CMES, vol.25, no.2, pp.99-113, 2008

Application of Local MQ-DQ Method to Solve 3D Incompressible Viscous Flows with Curved Boundary

Y.Y. Shan[1], C. Shu[1,2] and Z.L. Lu[3]

Abstract: The local multiquadric-based differential quadrature (MQ-DQ) method proposed by [Shu, Ding, and Yeo (2003)] is a natural mesh-free approach for derivative approximation, which is easy to be implemented to solve problems with curved boundary. Previously, it has been well tested for the two-dimensional (2D) case. In this work, this mesh-free method was extended to simulate fluid flow problems with curved boundary in three-dimensional (3D) space. The main concern of this work is to numerically study the performance of the 3D local MQ-DQ method and demonstrate its capability and flexibility for simulation of 3D incompressible fluid flows with curved boundary. Fractional step method was adopted for the solution of Navier-Stokes (N-S) equations in the primitive-variable form. Flow past a sphere with various Reynolds numbers was chosen as a test case to validate the 3D local MQ-DQ method. The computed solution was compared well with available data in the literature. The numerical solution shows that the local MQ-DQ method can be applied to solve incompressible viscous flow problems with curved boundary in 3D space effectively.

Keyword: Local MQ-DQ method; Error estimate; Flow past a sphere

1 Introduction

Radial basis functions (RBFs) are a powerful tool for function interpolation, especially for interpolation of multidimensional scattered data points. Due to their "mesh-free" nature, in the past decade, RBFs have received an increasing attention for solving partial differential equations (PDEs). The first trial of such exploration was made by [Kansa (1990a), Kansa (1990b)]. Subsequently, [Fasshauer (1997)], [Chen, Brebbia, and Power (1998)], [Chen and Tanaka (2002)], [Chen and Hon (2003)], [Larsson and Fornberg (2003)], [Atluri, Han, and Rajendran (2004)], [Ling and Kansa (2005)], [Atluri, Liu, and Han (2006a)], [Atluri, Liu, and Han (2006b)], [Han, Liu, Rajendran, and Atluri (2006)], [Sarler and Vertnik (2006)], [Divo and Kassab (2007)], [Wen and Hon (2007)], [Mai-Duy, Mai-Cao, and Tran-Cong (2007)], [Mai-Duy, Khennane, and Tran-Cong (2007)] also made great contributions in this development. It should be noted that most of above works related to the application of RBFs for the numerical solution of PDEs are actually based on the function approximation instead of derivative approximation. In other words, these works directly substitute the expression of function approximation by RBFs into a PDE, and then change the dependent variables into the coefficients of function approximation. The process is very complicated, especially for non-linear problems. That may be the reason why the method has not so far been extensively applied to solve practical problems.

To resolve this problem and make RBF methods more feasible in solving PDEs, a local method named "local radial basis function-based differential quadrature method (Local RBF-DQ)" has recently been proposed by [Shu, Ding, and Yeo (2003)]. This method adopts the idea of direct derivative approximation through the differential quadrature (DQ) [Bellman and Casti (1971), Bellman, Kashef, and Casti. (1972)] technique, thus

[1] Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore

[2] Corresponding author, Email: mpeshuc@nus.edu.sg

[3] Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People's Republic of China

can be consistently well applied to linear and nonlinear problems. Multiquadric (MQ) RBF [Hardy (1971), Franke (1982)] is selected from the RBFs available to date as the base functions in local RBF-DQ method due to its excellent performance in function interpolation. Thus it is regarded as local MQ-DQ method. In the local MQ-DQ approach, any spatial derivative at a knot is approximated by a linear weighted sum of all the functional values in the local supporting region around the reference knot. The weighting coefficients in the derivative approximation are determined by the function and linear vector space analysis. The local MQ-DQ method is very flexible, and has been applied to solve various kinds of fluid flow problems, such as inviscid compressible flow [Shu, Ding, Chen, and Wang (2005)] and incompressible viscous flows [Shu, Ding, and Yeo (2005), Ding, Shu, Yeo, and Xu (2006), Ding, Shu, Yeo, and Lu (2005)].

Despite the success of local MQ-DQ method in solving fluid flow problems, most work up to now only focused on the two dimensional cases. Very little work has been done to explore its capability of solving three dimensional (3D) fluid flow problems, especially with curved boundary. [Ding, Shu, Yeo, and Xu (2006)] applied the local MQ-DQ method to solve a classical driven cavity flow problem in 3D space and proved the feasibility of this method for 3D fluid flow problems. However, the case they studied is very simple and the geometry of the problem is quite regular (cubic). As we know, the main advantage of RBF-based schemes is their mesh-free property and their flexibility for handling problems with curved boundary. Adopting the local MQ-DQ method to solve the 3D driven cavity flow problem, however, cannot reflect its advantage.

In this paper, we will extend the local MQ-DQ method to solve 3D fluid flow problems with curved boundary. At this time, the theoretical error analysis for derivative approximation by RBF is not available. Thus, firstly, we will make an error estimate for the derivative approximation by the 3D local MQ-DQ method to provide a useful guidance for implementation of this method. Actually, this work is the 3D counterpart of that by [Ding, Shu, and Tang (2005)]. In [Ding, Shu, and Tang (2005)], Ding et al. found the influence of shape parameter and supporting points on numerical accuracy of the 2D local MQ-DQ method through numerical experiments. In this work, while applying the 3D local MQ-DQ method to solve fluid flow problems, we will select shape parameter and supporting points based on the results of the error estimates. The 3D flow past a sphere will be solved by the local MQ-DQ method to demonstrate its capability and flexibility in solving 3D flow problems with curved boundary. This problem has been studied by a number of researchers, thus can be selected as a model problem to test a new numerical method.

The rest of the paper is structured as follows. In Section 2, we will give a description of the local MQ-DQ method. An error estimate of derivative approximation by this method through numerical experiments is made in Section 3. Section 4 shows the problem of flow past a sphere and relevant numerical procedures. The numerical results for the flow past a sphere are discussed in Section 5. Section 6 gives some concluding remarks.

2 Description of Local MQ-DQ Method

As its name implies, the local MQ-DQ method is based on the MQ RBFs and DQ technique. The concept of DQ was first proposed by [Bellman and Casti (1971), Bellman, Kashef, and Casti. (1972)] to approximate the derivative of a smooth function. From the viewpoint of derivative approximation, the essence of the DQ method is that the partial derivative of any dependent variable can be approximated by a weighted linear sum of functional values at all discrete points within its support. In other words, the DQ approximation of the mth order derivative of a function $f(x)$ at x_i can be expressed as

$$\left.\frac{\partial^m f}{\partial x^m}\right|_{x=x_i} = \sum_{j=1}^{N_i} w_{i,j}^{(m)} f(x_j), \tag{1}$$

where x_j denotes the coordinates of discrete points in the support of reference node i, $f(x_j)$ and $w_{i,j}^{(m)}$ are the function values at these points and the related weighting coefficients. It should be noted that the subscript i is a global nodal index which represents the nodes in the domain, while the subscript j is a local index which represents the supporting points for the reference node i. Since the reference node is also one of its supporting points, we define $(\cdot)_{i,1} = (\cdot)_i$ for simplicity. N_i is the number of supporting points within the support for the reference node i. To determine the weighting coefficients $w_{i,j}^{(m)}$ in Eq. (1), a set of trial functions are required. In the present local MQ-DQ method, the MQ RBFs are selected for its mesh-free nature and super convergence.

