INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng. 2000; **48**:19–35

Explicit and implicit meshless methods for linear advection–diffusion-type partial differential equations

M. Zerroukat[1,*,†,‡], K. Djidjeli[2] and A. Charafi[3]

[1] *Wessex Institute of Technology, Ashurst Lodge, Ashurst, Southampton SO40 7AA, U.K.*
[2] *Department of Ship Science, University of Southampton, Highfield, Southampton SO17 1BJ, U.K.*
[3] *School of Computer Science and Mathematics, University of Portsmouth, Portsmouth PO1 2EG, U.K.*

SUMMARY

Simple, mesh/grid free, explicit and implicit numerical schemes for the solution of linear advection–diffusion problems is developed and validated herein. Unlike the mesh or grid-based methods, these schemes use well distributed quasi-random points and approximate the solution using global radial basis functions. The schemes can be seen as generalized finite differences with random points instead of a regular grid system. This allows the computation of problems with complex-shaped boundaries in higher dimensions with no need for complex mesh/grid structure and with no extra implementation difficulties. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: meshless methods; collocation; radial basis functions; random points; advection–diffusion; partial differential equations

1. INTRODUCTION

As computer capabilities kept improving throughout the last three decades, it became possible to solve more and more complex problems. The increase of possibilities for solving such complex physical end engineering problems has also been due to advances in numerical methods and the development of efficient algorithms. For instance, it became possible to simulate large-scale problems such as fluid flow around ships and aircrafts, meteorology, turbulence and wide range of computer intensive problems. Most of engineering and physical problems are solved using Finite Differences Methods (FDM), Finite Elements Methods (FEM), Control Volume Methods (CVM) or Boundary Element Methods (BEM). These numerical methods are all based on a mesh/grid discretization that has to be generated in advance or dynamically modified as the solution progresses (adaptive meshing). It is widely acknowledged that mesh generation remains one of the biggest challenges in mesh-based methods. Given enough computer power, even the most computationally

*Correspondence to: M. Zerroukat, Wessex Institute of Technology, Ashurst Lodge, Ashurst, Southampton SO40 7AA, U.K.
†E-mail: zerr@wessex.ac.uk.
‡Current address: NWP Division, Meteorological Office, London Road, Bracknell RG12 2SZ, U.K.

Received 26 November 1998
Revised 29 April 1999

intensive problems, such as 3-D Navier–Stokes equations, can be computed accurately providing an acceptable mesh is found. The generation of suitable mesh for such complex problems remains the major task and can take most of the computational time and effort than the solution of the governing partial differential equations themselves. Not only mesh-based methods can be very complex, for many problems, in particular for those with moving boundaries or large deformations (e.g. crack propagation, phase-change, explosion of stars etc.), they are acknowledged not to be cost-effective for such dynamic processes.

During recent years, considerable effort has been devoted to the development of so-called mesh-free methods, which are also referred to in the literature as meshless, element-free, gridless or cloud methods. The aim of meshless methods is to eliminate at least the structure of the mesh and approximate the solution entirely using the nodes/points as a quasi-random set of points rather than nodes of an element/grid-based discretization [1, 2]. Amongst early attempts, one can mention the generalized finite differences on an irregular (arbitrary) grid system [3–6]. Another type of meshless named Smooth Particle Hydrodynamics (SPH), which uses a set of disordered particles and has been used in computational astrophysics to model collision and explosion of stars [7–9]. This method is well suited to these kind of problems with rapidly expanding computational domains. Another class of these methods is the so-called Diffuse Element (DE) method, where only a set of nodes and a boundary description is needed to derive the Galerkin equation. The interpolation functions are polynomials associated with nodal values by weighted least-squares approximation. Although no underlying structure for the nodes is required, an additional auxiliary grid is used to numerically compute the integral resulting from the Galerkin approach [10]. Belytschko *et al.* [11, 12] extended the DE method by providing additional terms in the derivatives and a regular cell structure for computing the integrals by means of higher order quadratures. This method is usually referred to as the Element-Free Galerkin (EFG) method. This method can also be seen as a subclass of the so-called Partition of Unity (PU) methods [13]. Liu *et al.* [14, 15] proposed a different type of gridless based on reproducing kernels and wavelet analysis (Reproducing Kernel Particle (RKP) method). For a comparative study on these methods, i.e. RKP, SPH, DE, EFG, see Reference [16] for details.

For many years, radial basis functions have been synonymous with scattered data approximation, especially in higher dimensions. However, recently there has been an increased interest in their use in solving partial differential equations (PDEs). This approach, which approximates the whole solution of the PDE directly using radial basis functions, is very attractive due to the fact that this method is a truly mesh or grid free technique. Since the original work of Kansa [17], a series of papers have appeared in this topic, see for instance Dubal *et al.* [18], Moridis and Kansa [19] and Sharan *et al.* [20]. Despite their excellent results, earlier works related to the application of radial basis functions for the numerical solution of PDEs have been based on intuition rather than a formal mathematical analysis. However, since the paper of Kansa [17], substantial theoretical advances have been established. For instance, Franke and Schaback [21, 22] were able to give a convergence proof and error bounds of the numerical approach, at least for partial differential equations with constant coefficients. For a substantial coverage, see Golberg and Chen [23] and Fasshauer [24], for an extensive bibliography on the subject.

This paper presents numerical schemes to solve the advection–diffusion equation on an arbitrary/random collocation points system, by approximating directly the solution using global radial basis functions. The schemes are similar to finite differences but with the advantage of arbitrary point locations. Due to the radial nature of the basis functions used, the schemes also make no distinction regarding the dimension of the problem.

2. RADIAL BASIS FUNCTIONS INTERPOLATION

The approximation of a function $u(\mathbf{x})$, using radial basis functions, may be written as a linear combination of N radial functions, viz.,

$$u(\mathbf{x}) \simeq \sum_{j=1}^{N} \lambda_j \varphi(\mathbf{x}, \mathbf{x}_j) + \psi(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega \subset \mathbf{R}^d \tag{1}$$

where N is the number of data points, $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ is the vector position, d is the dimension of the problem, λ's are coefficients to be determined and φ is the radial basis function. Equation (1) can be written without the additional polynomial ψ, see Reference [25] for details.

In a comparative study, Franke [26] compares an extensive number of techniques for interpolation/approximation on a number of tests. The study found that overall the MultiQuadrics (MQ) and Thin Plate Splines (TPS) Radial Basis Functions (RBF) are the most accurate techniques for scattered data approximation. However, the accuracy of MQ depends on a shape parameter, for which there is no mathematical theory yet as to how to choose its optimal value. Hence most applications of MQ use experimental tuning parameters or expensive optimization techniques to evaluate the optimum shape parameter [27]. In this paper, the TPS is used, as the radial basis function for the schemes in Section 3, as it combines good accuracy without the additional burden of computing a shape parameter. Furthermore, TPS is based on sound mathematical theory [28], whereas for MQ, although it works well, its construction theory is yet to be established [29].

An mth-order TPS is defined as

$$\varphi(\mathbf{x}, \mathbf{x}_j) = \varphi(r_j) = r_j^{2m} \log(r_j), \quad m = 1, 2, 3, \ldots, \tag{2}$$

where $r_j = \|\mathbf{x} - \mathbf{x}_j\|$ is the Euclidean norm. Since φ given by (2) is C^{2m-1} continuous, a higher-order thin plate splines must be used, for higher-order partial differential operators. For the problem in Section 3, i.e. advection–diffusion PDEs (Second order), an $m = 2$ is used (i.e. second-order thin plate splines) to guarantee at least C^2 continuity for u.

If $\mathscr{P}_q^d$ denotes the space of d-variate polynomials of order not exceeding q, and letting the polynomials $P_1, \ldots, P_m$ be the basis of $\mathscr{P}_q^d$ in $\mathbf{R}^d$, then the polynomial $\psi(\mathbf{x})$, in Equation (1), is usually written in the following form:

$$\psi(\mathbf{x}) = \sum_{i=1}^{m} \zeta_i P_i(\mathbf{x}) \tag{3}$$

where $m = (q - 1 + d)!/(d!(q - 1)!)$.

