*Fifth International Conference on CFD in the Process Industries*
*CSIRO, Melbourne, Australia*
*13-15 December 2006*

# NUMERICAL COMPARISON OF RADIAL BASIS FUNCTIONS AND GENERALISED SMOOTHED PARTICLE HYDRODYNAMICS

**Joseph HA**

CSIRO Mathematical and Information Sciences, Clayton, Victoria 3169, AUSTRALIA

## ABSTRACT

Mesh free methods can be grouped into two approaches. One is based on field approximations such as moving least square approximations and radial basis functions (RBF) and the other is based on kernel approximations such as smoothed particle hydrodynamics (SPH). This paper presents a unified approach to implement the RBF and SPH methods for solving partial differential equations in general and for solving problems in computational fluid dynamics in particular.

There are many forms of RBF and SPH. This paper restricts attention to multiquadric RBF's and a particular SPH that satisfies certain completeness and reproducing conditions. Completeness and reproducing conditions enables SPH to incorporate boundary conditions in similar fashions to mesh based methods such as finite element. A number of numerical examples are presented to demonstrate the effectiveness of the two mesh free methods. Some remarks with respect to their computational efficiencies and implementation are also discussed.

## NOMENCLATURE

| | |
|---|---|
| $\mathbf{g}$ | gravity |
| $p$ | pressure |
| $T$ | temperature |
| $\mathbf{v}$ | velocity |
| $\alpha$ | thermal diffusivity |
| $\beta$ | coefficient of thermal expansion |
| $\rho$ | density |
| $\nu$ | kinematic viscosity |

## INTRODUCTION

Mesh free methods have attracted much attention recently. Two distinct directions are followed by these methods. One is based on field approximations such as radial basis functions (RBF), element free Galerkin and moving least square approximations. The other is based on kernel approximations such as smoothed particle hydrodynamics (SPH). The kernel approximations used in the original SPH proposed by Lucy (1977) and Gingold and Monaghan (1977) suffer from certain inconsistencies. Various approaches to remedy these inaccuracies have been reported in the literature. It has been shown that the kernel approximations can be corrected so that they reproduce linear functions exactly (see, for example, Belytschko et al. (1966). Other workers, such as Johnson and Beissel (1966), Randles and Libersky (1996), and Krongauz and Belytschko (1998), developed corrected derivative methods. Essentially, these methods replace the standard SPH approximant with more sophisticated interpolant that was constructed by imposing certain consistency conditions. Liu et al. (1995) showed that the reproducing kernel provides boundary correction as well as removing the tensile instability. Chen and Beraun (2000), on the other hand, developed a generalised SPH method (GSPH) by applying the kernel estimate into the Taylor series expansion. Their formulation extends not only the ability of standard SPH to model partial differential equations with higher order derivatives but to enforce boundary conditions directly as well.

The dependence of SPH kernel on the smoothing length introduces additional complexities into the application of adaptive mesh refinement techniques to SPH. It is thus of interest to explore other mesh free methods that do not have such complexity. In the last decade or so, another group of mesh free methods that is based on the function approximation by RBFs either globally or compactly supported was developed to solve partial differential equations (see, for example, Kansa, 1990). RBF interpolation is required to be exact at the nodes, so one drawback of these methods is the need to solve the full coefficient matrix arising from the function approximation. A common approach to improve computational efficiency is to ensure sparsity, either by using functions of compact support, or by using domain decomposition (see, for example, Dubal, 1994). In this paper, we applied the approach of SPH to RBF in using the nearest neighbours of a particle/node for estimating its derivatives. Thus, computer programs implementing the SPH and RBF methods can share the same structure. They differ only in their different estimates of the derivatives. The aim of this paper is to present the results of such implementation of RBF, and to compare them to GSPH. Most applications of standard SPH are to simulate compressible fluids. The second aim is to study the application of GSPH and RBF to some benchmark incompressible fluid problems for testing CFD codes. Also unlike standard SPH discretisation, all the numerical examples are obtained from substituting each term of the governing equations by their corresponding RBF or GSPH derivative approximations directly. Finally, the simplicity of implementing adaptive refinement and variable resolution to RBF are briefly examined.