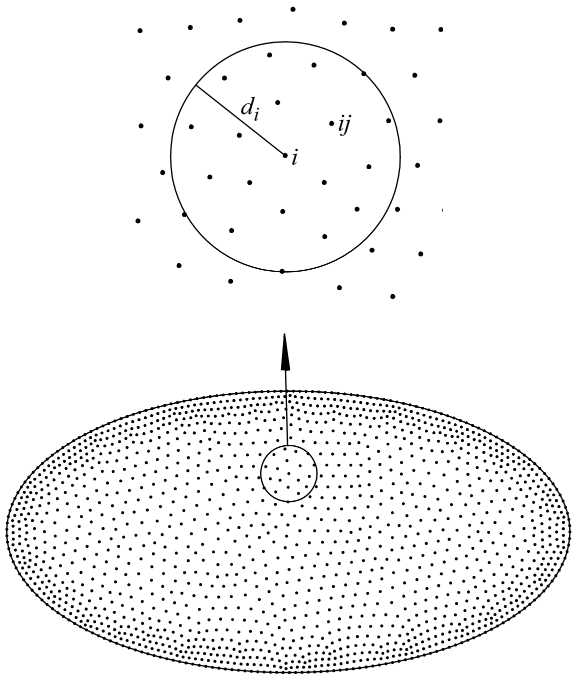


Figure 1: Supporting points around a reference point

Consider a sub-domain Ω_{x_i} as shown in Fig. 1, which forms the neighborhood or support of a reference point x_i and denotes as the domain of function approximation at x_i by the trial functions. To approximate the function u in Ω_{x_i} over a number of randomly-distributed supporting points, the RBF approximant $u^h(x)$, can be defined by a linear sum of MQ RBFs as

$$u^h(x) = \sum_{j=1}^{N_i} \lambda_j \varphi(\|x, x_j\|), \tag{2}$$

where $\varphi(\|x, x_j\|) = \sqrt{r_j^2 + c^2}$ denotes the MQ RBF at x_j, and x_j, $j = 1, 2, \ldots, N_i$, are the supporting points in Ω_{x_i}. r represents the Euclidean norm and c is a positive free shape parameter given by the end-user. N_i denotes the total number of supporting points in Ω_{x_i}. λ_j denotes the corresponding constant coefficients. Eq. (2) can be considered to construct an N_i dimensional linear vector space V_{N_i}. Obviously, the set of MQ RBFs $\varphi(\|x, x_j\|)$, $j = 1, 2, \ldots, N_i$, form the base functions (vectors) in the linear vector space. It can be seen that if all the base functions $\varphi(\|x, x_j\|)$ satisfy a linear relationship like Eq. (1), so does any function in V_{N_i}. Therefore, Eq. (1) can be well applied to discretize the derivatives of function $u(x)$. In the following, the first-order derivative is taken as an example to show the procedure for the determination of weighting coefficients.

Substituting the set of MQ RBFs into Eq. (1), the determination of corresponding coefficients for the first-order derivative is equivalent to solving the following linear equations:

$$\frac{\partial \varphi_k(x_i)}{\partial x} = \sum_{j=1}^{N_i} w_{i,j}^{(1x)} \varphi_k(x_j) \quad k = 1, 2, \ldots, N_i. \tag{3}$$

For simplicity, the notation $\varphi_k(x)$ is adopted to replace $\varphi(\|x, x_k\|)$ in Eq. (3).

The above equations can be further written in the matrix form,

$$\underbrace{\begin{bmatrix} \frac{\partial \varphi_1(x_i)}{\partial x} \\ \frac{\partial \varphi_2(x_i)}{\partial x} \\ \vdots \\ \frac{\partial \varphi_{N_i}(x_i)}{\partial x} \end{bmatrix}}_{\left\{\frac{\partial \varphi(x_i)}{\partial x}\right\}} =$$

$$\underbrace{\begin{bmatrix} \varphi_1(x_1) & \varphi_1(x_2) & \cdots & \varphi_1(x_{N_i}) \\ \varphi_2(x_1) & \varphi_2(x_2) & \cdots & \varphi_2(x_{N_i}) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{N_i}(x_1) & \varphi_{N_i}(x_2) & \cdots & \varphi_{N_i}(x_{N_i}) \end{bmatrix}}_{[A]} \underbrace{\begin{bmatrix} w_{i,1}^{(1x)} \\ w_{i,2}^{(1x)} \\ \vdots \\ w_{i,N_i}^{(1x)} \end{bmatrix}}_{\{w\}} \tag{4}$$

Clearly, there exists a unique solution only if the collocation matrix [A] is non-singular. The non-singularity of the collocation matrix [A] depends on the properties of used RBFs. According to the work of [Micchelli (1986)], matrix [A] is conditionally positive definite for MQ RBFs. This fact cannot guarantee the non-singularity of matrix [A]. [Hon and Schaback (2001)] showed that cases of singularity are quite rare. Therefore, the coefficient vector $\{w\}$ can be obtained by

$$\{w\} = [A]^{-1}\{\frac{\partial \varphi(x_i)}{\partial x}\}. \tag{5}$$

Now, we can use the coefficient vector $\{w\}$ to approximate the first-order derivative in the x-direction for any unknown function at node x_i. The calculation of weighting coefficients for other derivatives can follow the same procedure.

From the procedure of DQ approximation for derivatives, it can be observed that the weighting coefficients are only dependent on the selected RBFs and the distribution of the supporting points in the support domain Ω_{x_i}. During the process of numerical simulation, they are only computed once, and stored for all numerical discretization. Once the coefficients are computed, they will be stored and used to discretize the partial differential equation in a similar manner as in the traditional finite difference (FD) method. It should be noted that the computed coefficients can be consistently well applied to linear and nonlinear problems. Therefore, it is very convenient to use local MQ-DQ method to solve complex non-linear problems such as Navier-Stokes equations in fluid mechanics. From the above, we can also see that the implementation of local MQ-DQ method is very simple and straightforward.

3 Error Estimates of Local MQ-DQ Method in Three-dimensional Space

During the implementation of local MQ-DQ method, it is interesting to know the order of accuracy for the derivative approximation by the 3D local MQ-DQ method. Furthermore, the number of supporting points and free shape parameter c need to be determined by end-users. However, due to the lack of theoretical analysis, there is no error estimate available for derivative approximation by this method to date. Thus, how to determine the number of supporting points and free shape parameter c poses a challenging issue for the end-users.

In this section, we mainly focus on error estimates of the derivative approximation by the 3D local MQ-DQ method to explore the relationship between the approximation error, the number of supporting points and free shape parameter c. Based on the study, we can provide a useful guidance for the implementation of the 3D local MQ-DQ method. The performed analysis of error estimates is based on numerical experiments. In fact, this study is the extension of the work by [Ding, Shu, and Tang (2005)]. In [Ding, Shu, and Tang (2005)], Ding et al. performed error estimates of the second order derivative approximation by the local MQ-DQ method in 2D space through the solution of 2D Poisson equation. They found the relationship between the approximation error, the number of supporting points and free shape parameter c. In this work, numerical experiment is also made to find the relationship between the numerical error of the second order derivative approximation and the two factors.