To square the system of equations, in addition to the N equations resulting from collocating (1) at the N points, an extra m equations are required. This is ensured by the m conditions for (1), viz.,

$$\sum_{j=1}^{N} \lambda_j P_i(\mathbf{x}_j) = 0, \quad i = 1, \ldots, m \tag{4}$$

To determine the coefficients $(\lambda_1, \ldots, \lambda_N)$ and $(\zeta_1, \ldots, \zeta_m)$, the collocation method is used. The collocation method here simply means applying Equation (1) and the condition (4) at every centre $i = 1, \ldots, N$ where the solution $\{u_i,\ i = 1, N\}$ is known. This gives rise to a square system of equations with $(N + m)$ equations and $(N + m)$ unknowns $\{\lambda_i,\ i = 1, N\}$ and $\{\zeta_i,\ i = 1, m\}$,

i.e. $\{\mathbf{u}\} = \mathbf{A}\{\lambda\}$, where $\{\lambda\} = [\lambda_1, \ldots, \lambda_N, \zeta_1, \ldots, \zeta_m]^T$ ($[\ldots]^T$ denotes matrix transpose). It is worth mentioning that due to the fact that the interpolated solution using the coefficient vector $\{\lambda\}$ corresponds to the exact solution at the nodes, hence, the interpolation here is an exact one rather than an approximation, where the the interpolated values at the centres may not coincide with the given solution.

In a similar representation as (1), for any linear partial differential operator $\mathscr{L}$, $\mathscr{L}u$ can be approximated by

$$\mathscr{L}u(\mathbf{x}) \simeq \sum_{j=1}^{N} \lambda_j \mathscr{L}\varphi(\mathbf{x}, \mathbf{x}_j) + \mathscr{L}\psi(\mathbf{x}) \tag{5}$$

3. ADVECTION–DIFFUSION EQUATION

Since the numerical scheme will be presented for a general three-dimensional case, let us consider the following three-dimensional linear partial differential equation, given by

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = \kappa \nabla^2 u(\mathbf{x}, t) + \mathbf{v} \cdot \nabla u(\mathbf{x}, t), \quad \mathbf{x} \in \Omega \subset \mathbf{R}^3, \ t > 0 \tag{6}$$

together with the general boundary and initial conditions

$$c_1 u(\mathbf{x}, t) + c_2 \nabla u(\mathbf{x}, t) = f(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega, \ t > 0 \tag{7}$$

$$u(\mathbf{x}, t) = u_0(\mathbf{x}), \quad t = 0 \tag{8}$$

where $u(\mathbf{x}, t)$ is, let us say, the temperature at the position $\mathbf{x}$ at time t, ∇ the gradient differential operator, Ω is a bounded domain in $\mathbf{R}^3$, $\partial\Omega$ the boundary of Ω, κ the diffusion coefficient, $\mathbf{v} = [v_x \ v_y \ v_z]^T$ the advection coefficient (or velocity) vector, c_1 and c_2 are real constants, and $f(\mathbf{x}, t)$ and $u_0(\mathbf{x})$ are known functions.

3.1. Implicit scheme

First, let us discretize (6) according to the θ-weighted scheme giving

$$u(\mathbf{x}, t + \delta t) - u(\mathbf{x}, t) = \delta t \theta \{\kappa \nabla^2 u|_{t+\delta t} + \mathbf{v} \cdot \nabla u|_{t+\delta t}\} + \delta t (1 - \theta)\{\kappa \nabla^2 u|_t + \mathbf{v} \cdot \nabla u|_t\} \tag{9}$$

where $0 \leqslant \theta \leqslant 1$, and δt is the time step size. Rearranging (9), using the notation $u^n = u(\mathbf{x}, t^n)$ where $t^n = t^{n-1} + \delta t$, we obtain

$$u^{n+1} + \alpha \nabla^2 u^{n+1} + \beta \cdot \nabla u^{n+1} = u^n + \eta \nabla^2 u^n + \xi \cdot \nabla u^n \tag{10}$$

where $\alpha = -\kappa\theta\delta t$, $\beta = [\beta_x \ \beta_y \ \beta_z]^T = -\theta\delta t\mathbf{v}$, $\eta = \kappa\delta t(1 - \theta)$ and $\xi = [\xi_x \ \xi_y \ \xi_z]^T = \delta t(1 - \theta)\mathbf{v}$. Assuming that there are a total of $(N - 4)$ collocation points or centres, $u(\mathbf{x}, t^n) = u(x, y, z, t^n)$ can be approximated by

$$u^n(\mathbf{x}) \simeq \sum_{j=1}^{N-4} \lambda_j^n \varphi(r_j) + \lambda_{N-3}^n x + \lambda_{N-2}^n y + \lambda_{N-1}^n z + \lambda_N^n \tag{11}$$

where (x, y, z) denotes the co-ordinates in the Cartesian space. To determine the interpolation coefficients $(\lambda_1, \lambda_2, \ldots, \lambda_{N-1}, \lambda_N)$, the collocation method is used by applying (11) at every point $i = 1, \ldots, N-4$, giving

$$u^n(x_i, y_i, z_i) \simeq \sum_{j=1}^{N-4} \lambda_j^n \varphi(r_{ij}) + \lambda_{N-3}^n x_i + \lambda_{N-2}^n y_i + \lambda_{N-1}^n z_i + \lambda_N^n, \quad i = 1, \ldots, N \tag{12}$$

where $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$. The additional conditions due to (4) are written as

$$\sum_{j=1}^{N-4} \lambda_j^n = \sum_{j=1}^{N-4} \lambda_j^n x_j = \sum_{j=1}^{N-4} \lambda_j^n y_j = \sum_{j=1}^{N-4} \lambda_j^n z_j = 0 \tag{13}$$

Writing (12) together with (13) in a matrix form we have

$$\{\mathbf{u}\}^n = \mathbf{A}\{\lambda\}^n \tag{14}$$

where $\{\mathbf{u}\}^n = [u_1^n \ldots u_{N-4}^n \ 0 \ldots 0]^{\mathrm{T}}$, $\{\lambda\}^n = [\lambda_1^n \ldots \lambda_N^n]^{\mathrm{T}}$ and $\mathbf{A} = [a_{ij}, 1 \leqslant i, j \leqslant N]$ is given by

$$\mathbf{A} = \begin{bmatrix} \varphi_{11} & \cdots & \varphi_{1(N-4)} & x_1 & y_1 & z_1 & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \varphi_{(N-4)1} & \cdots & \varphi_{(N-4)(N-4)} & x_{N-4} & y_{N-4} & z_{N-4} & 1 \\ x_1 & \cdots & x_{N-4} & 0 & \cdots & \cdots & 0 \\ y_1 & \cdots & y_{N-4} & \vdots & \ddots & & \vdots \\ z_1 & \cdots & z_{N-4} & \vdots & & \ddots & \vdots \\ 1 & \cdots & 1 & 0 & \cdots & \cdots & 0 \end{bmatrix} \tag{15}$$

Note that Micchelli [30] proved the non-singularity of the matrix $\mathbf{A}$ in (15) resulting from a positive-definite radial basis function φ. In other words, the solution of the system of equations (14) always exists.