## GENERALISED SPH

Applying the kernel approximation to the Taylor series expansion for $f(x)$ in the neighbourhood of $x$, Chen and Beraun (2000) derived results that improve the approximation accuracy of SPH. In 1D, the GSPH approximation of a function $f(x)$ and its first two

derivatives are given in Equations (1) - (3). Higher order derivatives can easily be derived.

$$f(x) = \frac{\int f(x')W(x-x',h)dx'}{\int W(x-x')dx'} \quad (1)$$

$$\frac{df(x)}{dx} = \frac{\int (f(x)-f(x'))\frac{dW}{dx}dx'}{\int (x-x')\frac{dW}{dx}dx'} \quad (2)$$

$$\frac{d^2 f(x)}{dx^2} = \frac{\int (f(x)-f(x'))\frac{d^2W}{dx^2}dx' - \frac{df}{dx}\int (x-x')\frac{d^2W}{dx^2}dx'}{\frac{1}{2}\int (x-x')^2\frac{d^2W}{dx^2}dx'} \quad (3)$$

The same procedure can be followed to derive approximations for functions in higher dimensions. However, the derivative estimates for higher dimensions involve matrix inversion. It is clear that GSPH is computationally more expensive to use than conventional SPH. The extra terms in the above approximations can be interpreted as corrections to the boundary deficiency in the conventional SPH. The results are equivalent to some of the results of Liu et al. (1995) and Krongauz and Belytschko (1998) obtained from imposing certain completeness and consistency conditions. The above approximations are algebraically correct for a function if it is constant, for its first derivative if it is constant or linear, and for its second derivative if it is constant, linear or quadratic.

It is well appreciated that SPH is closely related to the finite element method. The main difference between the two methods is that the SPH kernel approximation of a function does not satisfy the Kronecker delta property. It is thus not possible to impose essential boundary conditions in conventional SPH. The inclusion of $f(x)$ and $df/dx$ in the above first and second derivative estimates enable the direct insertion of Dirichlet and Neumann boundary conditions, if they exist, in the GSPH method.

## RADIAL BASIS FUNCTION

The development of RBFs into a mesh free method for solving partial differential equations arises from the recognition that a radial basis function interpolant can be smooth and accurate on any set of nodes in any dimension. The starting point is that the approximation of a function $f(\mathbf{x})$ for a set of distinct points $\mathbf{x}_i, i = 1, \cdots, \mathrm{N}$ can be written as a linear combination of $N$ RBFs.

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i \phi(\| x - x_i \|) \quad (4)$$

where $\phi(\| \mathbf{x} - \mathbf{x_i} \|)$ denotes a positive definite RBF. The unknown coefficients $\alpha_i$ are to be determined from the system of equations formed by $f(\mathbf{x}_j), j = 1, \cdots, \mathrm{N}$. Once they are determined, the $m$-th spatial derivatives of $f(\mathbf{x})$ are approximated by taking the $m$-th spatial derivatives of the RBFs.

$$\frac{\partial^m f}{\partial x^m} = \sum_{i=1}^{\mathrm{N}} \alpha_i \frac{\partial^m f}{\partial x^m} \quad (5)$$

The application of Equations (4) and (5) provides the framework for the numerical solution of partial differential equations and their boundary conditions.

There are many RBFs either globally or compactly supported. An example of a globally supported RBF that is used extensively is multiquadric (MQ): $\phi(r) = \sqrt{r^2 + c^2}$, where $\boldsymbol{r} = \| \mathbf{x} - \mathbf{x_i} \|$ and $c > 0$. In this paper, a general MQ $\phi(\boldsymbol{r}) = \left(\boldsymbol{r}^2 + \boldsymbol{c}^2\right)^{\boldsymbol{q}}$ is used. It is well known that the shape parameters $q$ and $c$ strongly influence the accuracy of MQ approximation. An important unsolved problem is to find a method to determine the optimal value of $q$ and $c$. Following the work of Liu, et al. (2005), initial nodal spacing is used for the value of $c$ and 1.03 is used for the value of $q$ in the numerical examples presented in this paper. To improve boundary treatment, methods using MQ usually have a polynomial of zero degree added to the right hand side of Equation 4.

In this paper, the approach of SPH is applied to RBF in that only neighbouring particles/nodes within a given radial distance from the particle/node of interest are used in estimating its derivatives. The term smoothing length as applied in SPH is used to describe the size of this support region. In the RBF case, the smoothing length is equal to initial nodal spacing. This has the advantage of avoiding the inversion of large coefficient matrix making problems requiring large number of nodes more amenable to numerical solution. In general, the larger the supporting region the higher is the accuracy of the approximation. The need to invert coefficient matrix makes the method more expensive than the generalised SPH.

## NUMERICAL EXAMPLES

First, the GSPH and RBF approximations of $\partial u/\partial x$ of the function $u(x, y) = \sin(x)\cos(y^2)$ are studied. Figure 1 compares the errors of GSPH with those of RBF using supporting regions of $2h$ and $3h$, where $h$ denotes the smoothing length. It shows that $3h$ is a good value to use for the RBF supporting region. For these functional evaluations, the RBF method gives better accuracy than the generalised SPH at the expense of more computational effort.