3D Poisson equation which only has the second order derivatives is taken as a model problem, which can be written as:

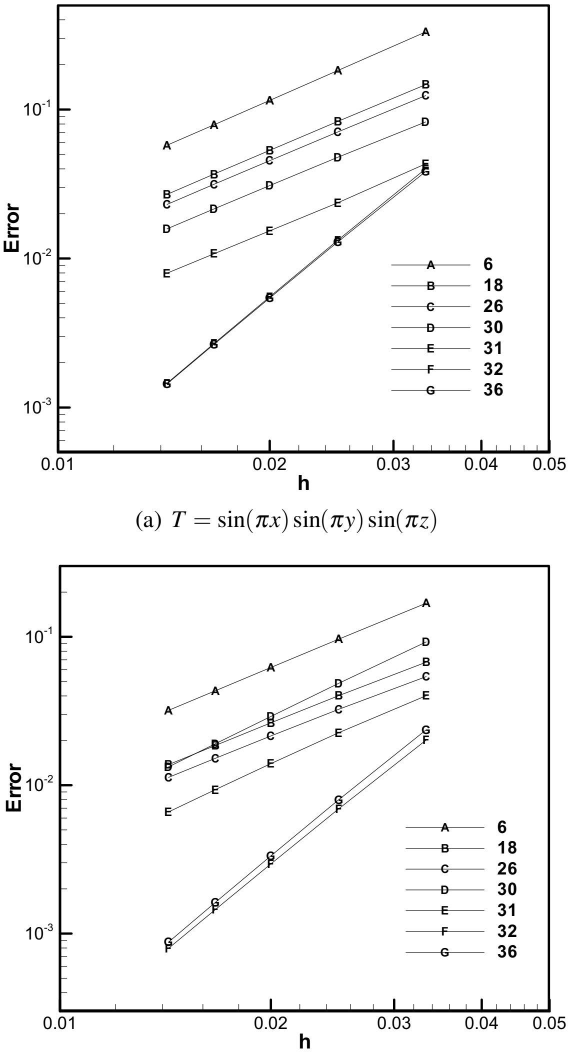
$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = f(x,y,z)$$
$$\text{in } \Omega = \{(x,y,z) | 0 \le x,y,z \le 1\},$$
$$T = g \text{ on } \partial\Omega \tag{6}$$

where f and g are determined in such a manner that the exact solution T of the Poisson equation is the given one. To study the performance of the 3D local MQ-DQ method in simulating two classical types of flow problems: periodic boundary value problems and general boundary value problems, we take $T_1 = \sin(\pi x)\sin(\pi y)\sin(\pi z)$ and $T_2 = x^4 + y^4 + z^4$ as two typical solution functions. Here, T_1 can represent the solution of the periodic boundary value problems and T_2 can stand for the solution of the general boundary value problems.

In order to estimate the approximation error, it is necessary to understand the relationships between the numerical error and the two factors. To fulfill this goal, numerical experiments are designed in such a manner that one factor is fixed and the other one is variable. In this paper, we firstly study the relationship between the numerical error and the number of supporting points. This means that the free shape parameter c is fixed, and through varying the number of supporting points, we can obtain its relationship with the numerical error. Then, we study the relationship between the numerical error and the free shape parameter c. In this process, the number of supporting points is fixed. We get its relationship with numerical error through varying the value of free shape parameter c. The effect of boundary conditions on the accuracy of numerical solution is not considered in this work.

3.1 Relationship between numerical error and number of supporting points

For the investigation of relationship between the numerical error and the number of supporting points n_s, the value of free shape parameter c is fixed as 0.2, which is suitable for all the cases considered. For each n_s, 3D Poisson equation is solved on five different uniform meshes, i.e. $31 \times 31 \times 31, 41 \times 41 \times 41, 51 \times 51 \times 51, 61 \times 61 \times 61$ and $71 \times 71 \times 71$. Thus the convergence rate of relative error versus the mesh size h for each n_s can be studied. The numerical solutions are illustrated in Figure 2 in the form of relative error versus h in the log-log scale.



(a) $T = \sin(\pi x)\sin(\pi y)\sin(\pi z)$

(b) $T = x^4 + y^4 + z^4$

Figure 2: Numerical errors versus mesh size for various number of supporting points

It can be observed in Figure 2 that the symbols representing the accuracy of solution with the same n_s are in perfect alignment. This implies that the 3D local MQ-DQ method accomplishes the so-called super convergence, i.e. an error estimate of $O(h^n)$. Furthermore, from Figure 2, we can also see that the convergence lines can be classified into two groups by the value of slope, with the number of supporting points ranging from 6 to 36. Specifically, the convergence rate is approximately 2.0 for the scheme with less than 31 supporting points and 3.9 for the scheme with 32 and 36 supporting points. Therefore, an error estimate with respect to the mesh size h and the number of supporting points can be written as

$$\|\varepsilon\| \sim O(h^n) \text{ and } n \approx \begin{cases} 2.0 & \text{for } 6 \le n_s \le 31 \\ 3.9 & \text{for } 32 \le n_s \le 36 \end{cases} \tag{7}$$

The above results show that the accuracy of the second order derivative approximation can be greatly improved at some critical number of supporting points. The mean values of convergence rate for each n_s are listed in Table 1. From Table 1, it can be seen that the convergence rates are independent of solution functions. In our work,

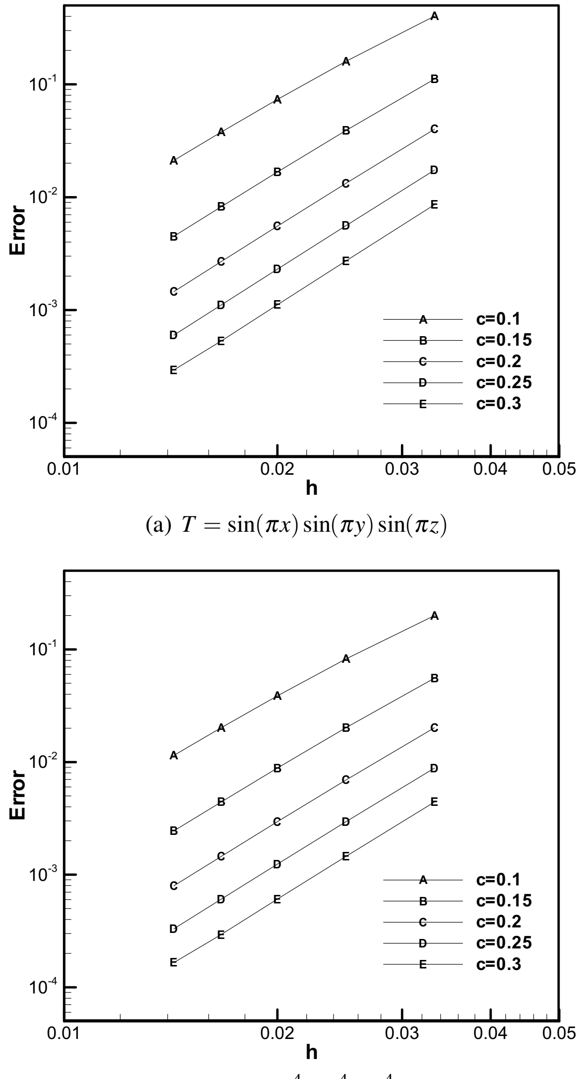
Table 1: Mean value of convergence rate with shape parameter $c = 0.2$

