests was the 1567 node, 7081 element mesh illustrated in Fig. 6(a). By successively refining this mesh, and solving using only CV-FE, a solution of comparable accuracy was eventually achieved, using the 19,367 node and 103,008 element mesh illustrated in Fig. 6(b). Table 5 summarises these results.

To determine the relative costs of the two methods in achieving this solution, we measure the runtime and memory usage of the CV-FE method over the fine mesh, as compared to the CV-RBF method over the coarse mesh. Table 6 lists the relative time and memory requirements, using the CV-RBF method over the coarse mesh as a baseline. We see that the use of a fine mesh with CV-FE has resulted in a seven-fold increase in runtime and a nine-fold increase in memory consumption. Thus for this problem, the use of CV-RBF over a coarse mesh offers a significant saving when it comes to generating a solution of the required accuracy.

4. Conclusions

A new three-dimensional finite volume discretisation method has been presented, incorporating radial basis functions as a means of interpolation, along with Gaussian quadrature as a means of integration. Each RBF interpolant incorporates only a local set of nodes, thereby ensuring it can be processed efficiently, and will not destroy the sparsity of the Jacobian matrix. Truncated singular value decompositions are used to solve each RBF matrix system, and to efficiently process additional right hand sides corresponding to the shifted function values required for the inexact Newton–Krylov solver.

A Jacobian-free Newton–Krylov method is implemented to solve the resultant nonlinear system of equations. It is recognised that a two-stage solution process is beneficial, whereby traditional CV-FE discretisation is used initially to solve the problem, before the solution is corrected using the more accurate CV-RBF discretisation. It is also recognised that the method of shape functions is actually a special case of RBFs, obtained when the number of nodes used is 4 (in three dimensions). This fact is exploited as part of the two-stage solution process, whereby the CV-FE Jacobian matrix is used in computing an effective preconditioner for use in the CV-RBF iterations.

Numerical experiments conducted on both linear and nonlinear test problems gauge the accuracy and efficiency of the new method. The method is found to consistently achieve higher accuracy and higher order than the traditional method based on shape functions. It is also shown that CV-RBF is more efficient at obtaining a solution of given accuracy, compared to using CV-FE on a fine mesh.

Future research in this area will see these ideas extended to allow the solution of advection–diffusion problems. In problems where the advective process dominates, the interaction of the method with techniques such as upwinding and flux limiting, which are typically used to overcome nonphysical oscillations in the solution, will be investigated.