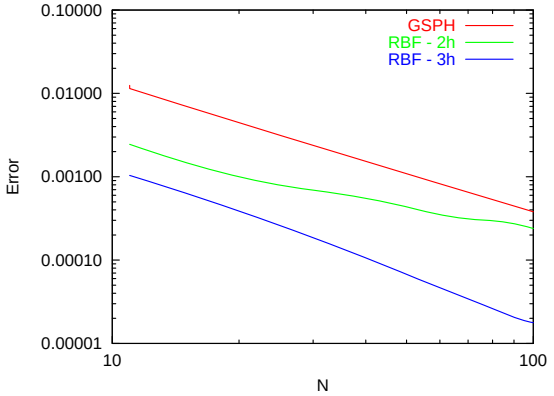


**Figure 1:** Errors for estimating $\partial u/\partial x$ using GSPH and RBF.

#### Heat Conduction

To demonstrate that the GSPH method can impose boundary conditions directly, the following heat conduction problem is solved in the domain $0 \le x \le 1$, $0 \le y \le 1$, initial condition $T(x,y,0) = -1$, Neumann boundary condition $\partial T(x,1,t)/\partial y = 0$ at $y = 1$ and Dirichlet condition at the other boundaries $T(0,y,t) = T(1,y,t) = T(x,0,t) = 1$.

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \tag{6}$$

where $T$ denotes temperature and $t$ time. Both RBF and GSPH give the result shown in Figure 2 and appear identical to the result obtained by Jeong et al. (2003) who implement the boundary conditions to the conventional SPH in a different way.

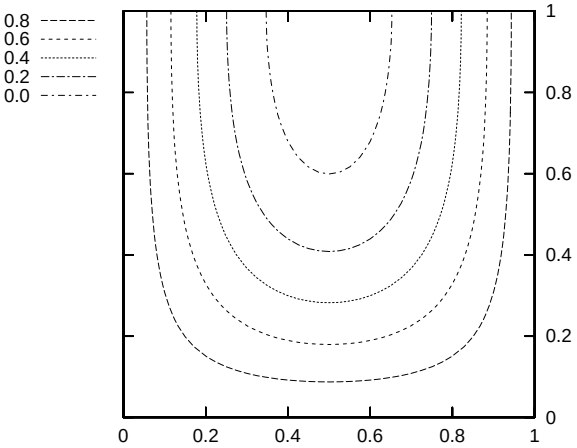


**Figure 2:** Temperature profiles at $t$=0.08.

#### Burger Equation

Here, the following 3D Burger's equation is solved using GSPH and RBF and the numerical results are compared with the analytical solution.

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} - \nu \nabla^2 \boldsymbol{v} = 0 \tag{7}$$

Figure 3 shows the computed results for $\nu$ = 0.25 and $\nu$ = 0.05 at various times. The solution becomes more shock-like as the viscosity parameter decreases. Figure 4 compares L1-norm errors for GSPH and RBF for $\nu$ = 0.05. The number of particles used is 41×41×41. Both methods give similar results.

#### Incompressible Navier-Stokes Equation

The GSPH and RBF methods are applied to three standard CFD test problems – 2D Poiseuille flow, 2D lid-driven cavity and natural convection in a square cavity. For the 2D Poiseuille flow and lid-driven cavity examples, the following Navier Stokes equation in 2D is solved

$$\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{t}} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\frac{1}{\rho} \nabla \boldsymbol{p} + \nu \nabla^2 \boldsymbol{v} + \boldsymbol{F} \tag{8}$$

For the Poiseuille flow, the boundary conditions are $\boldsymbol{v}$ = (0,0) on y = 0 and y = L, where L = 0.001 and $\boldsymbol{F}$ = 1.25x10$^{-6}$/Re. For the lid-driven cavity problem, the boundary conditions are $\boldsymbol{v}$ = (-1,0) on $y$=1 and $\boldsymbol{v}$ = (0,0) on the other three sides of the unit square.