Number of supporting points	$T = \sin(\pi x)\sin(\pi y)\sin(\pi z)$	$T = x^4 + y^4 + z^4$
$n_s = 6$	2.06	1.97
$n_s = 18$	2.02	1.91
$n_s = 26$	2.00	1.89
$n_s = 30$	1.97	2.32
$n_s = 31$	1.94	2.20
$n_s = 32$	3.95	3.87
$n_s = 36$	3.92	3.95

the maximum n_s tested is 36. This is because in practical applications, the n_s value is usually taken below 36.

3.2 *Relationship between numerical error and free shape parameter c*

To study the relationship between numerical error and the free shape parameter c, the number of supporting points is fixed at 32, based on the above observation. Five different values of free shape parameter c are tested, ranging from 0.1 to 0.3. For each c, 3D Poisson equation is solved on five different uniform meshes. The numerical solutions are shown in Figures 3(a)-(b) for the two solution functions, respectively. In each figure, the numerical results are plotted in the form of relative error versus mesh size in the log-log scale and a group of convergence lines are drawn according to the same value of shape parameter c. The mean values of convergence rate of relative error versus the mesh size h are listed in Table 2. From Figures 3(a)-(b), it is clear to see that the convergence lines standing for different shape parameters are parallel to each other. In other words, they have the same convergence rate (order of accuracy). This can also be confirmed by the mean value of convergence rate listed in Table 2. Therefore, from the viewpoint of convergence rate, we can say that the contributions of shape parameter c and mesh size h are utterly independent. From Table 2, it can be seen that the convergence rates for the different solution functions with various shape parameter c almost have the same value (≈ 3.9). This confirms our previous finding for the convergence property with respect to mesh size h, i.e. the convergence rate with respect to mesh size h is independent of the solution function.



(a) $T = \sin(\pi x)\sin(\pi y)\sin(\pi z)$

(b) $T = x^4 + y^4 + z^4$

Figure 3: Numerical errors versus mesh size for various shape parameter c

Table 2: Mean value of convergence rate with number of supporting points $n_s = 32$

Shape parameter	$T = \sin(\pi x)\sin(\pi y)\sin(\pi z)$	$T = x^4 + y^4 + z^4$
$c = 0.1$	3.61	3.53
$c = 0.15$	3.86	3.77
$c = 0.2$	3.95	3.87
$c = 0.25$	3.99	3.91
$c = 0.3$	3.98	3.88

4 Numerical Procedure for Simulating Flows past a Sphere

To demonstrate the ability of local MQ-DQ method for simulation of a flow with curved boundary in 3D space, we apply it to simulate laminar flow past a sphere for $20 \le Re \le 270$. The Reynolds number Re is based on the sphere diameter D and the free stream velocity U. The flow of a viscous fluid past a stationary isolated sphere may be considered as a simplified case of a general family of immersed bluff-body flows with widespread applications. The behavior of the flow past a sphere at different Reynolds numbers has been studied by a number of researchers. In our study, we can easily compare our results with those in the literature to validate the 3D local MQ-DQ method.

4.1 Flow domain and grid distribution

In this case, the sphere is located at the origin of the Cartesian coordinate system in a rectangular box, $-5 \le x \le 10$, $-5 \le y \le 5$ and $-5 \le z \le 5$. The grid distribution is shown in Figure 4, in which 510768 nodes are used, and the minimum and maximum distances between two neighboring nodes are respectively 0.025 and 0.125. For the convenience of illustration, only the grid points on the $x-y$ plane at $z = 0$ are plotted. From this figure, it can be seen that a Cartesian mesh is generated as the background mesh. For the domain around the sphere, grid points are distributed along concentric circles to make the points orthogonal to the solid boundary. As shown in Fig. 4, for the region between the inner circular domain and outer square domain, grid points are generated by the software “GAMBIT” to make the grid distribution smooth.

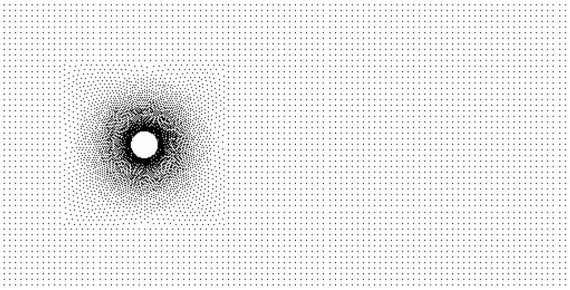


Figure 4: Grid point distribution on the $x-y$ plane at $z = 0$

4.2 Governing equations

The three-dimensional, unsteady, incompressible Navier-Stokes equations in the primitive variable form are taken as the governing equations, which are written as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{\partial p}{\partial x} + \frac{1}{Re}(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}) \quad (8)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{\partial p}{\partial y} + \frac{1}{Re}(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}) \quad (9)$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{\partial p}{\partial z} + \frac{1}{Re}(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}) \quad (10)$$

$$\frac{\partial u}{\partial \mathrm{x}} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (11)$$

where u, v and w are the velocity components along x-direction, y-direction and z-direction, respectively.

In order to solve the above Navier-Stokes equations, special splitting technique is required to deal with the difficulties arising from lack of an independent equation for the pressure, whose gradient contributes to the momentum equations. In this study, fractional step method proposed by [Chorin (1968)] is adopted to do the job. In addition, the local MQ-DQ method is used for spatial discretization.

4.3 Fractional step method

For convenient illustration of the fractional step method, governing equations (8)-(11) are written in the vector form as

$$\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}=-\nabla p+\frac{1}{Re}\Delta\mathbf{u} \tag{12}$$

$$\nabla\cdot\mathbf{u}=0 \tag{13}$$

Solution of above N-S equations encounters difficulties like the lack of an independent equation for the pressure and non-existence of a dominant variable in the continuity equation. One way to circumvent these difficulties is to uncouple the pressure computation from the momentum equations and then construct a pressure field so as to enforce the satisfaction of the continuity equation. In this work, a two step fractional step formulation is applied for 3D N-S equations with a collocated/ non-staggered arrangement. In this scheme, the solution is advanced from time level "n" to "$n+1$" through a predicting advection-diffusion step where pressure term is dropped from the momentum equations. In the advection-diffusion equations, convective and diffusive terms are discretized by using Crank-Nicolson scheme.