Assuming that there are $p < (N-4)$ internal (domain) points and $(N-4-p)$ boundary points, then the $(N \times N)$ matrix $\mathbf{A}$ can be split into $\mathbf{A} = [\mathbf{A}_d + \mathbf{A}_b + \mathbf{A}_e]$, where

$$\begin{aligned} \mathbf{A}_d &= [a_{ij} \text{ for } (1 \leqslant i \leqslant p, 1 \leqslant j \leqslant N) \text{ and } 0 \text{ elsewhere}] \\ \mathbf{A}_b &= [a_{ij} \text{ for } (p < i \leqslant N-4, 1 \leqslant j \leqslant N) \text{ and } 0 \text{ elsewhere}] \\ \mathbf{A}_e &= [a_{ij} \text{ for } (N-3 \leqslant i \leqslant N, 1 \leqslant j \leqslant N) \text{ and } 0 \text{ elsewhere}] \end{aligned} \tag{16}$$

Using the notation $\mathscr{L}\mathbf{A}$ to designate the matrix of the same dimension as $\mathbf{A}$ and containing the elements $\tilde{a}_{ij}$, where $\tilde{a}_{ij} = \mathscr{L}a_{ij}$ (i.e. $\mathscr{L}\mathbf{A} = [\mathscr{L}a_{ij}, 1 \leqslant i, j \leqslant N]$), then Equation (10) together with (7) can be written, in a matrix form, as

$$[\mathbf{C}_{\alpha\beta} + \mathbf{B} + \mathbf{A}_e]\{\lambda\}^{n+1} = [\mathbf{C}_{\eta\xi} + \mathbf{B} + \mathbf{A}_e]\{\lambda\}^n + \{\bar{\mathbf{F}}\}^{n+1} \tag{17}$$

where

$$\mathbf{B} = c_1\mathbf{A}_b + c_2\nabla\mathbf{A}_b, \quad \mathbf{C}_{\sigma\rho} = \mathbf{A}_d + \sigma\nabla^2\mathbf{A}_d + \rho\cdot\nabla\tilde{\mathbf{A}}_d, \quad \sigma = \alpha, \eta, \ \rho = \beta, \xi \tag{18}$$

and $\{\bar{\mathbf{F}}\}^{n+1} = \{\mathbf{F}\}^{n+1} - \{\mathbf{F}\}^n$, where $\{\mathbf{F}\}^n = [0 \ldots 0 \ f_{p+1}^n \ f_{p+2}^n \ldots f_{N-4}^n \ 0 \ldots 0]^{\mathrm{T}}$. Equation (17) is obtained by combining (10), which applies to the domain points p, while (7) applies to the

boundary points ($N-4-p$). To better illustrate the physical meaning of $\tilde{\mathbf{A}}_d$, let us rewrite the advective term in (18) as

$$\begin{aligned}\rho\cdot\nabla\tilde{\mathbf{A}}_d &= \rho_x\frac{\partial\tilde{\mathbf{A}}_d}{\partial x}+\rho_y\frac{\partial\tilde{\mathbf{A}}_d}{\partial y}+\rho_z\frac{\partial\tilde{\mathbf{A}}_d}{\partial z}\\ &= \rho_x\frac{\partial\tilde{\mathbf{A}}_d}{\partial r}\frac{\partial r_{ij}}{\partial x}+\rho_y\frac{\partial\tilde{\mathbf{A}}_d}{\partial r}\frac{\partial r_{ij}}{\partial y}+\rho_z\frac{\partial\tilde{\mathbf{A}}_d}{\partial r}\frac{\partial r_{ij}}{\partial z}\\ &= \frac{\partial\tilde{\mathbf{A}}_d}{\partial r}\left(\rho_x\frac{\partial r_{ij}}{\partial x}+\rho_y\frac{\partial r}{\partial y}+\rho_z\frac{\partial r_{ij}}{\partial z}\right)\\ &= \frac{\partial\mathbf{A}}{\partial r}\left\{\max\left(\rho_x\frac{\partial r_{ij}}{\partial x},0\right)+\max\left(\rho_y\frac{\partial r_{ij}}{\partial y},0\right)+\max\left(\rho_z\frac{\partial r_{ij}}{\partial z},0\right)\right\}\end{aligned}\tag{19}$$

In fact, the term $\{\max(\rho_x\partial r_{ij}/\partial x,0)+\max(\rho_y\partial r_{ij}/\partial y,0)+\max(\rho_z\partial r_{ij}/\partial z,0)\}$ mimics an *upwinding* finite difference type scheme, where for the advective term only the contribution of *upstream* points are taken into consideration. However, it must be emphasized that due to the local nature of finite differences, the advective contributions of only the *neighbouring* points in the upstream are considered, whereas, for the present scheme the advective contributions of *all the points in the upstream region* are taken into account.

Rewriting (17) in the form

$$\{\lambda\}^{n+1}=\mathbf{H}_{\alpha\beta}^{-1}\mathbf{H}_{\eta\xi}\{\lambda\}^n+\mathbf{H}_{\alpha\beta}^{-1}\{\bar{\mathbf{F}}\}^{n+1}\tag{20}$$

where $\mathbf{H}_{\sigma\rho}=[\mathbf{C}_{\sigma\rho}+\mathbf{B}+\mathbf{A}_e]$ and making use of (14), the solution vector $\{\mathbf{u}\}^{n+1}$ is computed from $\{\mathbf{u}\}^n$ as

$$\{\mathbf{u}\}^{n+1}=\mathbf{A}\mathbf{H}_{\alpha\beta}^{-1}\mathbf{H}_{\eta\xi}\mathbf{A}^{-1}\{\mathbf{u}\}^n+\mathbf{A}\mathbf{H}_{\alpha\beta}^{-1}\{\bar{\mathbf{F}}\}^{n+1}\tag{21}$$

Putting (21) in a simpler form, viz.,

$$\{\mathbf{u}\}^{n+1}=\mathbf{D}\{\mathbf{u}\}^n+\{\mathbf{E}\}^{n+1}\tag{22}$$

where

$$\mathbf{D}=\mathbf{A}\mathbf{H}_{\alpha\beta}^{-1}\mathbf{H}_{\eta\xi}\mathbf{A}^{-1}\quad\text{and}\quad\{\mathbf{E}\}^{n+1}=\mathbf{A}\mathbf{H}_{\alpha\beta}^{-1}\{\bar{\mathbf{F}}\}^{n+1}\tag{23}$$

Although $\mathbf{H}$ and $\mathbf{D}$, in (23), look complicated, their computations are simple and straightforward operations. Furthermore, if the same collocation points and a constant time-stepping scheme are used throughout the computational process, $\mathbf{H}$ and $\mathbf{D}$ are computed only once, hence computing $\{\mathbf{u}\}^n$ from $\{\mathbf{u}\}^{n-1}$ is a simple operation of order $O(N)$. Although Equation (22) is valid for any value of $\theta\in[0,1]$, we will use $\theta=1/2$ (i.e. the Crank–Nicholson scheme), hence $\mathbf{C}_{\alpha\beta}+\mathbf{C}_{\eta\xi}=2\mathbf{A}_d$, where $\eta=-\alpha=\kappa(\delta t/2)$ and $\xi=-\beta=(\delta t/2)\mathbf{v}$.

It is worth mentioning that there were some question marks about the well-posedness of the resulting system arising from the application of radial basis functions to partial differential equations and the proof about the solvability of such a system, see the paper of Fasshauer [31] for details. However, recently Franke and Schaback [21] gave the first theoretical foundations, concerning the

convergence proof and error bound for the solution of partial differential equations with collocation and radial basis functions. They showed that the radial basis function has to be much smoother than the smoothness required for a weak solution of the differential operator. As far as the Laplace and gradient operators and the thin plate splines are concerned, the requirements are met to guarantee the positive definiteness of the resulting matrix and therefore insuring the solvability of the system, see Reference [21] for details.

From engineering and application point of views, the implementation of such schemes are very simple and straightforward, irrespective of the dimension of the problem or the shape of the domain under consideration. These schemes are remarkably simple and, as our numerical results will show, can achieve similar results as other more complicated grid/mesh methods such as finite differences and finite elements methods.