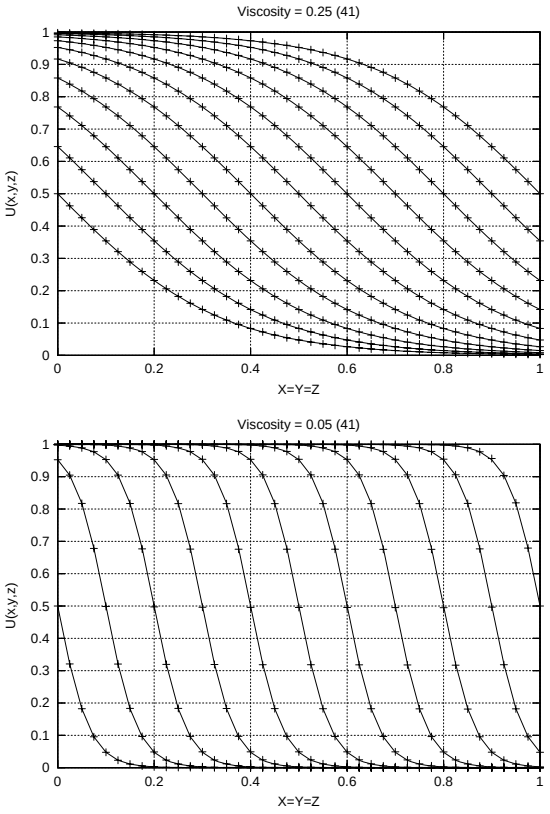


**Figure 3:** GSPH solution (+) of 3D Burger's equation compared to analytical solution (solid line) along the line x = y = z at various times for viscosities of 0.25 and 0.05.

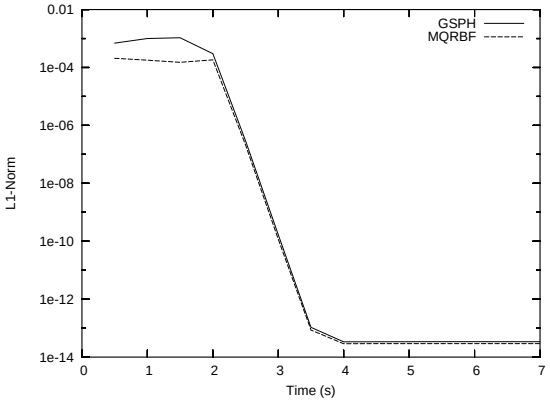


**Figure 4:** L1-norm errors for solving 3D Burger's equation.

For an incompressible fluid, the Navier Stokes equation is complemented by the incompressibility constraint, $\nabla \cdot \boldsymbol{v}$ = 0. In general, velocity $\boldsymbol{v}^{n+1}$ at time $t^{n+1}$ obtained by solving Equation (8) does not satisfy the incompressibility constraint. This constraint on velocity must be satisfied at all times. In this paper, the following steps are iterated until $\nabla \cdot \boldsymbol{v} \approx 0$ is reached.

1. $\Delta p_k = -\gamma \nabla \cdot \boldsymbol{v}_k^{n+1}$
2. $\Delta \boldsymbol{v}_k^{n+1} = \Delta t \nabla (\Delta p_k^n)$

Here, $k$ is the iteration counter and $\Delta f_k = f_{k+1} - f_k$. Upon convergence, the above procedure gives the new pressure $p^{n+1}$ and divergent free velocity $\boldsymbol{v}^{n+1}$ for time $t^{n+1}$. The parameter $\gamma$ controls the rate of convergence and must satisfy the stability requirements $0 \le \gamma \le (\Delta x)^2 / 4\Delta t$. The iteration is equivalent to solving a Poisson equation for the pressure.

### 2D Poiseuille Flow

Figure 5 shows the GSPH solutions of 2D Poiseuille flow on a $51 \times 51$ grid for Re = 0.0125, 5, 10 and 100 compare well with the analytical solutions. Similar RBF results are obtained. Sigalotti, et al. (2003) reported that their SPH solution for Re = 5 eventually becomes unstable after about 280 s. For the four cases considered here, the GSPH and RBF solutions do not exhibit any instability up to twice the time when the steady state solution is reached.