For a time increment $\Delta t=t^{n+1}-t^n$, the algorithm of fractional step method consists of two steps: Firstly, an intermediate velocity $\mathbf{u}^*$ is predicted by the advection-diffusion equation, which drops the pressure term. That is, for each interior node in the domain, the intermediate velocity $\mathbf{u}^*$ can be calculated by

$$\frac{\mathbf{u}^*-\mathbf{u}^n}{\Delta t}=\frac{1}{Re}\frac{1}{2}(\Delta\mathbf{u}^*+\Delta\mathbf{u}^n)-\frac{1}{2}(\mathbf{u}^*\cdot\nabla\mathbf{u}^*+\mathbf{u}^n\cdot\nabla\mathbf{u}^n) \tag{14}$$

Secondly, the complete velocity $\mathbf{u}$ at t^{n+1} is corrected by including the pressure field, given by

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^*}{\Delta t}=-\nabla p^{n+1} \tag{15}$$

The final velocity field is subject to the continuity constraint given by

$$\nabla\cdot\mathbf{u}^{n+1}=0 \tag{16}$$

Substituting Eq. (15) into Eq. (16) leads to the following Poisson equation for pressure

$$\Delta p^{n+1}=\frac{\nabla\cdot\mathbf{u}^*}{\Delta t} \tag{17}$$

The velocity $\mathbf{u}^{n+1}$ is updated by the solution of pressure equation (17) and equation (15).

4.4 Implementation of Boundary conditions

The physical boundary conditions of the problem are specified as follows. Free stream boundary conditions ($u=1$, $v=w=0$) are applied at the inflow and transverse boundaries. And Neumann boundary conditions are imposed at the outflow boundary. On the solid boundary, the non-slip boundary conditions ($u=v=w=0$) are imposed.

Besides the above physical boundary conditions, some other boundary conditions are worthy of attention. One of them is the enforcement of continuity equation on the solid boundary. To achieve this, continuity equation should be accurately enforced on the solid boundary, i.e.

$$\frac{\partial(\mathbf{u}\cdot\mathbf{n})}{\partial n}=0 \tag{18}$$

where n is the normal direction to the boundary surface.

For generality and consistency, the implementation of above Neumann boundary condition is carried out by using the so-called locally orthogonal "grid". As shown in [Shu, Ding, and Yeo (2003)], the reason of adopting this type of grid is for easy implementation. With the help of locally orthogonal "grid", the Neumann boundary condition for the velocity components can be discretized by one-side finite difference scheme. Then, this discretized boundary condition is used to update the corresponding velocity component at immediate interior point of the boundary knot (not the boundary knot itself), since the velocity on the boundary is already known.

The implementation of this boundary condition is shown below. For the convenience of illustrating the procedure clearly, a sketch is shown in Figure 5. For point 1, its original velocity vector is $\{u_1,\ v_1\}$ and its unit normal vector is $\{\frac{x}{r},\ \frac{y}{r}\}$. Thus, $U_{n1} = u_1\frac{x}{r} + v_1\frac{y}{r}$. Then we can get the normal and tangent velocity vector to be $\{U_{n1}\frac{x}{r},\ U_{n1}\frac{y}{r}\}$, $\{u_1 - U_{n1}\frac{x}{r},\ v_1 - U_{n1}\frac{y}{r}\}$, respectively. Similarly, $U_{n2} = u_2\frac{x}{r} + v_2\frac{y}{r}$.

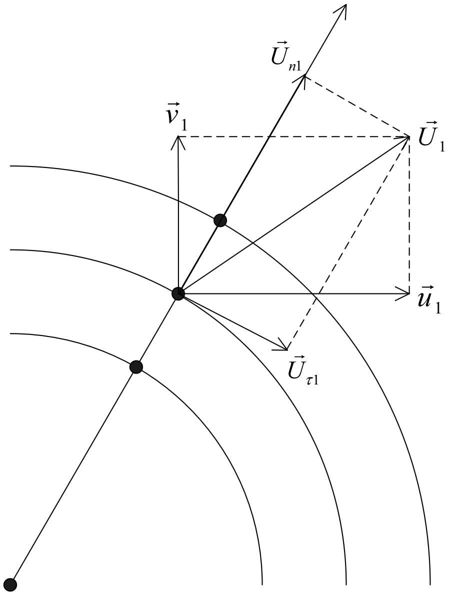


Figure 5: Sketch of the enforcement of continuity equation on the solid boundary

From the boundary condition $\frac{\partial(\mathbf{u}\cdot\mathbf{n})}{\partial n} = 0$, we get $\frac{\partial U_n}{\partial n} = 0$. Discretizing it by one-side FD scheme, we have

$$\left.\frac{\partial U_n}{\partial n}\right|_0 = \frac{dr_2^2\cdot U_{n1} - dr_1^2\cdot U_{n2} - (dr_2^2 - dr_1^2)U_{n0}}{dr_2^2\cdot dr_1 - dr_1^2\cdot dr_2} = 0 \quad (19)$$

Rearranging the above equation, we can get

$$U_{n1} = \frac{(dr_2^2 - dr_1^2)U_{n0} + dr_1^2\cdot U_{n2}}{dr_2^2} \quad (20)$$

Since $U_{n0} = 0$, U_{n1} can be simplified to be

$$U_{n1} = \frac{dr_1^2}{dr_2^2}U_{n2} \quad (21)$$

Thus, the normal velocity at point 1 is updated to be $U'_{n1} = \frac{dr_1^2}{dr_2^2}U_{n2}$ and its corresponding vector becomes $\{U'_{n1}\frac{x}{r},\ U'_{n1}\frac{y}{r}\}$. Since the tangent velocity vector of point 1 stays unchanged, the new velocity vector is updated to be:

$$\left\{U'_{n1}\frac{x}{r},\ U'_{n1}\frac{y}{r}\right\} + \left\{u_1 - U_{n1}\frac{x}{r},\ v_1 - U_{n1}\frac{y}{r}\right\} = \left\{u_1 + (U'_{n1} - U_{n1})\frac{x}{r}, v_1 + (U'_{n1} - U_{n1})\frac{y}{r}\right\} \quad (22)$$

From the above derivation, we can easily update the velocity at point 1 by

$$u'_1 = u_1 + (U'_{n1} - U_{n1})\frac{x}{r} \quad (23)$$

$$v'_1 = v_1 + (U'_{n1} - U_{n1})\frac{y}{r} \quad (24)$$

where $U_{n1} = u_1\frac{x}{r} + v_1\frac{y}{r}$, $U'_{n1} = \frac{dr_1^2}{dr_2^2}U_{n2}$ and $U_{n2} = u_2\frac{x}{r} + v_2\frac{y}{r}$.