To gain some insight into the stability of the implicit method (22), the matrix method is used to analyse the method. Using Equation (22), it follows that a perturbation $\{\mathbf{z}\}^n = \{\mathbf{u}\}^n - \{\hat{\mathbf{u}}\}^n$, where $\{\hat{\mathbf{u}}\}^n$ is the computed solution, satisfies the equation

$$\{\mathbf{z}\}^{n+1} = \mathbf{D}\{\mathbf{z}\}^n \tag{24}$$

The global error in (24) will not grow as $n \to \infty$ if the eigenvalues of the amplification matrix $\mathbf{D}$ are less than unity in modulus. Substituting the matrix $\mathbf{D}$ in (23) into Equation (24) leads to

$$\mathbf{H}_{\alpha\beta}\mathbf{A}^{-1}\{\mathbf{z}\}^{n+1} = \mathbf{H}_{\eta\xi}\mathbf{A}^{-1}\{\mathbf{z}\}^n \tag{25}$$

Substituting $\mathbf{H}_{\alpha\beta}$ and $\mathbf{H}_{\eta\xi}$ by their values, and using the relations $\alpha = \{\theta/(\theta-1)\}\eta$ and $\beta = \{\theta/(\theta-1)\}\xi$, Equation (25) leads to

$$[\mathbf{I} - \theta\delta t\mathbf{M}]\{\mathbf{z}\}^{n+1} = [\mathbf{I} + (1-\theta)\delta t\mathbf{M}]\{\mathbf{z}\}^n \tag{26}$$

where $\mathbf{M} = [\kappa\nabla^2\mathbf{A}_d + \mathbf{v}\cdot\nabla\tilde{\mathbf{A}}_d]\mathbf{A}^{-1}$ and $\mathbf{I}$ is the identity matrix.

The implicit method is stable if the eigenvalues of the matrix

$$[\mathbf{I} - \theta\delta t\mathbf{M}]^{-1}[\mathbf{I} + (1-\theta)\delta t\mathbf{M}]$$

are less than one in modulus, that is,

$$\left|\frac{1 + (1-\theta)\delta t\lambda_M}{1 - \theta\delta t\lambda_M}\right| \leqslant 1 \tag{27}$$

where λ_M are the eigenvalues of the matrix $\mathbf{M}$.

It can be seen from Equation (27) that when $\theta \geqslant 1/2$ and the eigenvalues λ_M are negative (or complex with negative real parts), the implicit method (22) is unconditionally stable.

As the eigenvalues of the matrix $\mathbf{M}$ are not easy to find in closed form, a numerical approach is used here to find the eigenvalues of the matrix $\mathbf{M}$ by solving the generalized eigenproblem $[\kappa\nabla^2\mathbf{A}_d + \mathbf{v}\cdot\nabla\tilde{\mathbf{A}}_d]\{\mathbf{s}\} = \lambda_M\mathbf{A}\{\mathbf{s}\}$, where $\{\mathbf{s}\}$ is the eigenvector, using the NAG routine F02BJF. For the test examples considered in Section 4, it is found that the eigenvalues of the matrix $\mathbf{M}$ are negative. Thus, the implicit method is unconditionally stable for those cases.

3.2. *Explicit scheme*

To derive the explicit scheme, we follow the procedure in Reference [32] by assuming that

$$u(\mathbf{x}, t) = \phi(\mathbf{x})\omega(t) \tag{28}$$

then (6) becomes

$$\frac{1}{\omega(t)}\frac{\mathrm{d}\omega}{\mathrm{d}t} = \kappa\frac{1}{\phi(\mathbf{x})}\nabla^2\phi + \frac{1}{\phi(\mathbf{x})}\mathbf{v}\cdot\nabla\phi \tag{29}$$

Integrating (29) with respect to t from t_0 to t, and making use of (28), the following can be obtained:

$$u(\mathbf{x},t) = u(\mathbf{x},t_0)\exp\left\{\frac{t}{u(\mathbf{x},t_0)}(\kappa\nabla^2 u(\mathbf{x},t_0) + \mathbf{v}\cdot\nabla u(\mathbf{x},t_0))\right\} \tag{30}$$

Let (12) be rewritten as

$$u(\mathbf{x}_i,t^n) = \sum_{j=1}^{N}\lambda_j^n\varpi(\mathbf{x}_i,\mathbf{x}_j) \tag{31}$$

where $[\lambda_1^n \ldots \lambda_N^n]^{\mathrm{T}} = \mathbf{A}^{-1}[u_1^n \ldots u_{N-4}^n 0 \ldots 0]^{\mathrm{T}}$ and

$$\varpi(\mathbf{x}_i,\mathbf{x}_j) = \begin{cases} \varphi(\mathbf{x}_i,\mathbf{x}_j) & \text{for } j = 1, N-4 \\ \psi(\mathbf{x}_i) & \text{for } j = N-3, N \end{cases} \tag{32}$$

and assuming that $t = t^{n+1} = t^n + \delta t$ and $t_0 = t^n$, then, the expression of $u(\mathbf{x}_i,t^{n+1}) = u_i^{n+1}$, inside the domain, $\mathbf{x}_i \in \Omega$, $i = 1,\ldots,p$, as a function of the distribution at $t = t^n$, is obtained from (30) as

$$\begin{aligned} u_i^{n+1} &= u_i^n \exp\left\{\frac{\delta t}{u_i^n}\sum_{j=1}^{N}\lambda_j^n(\kappa\nabla^2\varpi(\mathbf{x}_i,\mathbf{x}_j) + \mathbf{v}\cdot\nabla\tilde{\varpi}(\mathbf{x}_i,\mathbf{x}_j))\right\} \\ &= u_i^n\prod_{j=1}^{N}\exp\left(\lambda_j^n\frac{\delta t}{u_i^n}\gamma_{ij}\right) \quad \text{for } \mathbf{x}_i \in \Omega \end{aligned} \tag{33}$$

where

$$\begin{aligned} \gamma_{ij} &= \kappa\nabla^2\varpi(\mathbf{x}_i,\mathbf{x}_j) + \mathbf{v}\cdot\nabla\tilde{\varpi}(\mathbf{x}_i,\mathbf{x}_j) \\ &= \kappa\nabla^2\varpi(\mathbf{x}_i,\mathbf{x}_j) + v_x\frac{\partial\tilde{\varpi}_{ij}}{\partial r}\frac{\partial r_{ij}}{\partial x} + v_y\frac{\partial\tilde{\varpi}_{ij}}{\partial r}\frac{\partial r_{ij}}{\partial y} + v_z\frac{\partial\tilde{\varpi}_{ij}}{\partial r}\frac{\partial r_{ij}}{\partial z} \\ &= \kappa\nabla^2\varpi(\mathbf{x}_i,\mathbf{x}_j) + \frac{\partial\tilde{\varpi}_{ij}}{\partial r}\left\{v_x\frac{\partial r_{ij}}{\partial x} + v_y\frac{\partial r_{ij}}{\partial y} + v_z\frac{\partial r_{ij}}{\partial z}\right\} \\ &= \kappa\nabla^2\varpi(\mathbf{x}_i,\mathbf{x}_j) + \frac{\partial\varpi_{ij}}{\partial r}\left\{\max\left(v_x\frac{\partial r_{ij}}{\partial x},0\right) + \max\left(v_y\frac{\partial r_{ij}}{\partial y},0\right) + \max\left(v_z\frac{\partial r_{ij}}{\partial z},0\right)\right\} \end{aligned} \tag{34}$$

The term $\{\max(v_x\partial r_{ij}/\partial x,0) + \max(v_y\partial r_{ij}/\partial y,0) + \max(v_z\partial r_{ij}/\partial z,0)\}$ plays the same *upwinding* role as in (19). For the boundary points, $\mathbf{x}_i \in \partial\Omega$, $i = p+1,\ldots,N-4$, u_i^{n+1} can be derived

from (7) as

$$u_i^{n+1} = \frac{1}{c_1}\left(f(\mathbf{x}_i, t^{n+1}) - c_2 \sum_{j=1}^{N} \lambda_j^n \nabla \varpi(\mathbf{x}_i, \mathbf{x}_j) \right) \quad \text{for } \mathbf{x}_i \in \partial\Omega \tag{35}$$

For the explicit scheme, the following two explicit equations can be used

$$u_i^{n+1} = u_i^n \exp\left(\frac{\delta t}{u_i^n} \sum_{j=1}^{N} \lambda_j^n \gamma_{ij} \right) \tag{36}$$

or

$$u_i^{n+1} = u_i^n + \delta t \sum_{j=1}^{N} \lambda_j^n \gamma_{ij} \tag{37}$$

Equation (37) is obtained by taking only the first two terms of the Taylor's expansion of the exponential in (36), i.e. $\exp(x) \cong 1 + x$. It can also be verified that (37) is the same as that obtained from (22) when $\theta = 0$.