References

[1] C. Bailey, G.A. Taylor, M. Cross, P. Chow, Discretisation procedures for multi-physics phenomena, J. Comput. Appl. Math. 103 (1999) 3–17.
[2] T. Barth, Aspects of unstructured grids and finite volume solvers for the Euler and Navier–Stokes equations, 1994.
[3] S. Bellavia, B. Morini, A globally convergent Newton-GMRES subspace method for systems of nonlinear equations, SIAM J. Sci. Comput. 23 (2) (2001) 940–960.
[4] P.N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Statist. Comput. 11 (3) (1990) 450–481.
[5] P. Chow, M. Cross, K. Pericleous, A natural extension of the conventional finite volume method into polygonal unstructured meshes for CFD application, Appl. Math. Model. 20 (1996) 170–183.
[6] N. Croft, Unstructured mesh – finite volume algorithms for swirling, turbulent, reacting flows, PhD Thesis, University of Greenwish, London, UK, 1998.
[7] G. Dhatt, G. Touzot, The finite element method displayed translation of: une présentation de la méthode des éléments finis, Wiley-Inerscience, 1984.
[8] S.C. Eisenstat, H.F. Walker, Globally convergent inexact Newton methods, SIAM J. Optim. 4 (2) (1994) 393–422.
[9] S.C. Eisenstat, H.F. Walker, Choosing the forcing terms in an inexact Newton method, SIAM J. Sci. Comput. 17 (1) (1996) 16–32.
[10] B. Fornberg, T.A. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, Comput. Math. Appl. 43 (3–5) (2002) 473–490.
[11] R. Franke, Scattered data interpolation: tests of some methods, Math. Comput. 38 (1982) 181–200.
[12] C. Franke, R. Schaback, Solving partial differential equations by collocation using radial basis functions, Appl. Math. Comput. 93 (1) (1998) 73–82.
[13] Christophe Geuzaine, Jean-François Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities.
[14] F. Carl, Ollivier Gooch, Michael Van Altena, A high-order accurate unstructured mesh finite-volume scheme for the advection–diffusion equation, J. Comput. Phys. 181 (2) (2002) 729–752.
[15] Per Christian Hansen, Truncated singular value decomposition solutions to discrete ill-posed problems with ill-determined numerical rank, SIAM J. Sci. Statist. Comput. 11 (3) (1990) 503–518.
[16] R.L. Hardy, Multiquadric equations of topography and other irregular surfaces, J. Geophys. Res. 76 (1971) 1905–1915.
[17] R.L. Hardy, Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968–1988, Comput. Math. Appl. 19 (8–9) (1990) 105–210.
[18] F. Hermeline, A finite volume method for the approximation of diffusion operators on distorted meshes, J. Comput. Phys. 160 (2000) 481–499.
[19] Y.C. Hon, K.F. Cheung, X.Z. Mao, E.J. Kansa, A multiquadric solution for shallow water equation, ASCE J. Hydraul. Eng. 125 (5) (1999) 524–533.
[20] H. Jasak, A.D. Gosman, Automatic resolution control for the finite-volume method, part 1: a-posteriori error estimates, Numer. Heat Transf. Part B: Fund. 38 (2000) 237–256.
[21] P.A. Jayantha, I.W. Turner, A second order finite volume technique for simulating transport in anisotropic media, Int. J. Numer. Methods Head Fluid Flow 13 (1) (2003) 31–56.
[22] P.A. Jayantha, I.W. Turner, A second order control-volume finite-element least-squares strategy for simulating diffusion in strongly anisotropic media, J. Comput. Math. 13 (1) (2005) 31–56.
[23] E.J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid dynamics, Comput. Math. Appl. 19 (8/9) (1990) 127–161.
[24] C.T. Kelly, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.
[25] C.T. Kelly, Solving Nonlinear Equations with Newton's Method, SIAM, 2003.
[26] D.A. Knoll, D.E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comp. Phys. 193 (2004) 357–397.
[27] J. Li, Y. Chen, D. Pepper, Radial basis function method for 1-D and 2-D groundwater contaminant transport modeling, Comput. Mech. 32 (1–2) (2003) 10–15.
[28] F. Liu, I.W. Turner, V. Anh, N. Su, A two-dimensional finite volume method for transient simulation of time- and scale-dependent transport in heterogeneous aquifer systems, J. Appl. Math. Comput. 11 (1–2) (2003) 215–241.
[29] T.J. Moroney, I.W. Turner, A finite volume method based on radial basis functions for two-dimensional nonlinear diffusion equations, Appl. Math. Modell. 30 (10) (2006) 1118–1133.
[30] T.J. Moroney, An investigation of a finite volume method incorporating radial basis functions for simulating nonlinear transport, PhD Thesis, School of Mathematical Sciences, Queensland University of Technology, 2006.
[31] J.Y. Murthy, S.R. Mathur, Computation of anisotropic conduction using unstructured meshes, J. Heat Transf. 120 (1998) 583–591.

[32] M.N. Özişik, Heat Conduction, Wiley, 1980.
[33] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, 1980.
[34] M. Pernice, H.F. Walker, NITSOL: A Newton iterative solver for nonlinear systems, SIAM J. Sci. Comput. 19 (1) (1998) 302–318.
[35] P. Perré, I.W. Turner, A heterogeneous wood drying computational model that accounts for material property variation across growth rings, Chem. Eng. J. 86 (2002) 117–131.
[36] P.-O. Persson, G. Strang, A simple mesh generator in MATLAB, SIAM Rev. 46 (2) (2004) 329–345.
[37] M.J.D. Powell, The theory of radial basis function approximations in 1990, in: Advances in Numerical Analysis, Wavelets, Subdicision Algorithms and Radial Basis Functions, vol. II, Oxford University Press, 1990, pp. 105–210.
[38] C. Prakash, S.V. Patankar, A control volume-based finite-element method for solving the Navier–Stokes equations using equal-order velocity–pressure interpolation, Numer. Heat Transf. 8 (1985) 259–280.
[39] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS, 1996.
[40] Y. Saad, M.H. Shultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (3) (1996) 856–869.
[41] Yousef Saad, Henk A. van der Vorst, Iterative solution of linear systems in the 20th century, J. Comput. Appl. Math. 123 (1–2) (2000) 1–33.
[42] S. Song, M. Chen, Third order accurate large-particle finite volume method on unstructured triangular meshes, SIAM J. Sci. Comput. 23 (5) (2002) 1456–1463.
[43] A. Teskeredžić, I. Demirdžić, S. Muzaferija, Numerical method for heat transfer, fluid flow and stress analysis in phase-change problems, Numer. Heat Transf. Part B 42 (5) (2002) 437–459.
[44] S.L. Truscott, I.W. Turner, An investigation of the accuracy of the control volume finite element method based on triangular prismatic elements for simulating diffusion in anisotropic media, Numer. Heat Transf., Part B: Fund. 46 (3) (2004) 243–268.