The other concern is the boundary conditions for pressure equation. In general, there is no explicit boundary condition for pressure on the solid surface. However, it can be derived from the momentum equations at the boundary node. For the present case, the pressure gradient normal to the boundary at the boundary nodes can be expressed as

$$\frac{\partial p}{\partial n} = \nabla p\cdot\vec{n} = \begin{pmatrix}\frac{\partial p}{\partial x} & \frac{\partial p}{\partial y} & \frac{\partial p}{\partial z}\end{pmatrix}\cdot\begin{pmatrix}\frac{x}{r}\\ \frac{y}{r}\\ \frac{z}{r}\end{pmatrix} = \frac{\partial p}{\partial x}\frac{x}{r} + \frac{\partial p}{\partial y}\frac{y}{r} + \frac{\partial p}{\partial z}\frac{z}{r} \quad (25)$$

where $\frac{\partial p}{\partial x} = \frac{1}{Re}\nabla^2 u$, $\frac{\partial p}{\partial y} = \frac{1}{Re}\nabla^2 v$ and $\frac{\partial p}{\partial z} = \frac{1}{Re}\nabla^2 w$

4.5 Solution Procedure

The solution procedure of applying the fractional step method to solve the 3D N-S equations is shown below:

(1) The advection-diffusion equation (14) is solved by SOR method to obtain the intermediate velocity $\mathbf{u}^*$. (Boundary conditions for the intermediate velocity are chosen to be the same as the physical boundary conditions).

(2) The pressure Poisson equation (17) is solved by SOR method to obtain the pressure for the next step.

(3) Update the pressure boundary conditions (25) on the solid boundary.

(4) Update the velocity in the domain for the next step through equation (15). (Velocity on the boundary is fixed)

(5) Enforce the continuity equation on the solid boundary to update the velocity nearby the boundary through equation (19a-19b).

(6) Go to step (1) until a solution of steady state is reached.

5 Results and Discussion

The problem of flow past a sphere has been studied by [Johnson and Patel (1999)] in details. They found that when $20 \leq Re \leq 210$, the flow is separated, steady, axisymmetric and topologically similar. And when $210 \leq Re \leq 270$, although the flow still remains steady state, it is non-axisymmetric. In this paper, the flows with Reynolds numbers of 50, 100, 150 and 200 are studied as axisymmetric case and Reynolds number of 250 as a non-axisymmetric case.

In the local MQ-DQ method, 32 supporting points are employed for every reference node and the shape parameter is set to be 0.18, based on the observation in Section 3. The time step for temporal discretization is set to be 0.01. For the convergence criterion of steady flow, L_2 norm of velocity difference between the new and old time levels, i.e. $\left\|u^{n+1}-u^n\right\|$, is set to be less than 10^{-4}, which is considered to be small enough for a converged solution.

5.1 *Steady axisymmetric flow*

For flow in this regime, streamlines are shown in Figure 6, which displays the streamlines on the (x, y)-plane at $z = 0$ for Reynolds numbers of 50, 100, 150 and 200. In these and all the following figures, unless otherwise noted, the flow direction is from left to right. From Figure 6, it can be seen that the flow separates from the surface of the sphere at a separation angle and rejoins at a point on the axis of the flow to form a closed separation bubble and a toroidal vortex. It can also be seen from Figure 6 that the flow structure for all the four Reynolds numbers remains topologically the same with changes only in the separation angle, the vortex's position and the separation bubble length. To illustrate the accuracy of the present results, the separation bubble length L_{sep} together with the drag coefficient C_d obtained by the local MQ-DQ method is compared with those by [Johnson and Patel (1999)]. The drag coefficient is defined as $C_d = F_x/(\frac{1}{2}\rho U_\infty^2 \pi r^2)$, where F_x is the force in the streamwise direction. Table 3 shows good agreement between the present results and the aforementioned results.

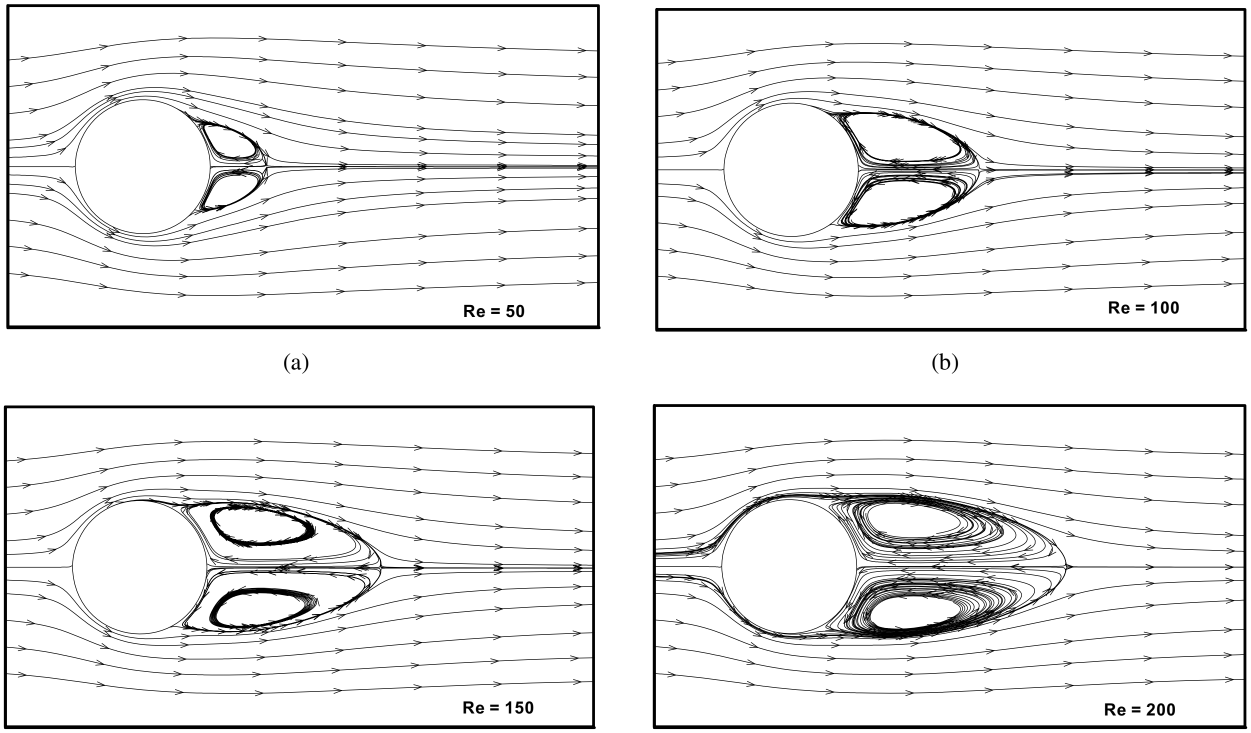
Pressure contours for Reynolds numbers of 50, 100, 150 and 200 are shown in Figure 7. Contours are drawn for every 0.05 increment with dashed lines used for negative values. By comparing the pressure contours, we can see that, within the wake, in the vicinity of the toroidal vortex shown in Figure 6, there is no pressure minimum in the symmetry plane until a Reynolds number of 200. As shown in Figure 7(d), the closed circles in the wake indicate a pressure minimum located very near the center of the toroidal vortex.