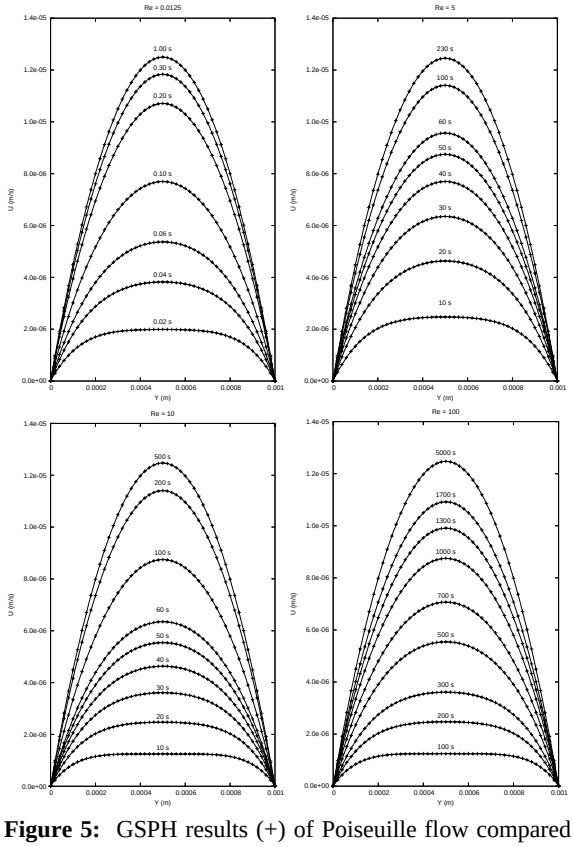


**Figure 5:** GSPH results (+) of Poiseuille flow compared to series solutions (solid lines) for Re = 0.0125, 5, 10 and 100.

### 2D Lid-Driven Cavity

Figure 6 shows that the GSPH solutions for Re = 1000 on a $129 \times 129$ grid using 3 different kernels compare well with the benchmark solutions 1, 2 and 3 of Ghia et al. (1982), Botella and Peyret (1998) and Erturk et al. (2005) respectively. In the figure, W3 denotes the cubic spline kernel of Monaghan (1992), W4 the quartic spline kernel of Liu et al (2003) and W5 the quintic spline kernel of Morris et al. (1997). For this problem, W4 gives the best result with W3 and W5 giving similar results. Figure 7 shows the RBF result is similar to the GSPH result using kernel W4 for Re = 1000.

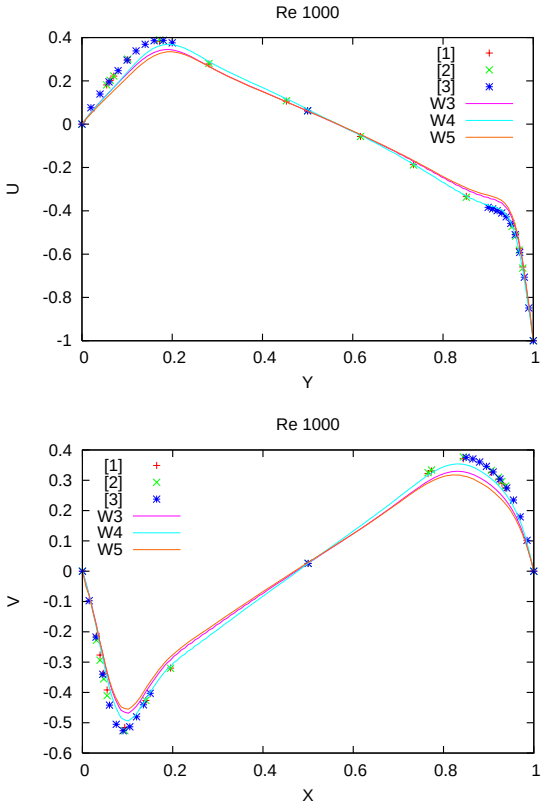


**Figure 6:** Comparison between published results and GSPH solutions for Re = 1000 of the lid-driven cavity problem using different kernels.

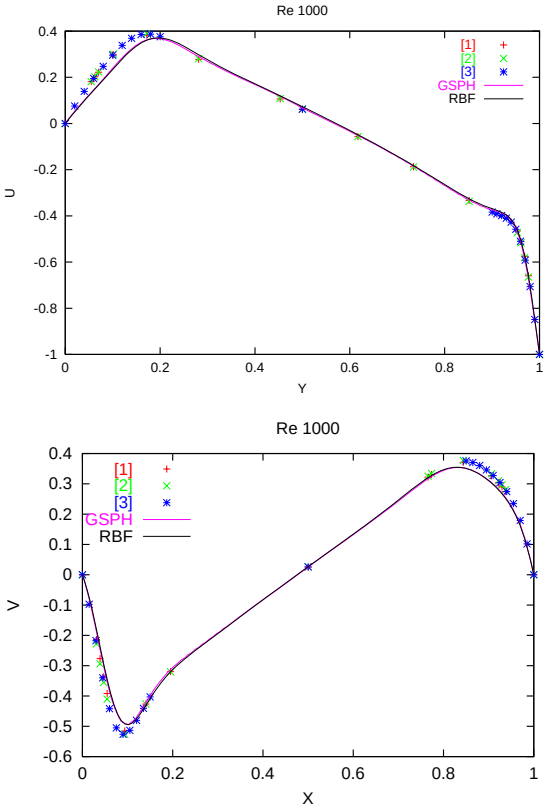


**Figure 7:** RBF results of lid-driven cavity problem compared to GSPH and published results for Re = 1000.

#### Natural Convection in a Differentially Heated Cavity

For incompressible fluid flow in a differentially heated square cavity of side $L$, the following equations are solved