To analyse the explicit scheme (37), as before, the matrix method is used. Using Equation (37), it follows that a perturbation $\{\mathbf{z}\}^n = \{\mathbf{u}\}^n - \{\hat{\mathbf{u}}\}^n$ satisfies the equation

$$\{\mathbf{z}\}^{n+1} = [I + \delta t \boldsymbol{\Omega} \mathbf{A}^{-1}]\{\mathbf{z}\}^n \tag{38}$$

where Ω is a matrix with elements γ_{ij} given by Equation (34).

The explicit method is stable if

$$|1 + \delta t \lambda_M| \leqslant 1 \tag{39}$$

where λ_M are the eigenvalues of the matrix $\mathbf{M} = [\kappa \nabla^2 \mathbf{A}_d + \mathbf{v} \cdot \nabla \tilde{\mathbf{A}}_d]\mathbf{A}^{-1}$ (as in the implicit scheme).

From Equation (39), it can be seen that when the eigenvalues λ_M are negative, the explicit method (37) is stable provided

$$\delta t \leqslant 2/|\lambda_M| \tag{40}$$

For the case when the eigenvalues are complex with negative real parts, the explicit method (37) is stable provided

$$2\mathrm{Re}(\lambda_M) + \delta t |\lambda_M|^2 \leqslant 0 \tag{41}$$

As it can be seen, the analysis of the explicit method involves, knowledge of the eigenvalues of the matrix $\mathbf{M}$ and, in particular, it requires the eigenvalue with greater modulus not to exceed a certain value. Thus, instead of calculating all the eigenvalues using the NAG routine F02BJF, the Power Method [33] can be used to find only the largest modulus eigenvalue of $\mathbf{M}$ (i.e. finding the largest modulus eigenvalue, say λ_s, of the generalized eigenproblem $[\kappa \nabla^2 \mathbf{A}_d + \mathbf{v} \cdot \nabla \tilde{\mathbf{A}}_d]\{\mathbf{s}\} = \lambda_s \mathbf{A}\{\mathbf{s}\}$, where $\{\mathbf{s}\}$ is the corresponding eigenvector). From computational point of view, the algorithm (Power Method) for calculating the largest modulus eigenvalue of $\mathbf{M}$ is straightforward and fast. Moreover, the computation of the largest modulus eigenvalue of the matrix $\mathbf{M}$ is performed only once and can be seen as a pre-processing task for any given scattered points distribution.

As mentioned previously, for efficiency purposes it is much more practical to keep the same collocation points as well as maintaining a constant time step throughout, hence the computation of $\mathbf{A}^{-1}$ and $\gamma = [\gamma_{ij}, 1 \leqslant i, j \leqslant N]$ is performed only once. This makes the computation of

$\mathbf{u}^{n+1} = [u_1^{n+1} \ldots u_{N-4}^{n+1}\ 0 \ldots 0]^{\mathrm{T}} = \Gamma(\mathbf{u}^n, \delta t)$ a simple operation which can also be highly vectorized or parallelized due to the inherent parallelism in explicit formulae where, unlike implicit schemes, recurrences are non-existent. The explicit equations are more suitable for vector and parallel processing due to the fact that the unknown vector is dependent only from an already known data, ideal process for a high locality and scalability. Similarly as in finite differences, Equation (33) is conditionally stable. However, the stability of the scheme can easily be preserved by an automatic and progressively discarded time sub-divisions as suggested in Reference [32]. This can be easily illustrated by the following algorithm:

$$\delta t_{\mathrm{s}} = 2\alpha/|\lambda_M|;\ \ n_v = \mathrm{int}(\delta t/\delta t_{\mathrm{s}}) + 1;\ \ \delta t_c = \delta t/n_v;$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n;$$

do $i = 1, \ldots, n_v$

$$\mathbf{u}^{n+1} = \Gamma(\mathbf{u}^{n+1}, \delta t_c)$$

end do

where $0 < \alpha < 1$. It is worth mentioning that the eigenvalues λ_M are simply dependent on the relative positions of the centres or collocation points. Hence, the check for stability criteria can be performed only once before the time-marching scheme. This would allow stability to be maintained without additional computations from one time step to another.

In general, the way in which the collocation points are distributed is similar to that for mesh/element distribution. Generally, a fair distribution with increased density around hot-spots is the norm, see Belytschko *et al.* [1] and Onate *et al.* [2] for details. Hot-spots are usually sub-regions of sharp variations or of complex geometry. Well-distributed points can be achieved either by generating a regular type grid points with increased density around the hot-spots or simply using a quasi-random generator with a variable density depending on the sub-region, for which the points are generated. In order to have well-distributed quasi-random points in the domain of interest, hence avoiding the chance of unnecessary clustering, the variant of Sobol quasi-random sequences suggested by Antonov and Saleev, are used as a basis for well-distributed point generation algorithm [34]. Furthermore, for more evenly improved distribution, more than the desired points can be generated by the quasi-random generator, then trimmed out using the thinning algorithm [25], which removes the points that are too close to each other (i.e. do not satisfy certain minimum distance).

4. TEST EXAMPLES

4.1. Example 1: one-dimensional case

For validation, a simple 1-D problem which has an analytical solution is considered in this example. It consists of a Dirichlet problem defined as

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + v \frac{\partial u}{\partial x}, \quad 0 < x < 1,\ t > 0 \tag{42}$$

Table I. Comparison of EULTPS, EXPTPS and ICNTPS with the analytical solution for Example 1. $u(x,t)$ is tabulated at some random locations x at different times $t = 0.2$, 0.6, and 1. $n_c = 11$ (random points), $\kappa = 0.1$, $b = 0.1$, $c = 1.61803$, $a = 1.0$, $\delta t = 0.01$, and $v = 0.1$, $Pe = v/\kappa = 1$.

t	$x = 0.25000$	$x = 0.31250$	$x = 0.37500$	$x = 0.50000$	$x = 0.62500$	$x = 0.75000$	$x = 0.8750$	$\varepsilon(t)$
	0.68591	0.61921	0.55865	0.45441	0.36973	0.30102	0.24495	0.00231
0.2	0.68592	0.61922	0.55866	0.45442	0.36973	0.30102	24495	0.00231
	0.68585	0.61917	0.55863	0.45443	0.36976	0.30105	0.24498	0.00228
	0.68079	0.61531	0.55612	0.45429	0.37110	0.30315	0.24764	—
	0.71720	0.64761	0.58406	0.47392	0.38420	0.31195	0.25393	0.00392
	0.71721	0.64763	0.58408	0.47393	0.38421	0.31196	0.25393	0.00392
0.6	0.71718	0.64759	0.58405	0.47392	0.38422	0.31198	0.25395	0.00390
	0.70857	0.64042	0.57882	0.47283	0.38625	0.31552	0.25775	—
	0.74738	0.67499	0.60881	0.49390	0.40017	0.32472	0.26425	0.00441
1.0	0.74740	0.67501	0.60883	0.49392	0.40019	0.32474	0.26426	0.00442
	0.74738	0.67500	0.60881	0.49391	0.40019	0.32474	0.26426	0.00441
	0.73749	0.66655	0.60244	0.49213	0.40201	0.32840	0.26827	—

Note: For every t, the first, second, third and the fourth row of data correspond to the EULTPS, EXPTPS, ICNTPS and analytical solutions, respectively. The average error $\varepsilon(t) = [1/(n_c - 1)] \sum_{j=1}^{n_c-2} |u(x_j,t)_{\text{numerical}} - u(x_j,t)_{\text{analytical}}|$.

$$u(0,t) = a\exp(bt), \quad u(1,t) = a\exp(bt - c), \quad t > 0 \tag{43}$$

$$u(x,0) = a\exp(-cx) \tag{44}$$

The analytical solution is given by

$$u(x,t) = a\exp(bt - cx) \quad \text{and} \quad c = \frac{v \pm \sqrt{v^2 + 4\kappa b}}{2\kappa} > 0 \tag{45}$$

For simplicity, the numerical scheme described in Section 3.1 will be referred to as ICNTPS (Implicit Crank–Nicholson Thin Plate Spline), while that of Section 3.2 will be referred to as EXPTPS (Explicit Thin Plate Spline) when Equation (36) is used and as EULTPS (Euler Thin Plate Spline) when Equation (37) is used.