5.2 *Steady non-axisymmetric flow*

As has been pointed out by [Johnson and Patel (1999)], at a Reynolds number of 211, the calculated flow solution no longer exhibits axial symmetry. The flow does, however, remain steady state. Although non-axisymmetric, the flow does contain a plane of symmetry. The extent of this steady non-axisymmetric range has been documented as approximately $210 < Re < 270$. The results within this regime were found to be essentially self-similar, or topologically identical, thus only the solution at $Re = 250$ is considered in the paper of [Johnson and Patel (1999)] since the departure from symmetry is quite pronounced. In

Table 3: Comparison of recirculating length and drag coefficient for Re = 50, 100, 150 and 200

Re	Source	L_{sep}	C_d
50	Present	0.413 0.418	1.564 1.583
	[Johnson & Patel (1999)]		
100	Present	0.896 0.884	1.103 1.112
	[Johnson & Patel (1999)]		
150	Present	1.292 1.214	0.901 0.908
	[Johnson & Patel (1999)]		
200	Present	1.552 1.463	0.786 0.798
	[Johnson & Patel (1999)]		



(a) (b)

(c) (d)

Figure 6: Calculated axisymmetric streamlines past the sphere

this study, we also only consider the case with *Re* = 250 for the convenience of comparison.

Figures 8(a) and 8(b) show streamlines in the (x, z)- and (x, y)- planes, respectively. It is clear from Figure 8(a) that the flow field is symmetric about the (x, z)-plane, which divides the figure across the center. Comparatively, from Figure 8(b), it is apparent that the toroidal vortex has tilted. It is also clear from the difference between the top and bottom of the vortex ring, that its size is not constant in the azimuthal direction. Additionally, the toroid is clearly no longer a closed separation bubble. The upper spiral is actually fed by fluid originating from upstream while the lower spiral releases fluid into the wake after sending it up and around the upper spiral.

Pressure contours for the (x, z)- and (x, y)-planes are shown in Figure 9. Like Figure 7, the contours are drawn in levels of 0.05. The pressure field in the (x, z)-plane is completely symmetric and closely resembles the contours in Figure 7(d) for a Reynolds number of 200, although the pressure in the core of the vortex is clearly lower owing to the higher centrifugal acceleration of the vortex.