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v}\cdot\nabla\boldsymbol{v} + \frac{1}{\rho}\nabla p - \nu\nabla^2\boldsymbol{v} = \beta(T - T_r)\boldsymbol{g} \qquad (9)$$

$$\frac{\partial T}{\partial t} + \boldsymbol{v}\cdot\nabla T = \alpha\nabla^2 T \qquad (10)$$

The initial conditions are $\boldsymbol{v}(x, y, 0) = (0, 0)$ and $T(x, y, 0) = T_r$ . The boundary conditions are $\boldsymbol{v} = (0, 0)$ on cavity boundary, $T(0, y, t) = T_h$ , $T(L, y, t) = T_c$ , $\partial T(x,0,t)/\partial y = \partial T(x,L,t)/\partial y = 0$. Here, $T_r$ , $T_h$ and $T_c$ denote the reference, hot and cold wall temperatures respectively. Table 1 shows that the GSPH and RBF (enclosed by <>) results compare well with the benchmark solutions for Prandtl number 0.71 and Rayleigh numbers $10^4$ - $10^6$. In the table, the numbers enclosed by [] and () are the results of Leal et al. (1999) and de Vahl Davis (1983) respectively. Using a remeshed SPH approach, Chaniotis, et al. (2002) obtained the values of $u_{max}$ = 17.31 at y = 0.823 and $v_{max}$ = 20.05 at x = 0.112 for Ra = $10^4$ which are not as close to the results of Leal et al. and de Vahl Davis as the GSPH and RBF results.

| | $u_{max}$ | $y$ | $v_{max}$ | x |
|---|---|---|---|---|
| Ra = $10^4$ | [16.18]<br>16.18<br><16.08><br>(16.178) | [0.823]<br>0.822<br><0.827><br>(0.823) | [19.63]<br>19.63<br><20.05><br>(19.617) | [0.119]<br>0.119<br><0.113><br>(0.119) |
| Ra = $10^5$ | [34.74]<br>34.76<br><33.46><br>(34.73) | [0.855]<br>0.853<br><0.857><br>(0.855) | [68.62]<br>68.64<br><68.40><br>(68.59) | [0.066]<br>0.0656<br><0.0622><br>(0.066) |
| Ra = $10^6$ | [64.83]<br>64.91<br><64.30><br>(64.63) | [0.850]<br>0.847<br><0.851><br>(0.850) | [220.6]<br>220.72<br><218.40><br>(219.36) | [0.0379]<br>0.0375<br><0.0355><br>(0.0379) |

**Table 1:** Comparison of natural convection results.

#### Adaptive Refinement and Variable Resolution

One common approach to improve computational efficiency is to use small node spacing in region of large gradient but large node spacing in smooth region. It is relatively simple to apply these approaches to RBF as there is no dependence on smoothing length. For SPH, extra terms are required in the gradient approximation when $\boldsymbol{h} \equiv \boldsymbol{h}(\boldsymbol{r})$ :

$$\nabla_{\boldsymbol{r}} \boldsymbol{W}(\boldsymbol{u}, \boldsymbol{h}(\boldsymbol{r})) = \frac{1}{\boldsymbol{u}}\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{u}}\boldsymbol{u} + \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{h}}\nabla_{\boldsymbol{r}}\boldsymbol{h}(\boldsymbol{r}) \qquad (11)$$

The second term on the right, known as $\nabla \boldsymbol{h}$ term, is usually neglected in SPH. It is not in general known how much error is involved in ignoring this term. It could be significant especially in region where the variation of $h$ with space is large.

Figure 8 shows the application of adaptive refinement to the 1D Burger's equation. The spacing between nodes is evolved with time to adapt to changes in the solution. The node spacing is halved when the second derivative is greater than 20.