Table I compares the solutions due to ICNTPS, EXPTPS, EULTPS and the analytical solution for the example given by (42)–(44) when the flow is in the positive direction ($v > 0$), using n_c quasi-random collocation points. It has to be noted that n_c is the total number of collocation points. Using an additional polynomial with degree 1 in (1) and constructing $\mathbf{A}$ in a way to obtain an $N \times N$ system of equations, $n_c = (N - d - 1)$ for a d-dimensional problem. It can be seen from Table I that all the schemes have a good agreement with the analytical solution, with ICNTPS as the most efficient scheme since it requires less cpu-time by comparison, due to the fact that it does not need sub-time step divisions for stability constraints.

From stability point of view, it is found that the implicit method ICNTPS is unconditionally stable, whereas the explicit methods EULTPS and EXPTPS are stable for $\delta t \leqslant 2/|\lambda_s| = 0.0165$, where λ_s is the largest modulus eigenvalue of the matrix $\mathbf{M} = [\kappa\nabla^2\mathbf{A}_d + \mathbf{v} \cdot \nabla\tilde{\mathbf{A}}_d]\mathbf{A}^{-1}$; $\lambda_s = -121.544 < \lambda_M < 0$. Furthermore, several tests were carried out by choosing different values of δt which lie within the stability condition (40) are found to always give a stable numerical

Table II. Comparison of the different schemes with the analytical solution for Example 1. $n_c = 21$, $\kappa = 0.1$, $b = 0.2$, $c = 5.61553$, $a = 1.0$, $\delta t = 0.001$, and $v = -0.3$, $Pe = -3$.

t	$x = 0.06250$	$x = 0.25000$	$x = 0.46875$	$x = 0.56250$	$x = 0.75000$	$x = 0.96875$	$\varepsilon(t)$
	1.04577	0.94095	0.83264	0.79054	0.71302	0.62963	0.00082
0.4	1.04577	0.94096	0.83265	0.79055	0.71302	0.62963	0.00082
	1.04577	0.94096	0.83265	0.79055	0.71302	0.62963	0.00082
	1.04593	0.94140	0.83258	0.78988	0.71094	0.62876	—
	1.13286	1.01934	0.90222	0.85674	0.77285	0.68215	0.00105
	1.13286	1.01935	0.90224	0.85675	0.77286	0.68215	0.00105
0.8	1.13286	1.01935	0.90224	0.85675	0.77286	0.68215	0.00105
	1.13304	1.01981	0.90192	0.85567	0.77015	0.68113	—
	1.17910	1.06097	0.93913	0.89180	0.80449	0.71000	0.00112
1.0	1.17910	1.06098	0.93914	0.89182	0.80450	0.71000	0.00112
	1.17910	1.06098	0.93914	0.89182	0.80450	0.71000	0.00112
	1.17928	1.06142	0.93873	0.89059	0.80158	0.70893	—

See footnote of Table I for explanatory notes.

solution, whereas by choosing δt outside the stability condition, it is found that the solution becomes unbounded after a few time steps.

Table II shows a comparison between the analytical, ICNTPS, EXPTPS, and EULTPS solutions for the case of $v<0$. It can be seen that again the schemes are in good agreement with the exact solution, irrespective of the velocity direction. It can be noticed that the effect of an opposite advection has contributed in maintaining higher values of $u(x,t)$ for the second half $x>0.5$ by comparison with Table I. Further tests were carried out which show that both the explicit and implicit methods also work equally well using regular finite-difference type grid points. Figure 1 compares the solutions for an advective-dominated case with large Peclet number $Pe = v/\kappa = 10$. It can be seen that the good accuracy of the schemes is still preserved. For stability analysis of this case, it is found that the implicit method ICNTPS is unconditionally stable, whereas the explicit methods EULTPS and EXPTPS are stable for $\delta t \leqslant 2/|\lambda_s| = 0.0043$; $\lambda_s = -466.971$ (as expected from the stability condition (40)). It can be seen, as expected, that the high Peclet number increases the stability constraints with comparison to the case in Table I.

4.2. *Example 2: two-dimensional case*

Let us consider the following PDE which is encountered in many transport phenomena such as vorticity, advective, convective heat transfer, decay, species migration, and other diffusive–advective processes

$$\frac{\partial u}{\partial t} = \kappa_x \frac{\partial^2 u}{\partial x^2} + \kappa_y \frac{\partial^2 u}{\partial y^2} + v_x \frac{\partial u}{\partial x} + v_y \frac{\partial u}{\partial y} \quad \text{on } \Omega = \{(x,y) \mid 0 \leqslant x, y \leqslant 1\},\ t>0 \tag{46}$$

with a Dirichlet boundary type condition

$$u(x,y,t) = h(x,y) - gt \quad \text{on } \partial\Omega = \{(x,y) \mid x = 0,1, y = 0,1\},\ t>0 \tag{47}$$

and the initial condition

$$u(x,y,t) = h(x,y) \quad \text{on } \Omega,\ t = 0 \tag{48}$$

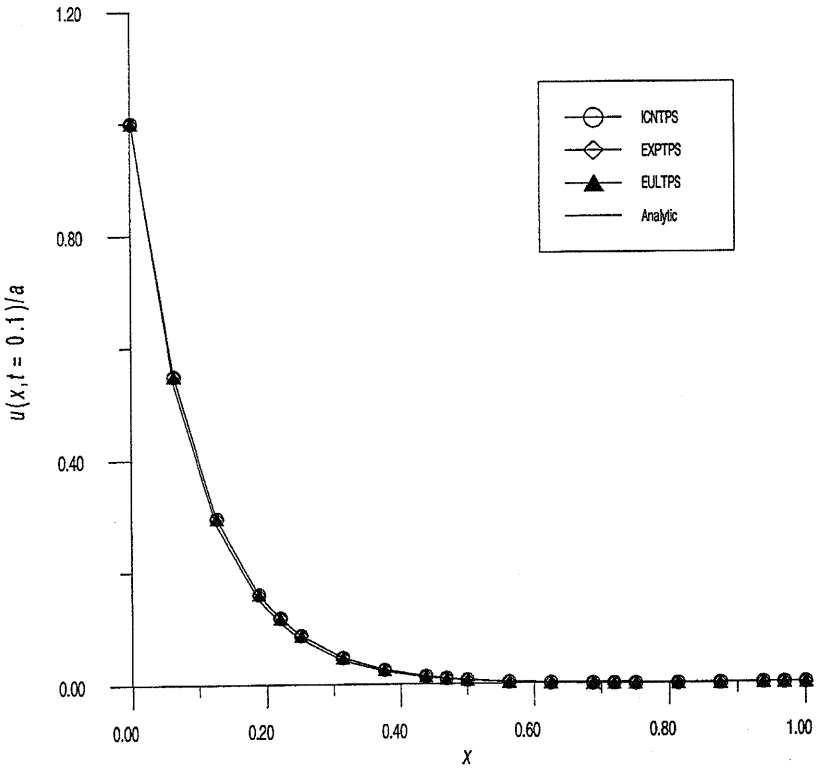

Figure 1. The solution $u(x,t)$ at $t = 0.1$ using 21 random points (Example 1), $n_c = 21$, $\kappa = 0.01$, $b = 0.1$, $c = 10.099$, $\alpha = 1000.0$, $\delta t = 0.01$ and $v = 0.1$, $Pe = v/\kappa = 10$.