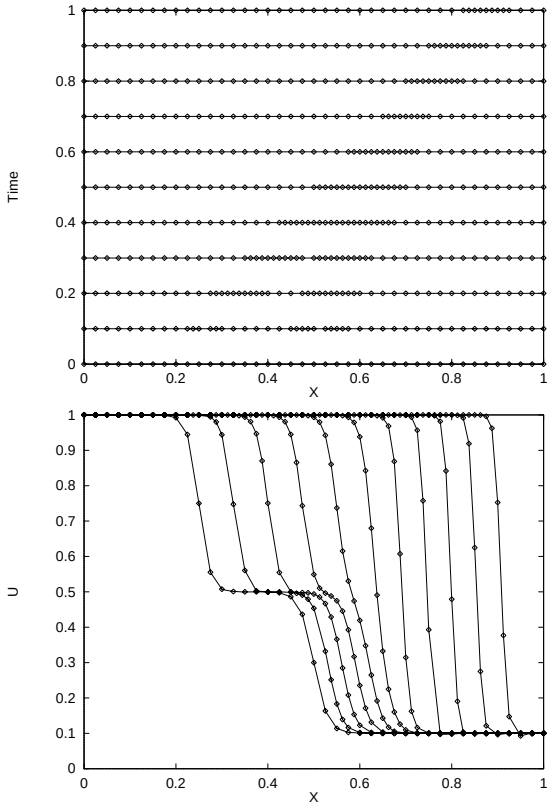


**Figure 8:** Adaptive grid RBF result (diamond) of 1D Burger's equation at various times for viscosity 0.003 compared to analytical solutions (solid line). The top graph shows the positions of grid points at various times.

Next, non-uniform nodal spacing according to the following equation is used to solve the lid-driven cavity problem for Re = 1000 using RBF.

$$\boldsymbol{x_i} = \frac{1}{2}\left(1 - \sqrt{2}\cos\left(\frac{\pi}{4} + \frac{\boldsymbol{i} - 1}{2(\boldsymbol{n} - 1)}\pi\right)\right) \qquad (12)$$

Figure 9 shows the RBF results using non-uniform nodal spacing are now much closer to the published results than the RBF results using uniform nodal spacing shown in Figure 7.

## CONCLUSIONS

This paper presents a unified approach to implement the RBF and SPH methods for numerical computations. The approach of SPH in using the nearest neighbours within the supporting region of a particle to estimate its derivatives of a function is applied to RBF. The size of supporting region depends on the smoothing kernel used in the case of SPH but is a parameter in the case of RBF. In the numerical examples considered in this paper, a supporting region of width 3$h$, $q$ = 1.03 and nodal spacing for $c$ are used for RBF.

The numerical examples presented in the last section demonstrated that GSPH and RBF give accurate results to the problems considered. Unlike conventional SPH, they have the advantage of being able to impose boundary conditions directly. Also, they are just as easy to implement as the conventional SPH. There is no dimensional difference between 1D, 2D and 3D as far as computer coding for their implementation is concerned.

Apart from correcting the boundary deficiency problem, GSPH is less affected by particle disorder than conventional SPH because of the normalisation term in the denominator (refer to Equations 1-3).

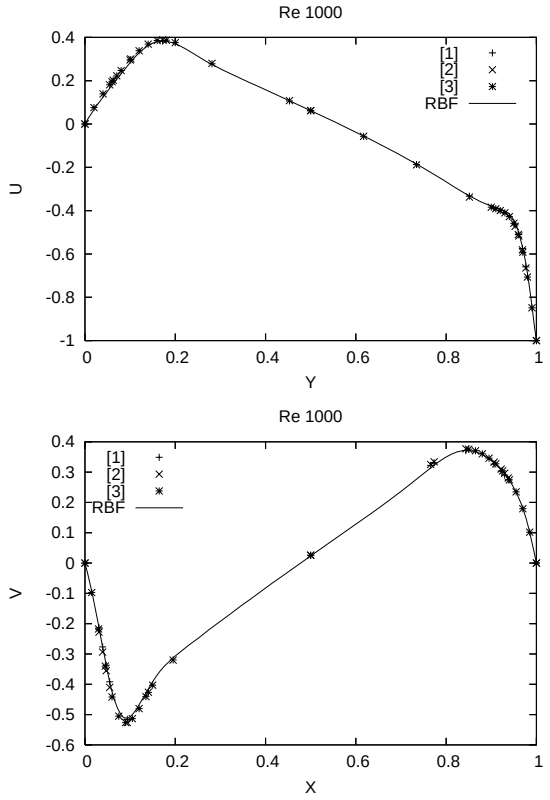


**Figure 9:** Variable resolution RBF results of lid-driven cavity problem for Re = 1000 compared to published solutions.

## REFERENCES


BELYTSCHKO, T., KRONGAUZ, Y., ORGAN, D., FLEMING, M. AND KRYSL, P., 1996, "Meshless methods: An overview and recent developments", *Comput. Meth. Appl. Mech. Engng.*, **139**, 3-47.