where g is a positive constant and $h(x, y)$ is given by

$$h(x,y) = 5\exp\left\{-\frac{(9x-2)^2}{4} - \frac{(9y-2)^2}{4}\right\} + 7\exp\left\{-\frac{(9x+1)^2}{50} - \frac{(9y+1)}{10}\right\}$$
$$+ 4\exp\left\{-\frac{(9x-7)^2}{4} - \frac{(9y+3)^2}{4}\right\} - 2\exp\{-(9x-4)^2 - (9y-7)^2\} \qquad (49)$$

Example 2 is solved using the present schemes with 121 quasi-random points and the finite difference method (FDM) with upwinding on an equivalent grid system of 11×11, see Figure 2. The quasi-random points are generated using the variant of the Sobol sequence suggested by Antonov and Saleev, see Reference [34] for details. Figure 3 shows a comparison of the solutions at different times using ICNTPS and FDM. It can be seen that both methods give more or less similar solutions. Similar solutions to that due to ICNTPS are obtained using both EULTPS and EXPTPS. As stability analysis is concerned, again it is found that the implicit method is unconditionally stable, whereas the explicit methods are conditionally stable. However, the automatic subdivisions,

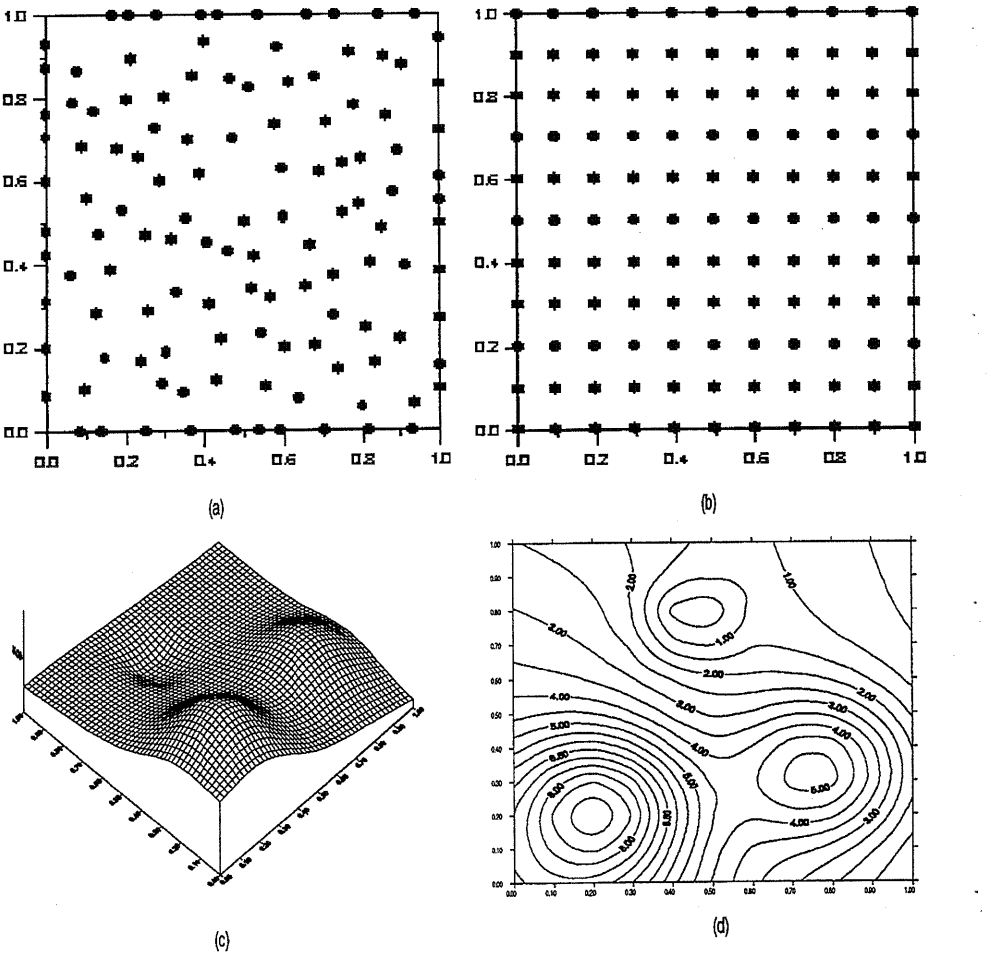

Figure 2. Discretization and the initial condition for Example 2: (a) 121 quasi-random points, (b) 121 regular type FDM grid points, (c) 3-D plot of initial distribution $u(x, y, 0)$, (d) contour of $u(x, y, 0)$.

based on the stability criteria, allow the explicit methods to be used for larger time steps without stability failures and the need for large memory requirements.

5. CONCLUSIONS

The numerical results show that radial basis functions based meshless schemes achieve comparable results as other mesh/grid-based methods. Furthermore, they are remarkably simple, especially for complicated domains and higher dimensions.

The thin plate spline has been chosen due to the fact that it is as accurate as the multiquadrics, but without the additional need to compute a shape parameter [25]. It has already been reported

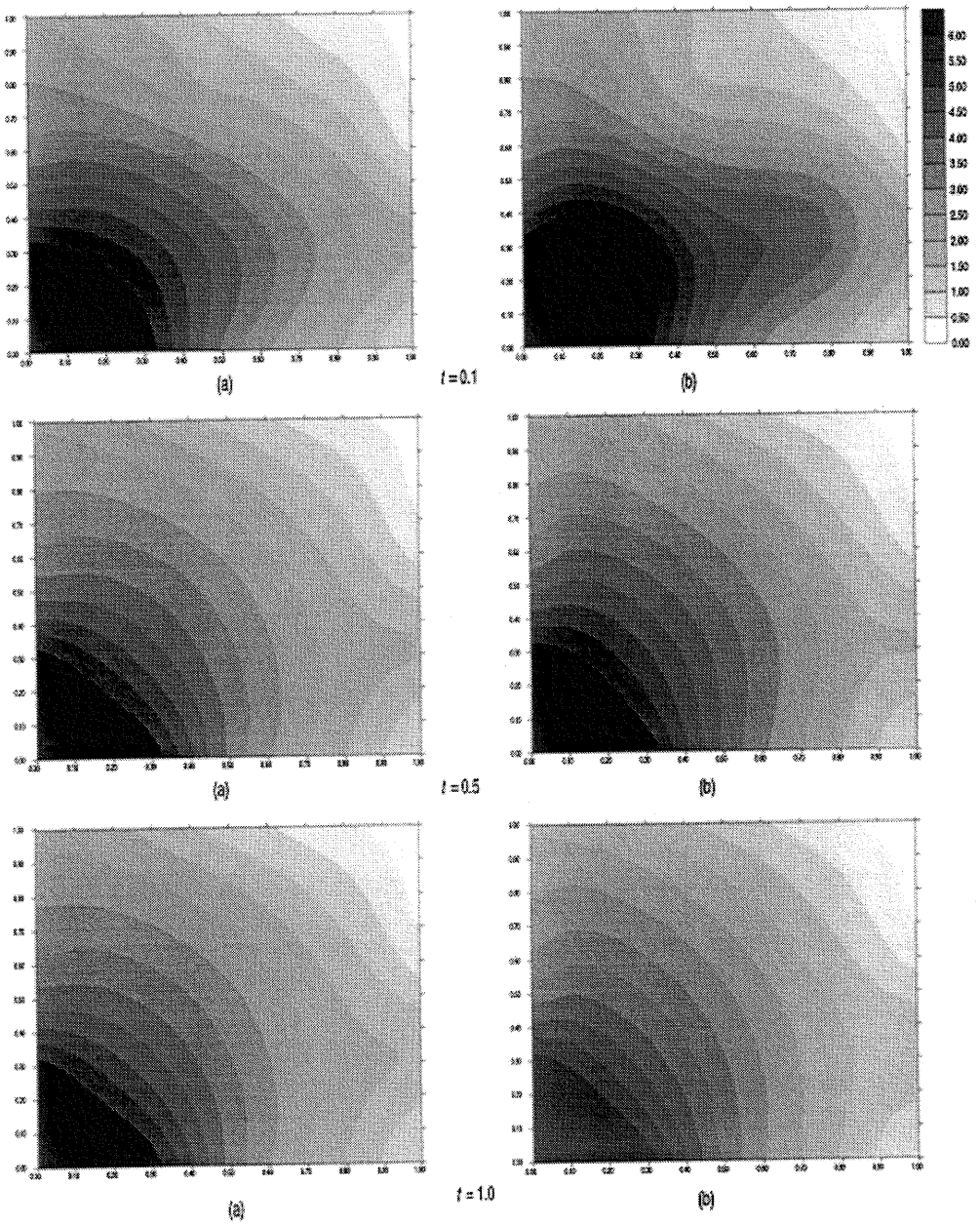

Figure 3. Distribution $u(x, y, t)$ at different times $t = 0.1, 0.5$, and 1.0 using (a) ICNTPS and (b) FDM. $\kappa_x = 0.2$, $\kappa_x = 0.3$, $v_x = 0.1$, $v_y = -0.2$, $g = 0.1$, $\delta t = 0.05$.