BOTELLA, O. AND PEYRET, R., 1998, "Benchmark spectral results on the lid-driven cavity flow", *Computers & Fluids*, **27**, 421-433.

CHANIOTIS, A. K., POULIKAKOS, D. AND KOUMOUTSAKOS, P., 2002. "Remeshed Smoothed Particle Hydrodynamics for the Simulation of Viscous and Heat Conducting Flows", *Journal of Computational Physics*, **182**, 67-90.

CHEN, J.K. AND BERAUN, J.E., 2000, "A generalized smoothed particle hydrodynamics method for nonlinear dynamic problems", *Comput. Methods Appl. Mech. Engrg.*, **190**, 225-239.

DE VAHL DAVIS, G., 1983, "Natural convection of air in a square cavity: a benchmark numerical solution", *Int. J. Numer. Methods Fluids*, **3**, 243-264.

DUBAL, M.R., 1994, "Domain decomposition and local refinement for multiquadric approximations. I: Second-order equations in one-dimension", *Int. J. Appl. Sci. Comput.*, **1**, 146-171.

ERTURK, E., CORKE, T. C. AND GÖKÇÖL, C., 2005, "Numerical solutions of 2-D steady incompressible driven cavity flow at high Reynolds numbers", *International Journal for Numerical Methods in Fluids*, **48**, 747-774.

GHIA, U., GHIA, K.N. AND SHIN, C.T., 1982, "High-Re solutions of incompressible flow using the Navier-Stokes equations and a multigrid method", *Journal of Computational Physics*, **48**, 387-411.

GINGOLD, R. AND MONAGHAN, J., 1977, "Smoothed particle hydrodynamics: theory and application to non-spherical stars", *Mon. Not. Roy. Astron. Soc.*, **181**, 375-389.

JEONG, J.H., JHON, M.S., HALOW, J.S. AND VAN OSDOL, J., 2003, "Smoothed particle hydrodynamics: Applications to heat conduction", *Computer Physics Communications*, **153**, 71-84.

JOHNSON, G.R. AND BEISSEL, S.R., 1996, "Normalized smoothing functions for SPH impact calculations", *Int. J. Numer. Meth. Engng.*, **39**, 2725-2741.

KANSA, E.J., 1990, "Multiquadrics - A scattered data approximation scheme with applications to computational dynamics - I. Surface approximations and partial derivative estimates", *Comput. Math. Appl.*, **19**, 127-145.

KRONGAUZ, Y. AND BELYTSCHKO, T., 1998, "Consistent pseudo-derivatives in meshless methods", *Comput. Meth. Appl. Mech. Engng.*, **146**, 371-386.

LEAL, M.A., PEREZ-GUERRERO, J.S. AND COTTA, R.M., 1999, "Natural convection inside two-dimensional cavities: the integral transform method", *Communications in Numerical Methods in Engineering*, **15**, 113-125.

LIU, W.K., JUN, S., ADEE, J. AND BELYTSCHKO, T., 1995, "Reproducing kernel particle methods for structural dynamics", *Int. J. Numer. Meth. Engng.*, **38**, 1655-1679.

LIU, M.B., LIU, G.R. AND LAM, K.Y., 2003, "Constructing smoothing functions in smoothed particle hydrodynamics with applications", *Journal of Computational and Applied Mathematics*, **155**, 263-284.

LIU, G. R., ZHANG, G. Y., GU, Y. T. AND WANG, Y. Y., 2005. "A meshfree radial point interpolation method (RPIM) for three-dimensional solids", *Computational Mechanics*, **36**, 421-430.

LUCY, L.B., 1977, "A numerical approach to the testing of the fission hypothesis", *J. Astron.*, **82**, 1013-1024.

MONAGHAN, J.J., 1992, "Smoothed particle hydrodynamics", *Ann. Rev. Astron. Astrophys.*, **30**, 543-574.

MORRIS, J.P., FOX, P.J. AND ZHU, Y., 1997, "Modeling low Reynolds number incompressible flows using SPH", *J. Comput. Phys.*, **136**, 214-226.

RANDLES, P.W. AND LIBERSKY, L.D., 1996, "Smoothed particle hydrodynamics: some recent improvement and applications", *Comput. Methods Appl. Mech. Eng.*, **139**, 375-408.

SIGALOTTI, L.D.G., KLAPP, J., SIRA, E., MELEÁN, Y. AND HASMY, A., 2003. "SPH simulations of time-dependent Poiseuille flow at low Reynolds numbers", *J. of Computational Physics*, **191**, 622-638.