elsewhere that radial basis functions based schemes works well on both regular type grid or quasi-random points distribution and with more or less similar results when different radial basis functions are used [25]. This should remain valid here and certainly for most linear PDEs. For instance, Zerroukat *et al.* [25] showed that for the heat equation both MQ and TPS achieve good accuracy on problems with regular or complex shaped domains.

When using global radial functions, the resulting matrices are fully populated, which may give rise to ill-conditioning, especially for large number of points. Although, the ill-conditioning problem can be resolved using pre-conditioning techniques [35], one has to face the fact that the CPU-time increases with the number of collocation points. This makes the computation of complex problems, which necessitate large number of points, relatively expensive. One alternative is the use of compactly supported radial basis functions [36–38]. This results in a sparse matrices and avoids the above-mentioned problems related to global radial basis functions. Furthermore, the use of compactly supported radial basis functions results in an acceptable loss of accuracy but a huge saving in CPU-time by comparison to globally supported ones [38].

ACKNOWLEDGEMENTS

The authors wish to thank Prof. R. Schaback and Dr. Holger Wendland of the Institute for Numerical and Applied Mathematics, University of Goettingen, Germany, for their valuable discussions on many occasions.

REFERENCES

1. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1996; **139**:3–47.
2. Onate E, Idelsohn S, Zienkiewicz OC, Taylor RL, Sacco C. A finite point method in computational mechanics: applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering* 1996; **39**:3839–3866.
3. Jensen PS, Finite difference techniques for variable grids. *Computers and Structures* 1972; **2**:17–29.
4. Perrone N, Kao R. A general finite difference method for arbitrary meshes. *Computers and Structures* 1975; **5**:45–47.
5. Liszka T, Orkisz J. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures* 1980; **11**:83–95.
6. Liszka T. An interpolation method for an irregular set of nodes. *International Journal for Numerical Methods in Engineering* 1984; **20**:1594–1612.
7. Gingold RA, Moraghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1997; **181**:375–389.
8. Moraghan JJ. Why particles methods work. *SIAM Journal on Scientific and Statistical Computing* 1982; **3**:422–433.
9. Moraghan JJ. An introduction to SPH. *Computer Physics Communications* 1988; **48**:89–96.
10. Nayroles B, Touzot G, Villon P. Generalizing the FEM: diffuse approximation and diffuse elements. *Computational Mechanics* 1992; **10**:307–318.
11. Belytschko T, Lu Y, Gu L. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering* 1994; **37**:229–256.
12. Lu Y, Belytschko T, Gu L. A new implementation of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering* 1994; **113**:397–414.
13. Duarte CA, Oden JT. *Hp* clouds: a meshless method to solve boundary value problems. *Technical Report 95-05*, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 1995.
14. Liu WK, Jun S, Belytschko T, Reproducing Kernel Particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**:1081–1106.
15. Liu WK, Chen Y, Jun S, Chen JS, Belytschko T, Pan C, Uras RA, Chang CT. Overview and applications of the reproducing kernel particle methods. *Archives of Computational Methods in Engineering*, in press.
16. Duarte CA. A review some meshless methods to solve partial differential equations. *Technical Report 95-06*, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 1995.
17. Kansa EJ, Multiquadrics: a scattered data approximation scheme with applications to computational fluid-dynamics. *Computational Mathematics and Applications* 1990; **19**:147–161.

18. Dubal MR, Oliveira SR, Matzner RA. Solution of elliptic equations in numerical relativity using multiquadrics. In *Approaches to Numerical Relativity*, R. d'Inverno (ed.). Cambridge University Press: Cambridge, 1992; 265–280.
19. Moridis GJ, Kansa EJ. The Laplace transform multiquadric method: a highly accurate scheme for the numerical solution of linear partial differential equations. *Journal of Applied Science Computations* 1994; **1**:375–407.
20. Sharan M, Kansa EJ, Gupta S. Application of multiquadric method for numerical solution of elliptic partial differential equations. *Applied Mathematical Computations* 1997; **84**:275–303.
21. Franke C, Schaback R. Solving partial differential equations by collocation using radial basis functions. *Applied Mathematical Computations* 1998; **93**:73–82.
22. Franke C, Schaback R. Convergence orders of meshless collocation methods using radial basis functions. *Advances in Computational Mathematics* 1998; **8**:381–399.
23. Golberg MA, Chen CS. A bibliography on radial basis function approximations. *Boundary Elements Communications* 1996; **7**:155–163.
24. Fasshauer GE, A overview of radial basis functions. Department of Mathematics, Illinois Institute of Technology, Chicago, IL 60616, U.S.A. Reprint in: http://amadeus.csam.iit.edu/faass/refs.ps.gz.
25. Zerroukat M, Power H, Chen CS, A numerical method for heat transfer problems using collocation and radial basis functions. *International Journal for Numerical Methods in Engineering* 1998; **42**:1263–1278.
26. Franke R, Scattered data interpolation: test of some methods. *Mathematics of Computation* 1982; **38**:181–200.
27. Carlson RE, Foley T. The parameter R^2 in multiquadric interpolation. *Computational Mathematics and Applications* 1991; **21**:29–42.
28. Duchon J. Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In *Constructive Theory of Functions of Several Variables*, Lecture Notes in Mathematics, Vol. 38, Schempp W, Zeller K (eds). Springer: Berlin, 1977; 85–100.
29. Hardy RL. Theory and application of the mutiquadric-biharmonic method. *Computers and Mathematics with Applications* 1990; **19**:163–208.
30. Micchelli CA. Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constructive Approximation* 1986; **2**:11–22.
31. Fasshauer GE. Solving partial differential equations by collocation with radial basis functions. In *Proceedings of Chamonix 1996*, Le Mechaute M, Rabut C, Schumaker LL (eds). Vanderbilt University Press: Nashville, TN. 1997: 1–7.
32. Zerroukat M, Chatwin CR. A more accurate FD solution for the 3D non-linear heat equation. *Communications in Numerical Methods in Engineering* 1995; **11**:535–544.
33. Twizell EH, *Computational Methods for Partial Differential Equations*. Ellis Horwood Ltd. Chichester, 1984.
34. Press WH, Teukolsky SA, Vetterling WT, Flannery BP. *Numerical Recipes in Fortran* (2nd edn.). Cambridge University Press: Cambridge, 1992; 299–305.
35. Dyn N, Levin D, Rippa S. Numerical procedures for global surface fitting and scattered data by radial functions. *SIAM Journal on Scientific and Statistical Computing* 1986; **7**:639–659.
36. Wendland H. Piecewise polynomial, positive definite and compactly supported radial basis functions on minimal degree. *Advances in Computational Mathematics* 1995; **4**:389–396.
37. Wu Z. Compactly supported positive definite radial functions. *Advances in Computational Mathematics* 1995; **4**:283–292.
38. Zerroukat M. A boundary element scheme for diffusion problems using compactly supported radial basis functions. *Engineering Analysis with Boundary Elements* 1999; **23**:201–209.