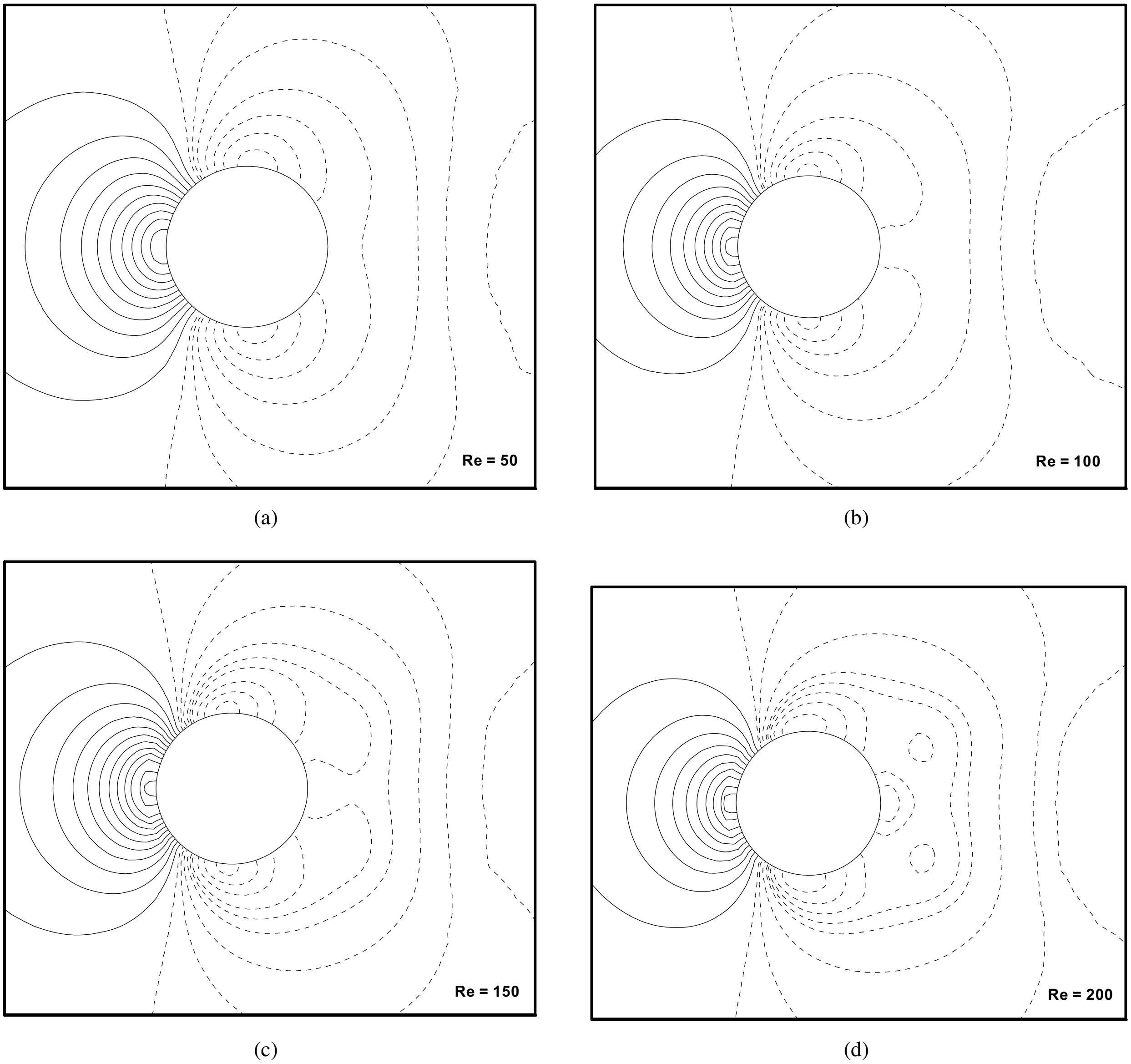


Figure 7: Pressure contours for axisymmetric flow

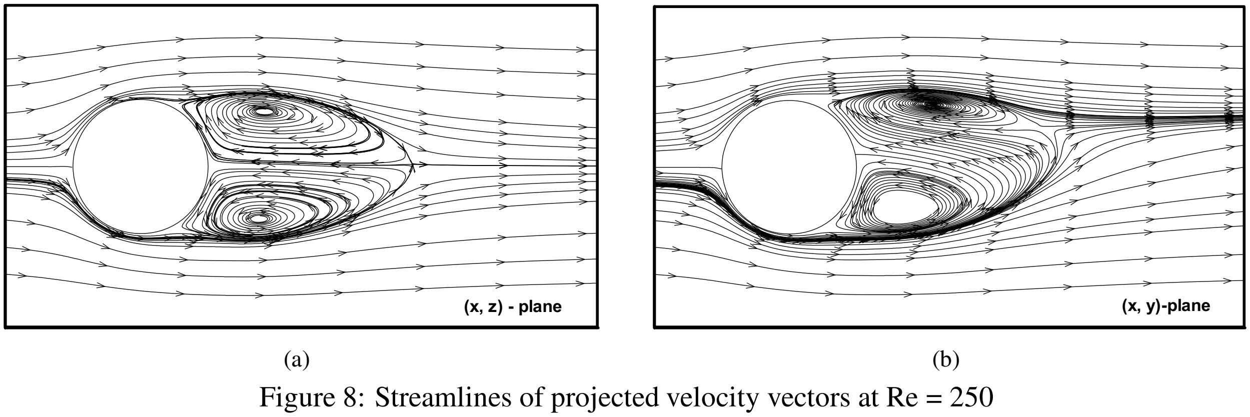


Figure 8: Streamlines of projected velocity vectors at Re = 250

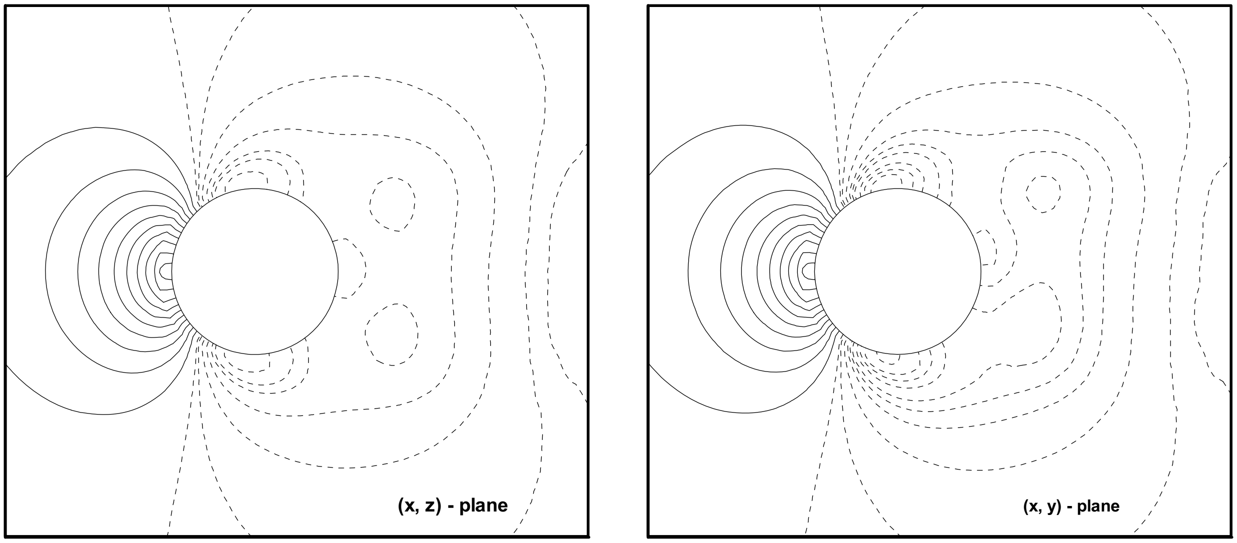


Figure 9: Pressure contours Re = 250

Pressure contours in the (x ,y)-plane are not symmetric. The pressure minimum in the region of the lower focus of Figure 8(b) is lower than that in the region of the upper focus. This azimuthal pressure gradient propagates through the core of the toroidal vortex inducing flow along the vortex axis. This breakdown in axial symmetry, which begins at $Re \approx 211$, corresponds closely to the occurrence of a global pressure minimum in the center of the vortex. It appears, therefore, that the instability of the axisymmetric flow is connected to the generation, by radial acceleration around the vortex center, of a ring of low pressure in the wake.

6 Conclusions

In this paper, the local MQ-DQ method was extended to solve fluid flow problems with curved boundary in three-dimensional space. An error estimate was provided for the 3D local MQ-DQ method to study the influence of the number of supporting points and shape parameter on its numerical accuracy. It was observed that the accuracy of numerical solutions can be improved by increasing the value of shape parameter and the convergence rate can be improved by increasing the number of supporting points. Based on these findings, the local MQ-DQ method is then applied to solve three-dimensional, time-dependent, incompressible Navier-Stokes equations in the primitive variable form. The problem of flow past a sphere was simulated to demonstrate its capability and flexibility in solving 3D fluid flow problems with curved boundary. The obtained numerical results were compared well with data in the literature. The numerical experiments show that the local MQ-DQ method is a promising scheme for solving 3D fluid flow problems with curved boundary.

Acknowledgement: This work was supported by Academic Research Fund (AcRF Tier 2) of Ministry of Education of Singapore (R-265-000-182-112).

References

Atluri S. N.; Han Z. D.; Rajendran A. M. (2004): A new implementation of the meshless finite volume method, through the MLPG "Mixed" approach, *CMES: Computer Modeling in Engineering & Sciences* 6 (6): 491-513.

Atluri S. N.; Liu H. T.; Han Z. D. (2006a): Meshless Local Petrov-Galerkin (MLPG) Mixed Finite Difference Method for Solid Mechanics, *CMES: Computer Modeling in Engineering & Sciences*, 15 (1)1-16.

Atluri S. N.; Liu H. T.; Han Z. D. (2006b): Meshless Local Petrov-Galerkin (MLPG) Mixed

Collocation Method for Elasticity Problems, *CMES: Computer Modeling in Engineering & Sciences*, 14 (3) 141-152.

Bellman, R.E.; Casti, J. (1971): Differential quadrature and long term integration, *J. Math. Anal. Appl.* **34** 235-238.

Bellman, R. E.; Kashef, B.G.; Casti, J. (1972): Differential quadrature: a technique for the rapid solution of nonlinear partial differential equations, *J. Comput. Phys.* **10** 40-52.

Chen, C. S.; Brebbia, C.A.; Power, H. (1998): Dual reciprocity method using for Helmholtz-type operators, *Boundary Elements*, **20** 495-504.

Chen, W.; Tanaka, M. (2002): A meshless, integration-free, and boundary-only RBF technique, *Comput. Math. Appl.*, **43** 379-391.

Chen W.; Hon, Y. C. (2003): Numerical convergence of boundary knot method in the analysis of Helmholtz, modified Helmholtz, and convection-diffusion problems, *Comput. Methods Appl. Mech. Engrg.*, **192**, 1859-1875.

Chorin, A. J. (1968): Numerical solution of the Navier-Stokes equations, *Math. Comput.* **22** 745-762.

Ding, H.; Shu, C.; Yeo, K. S.; Xu D. (2006): Numerical computation of three-dimensional incompressible viscous flows in the primitive variable form, *Comput. Methods Appl. Mech. Engrg.* **195** 516-533.

Ding, H.; Shu, C.; Yeo, K. S.; Lu, Z. L. (2005): Simulation of Natural Convection in Eccentric Annuli Between A Square Outer Cylinder and a Circular Inner Cylinder Using Local MQ-DQ Method, *Numerical Heat Transfer*, **47** 291-313.

Ding, H.; Shu, C.; Tang, D. B. (2005): Error Estimates of Local Multiquadric-based Differential Quadrature (LMQDQ) Method through Numerical Experiments, *Int. J. Numer. Meth. Engng*, **63** 1513-1529.

Divo, E.; Kassab, A. J. (2007): An efficient localized radial basis function meshless method for fluid flow and conjugate heat transfer, *ASME Journal of Heat Transfer*, **129** 124-136.

Fasshauer, G. E. (1997): Solving partial differential equations by collocation with radial basis functions, Surface fitting and multiresolution methods, eds. A.L. Mehaute, C. Rabut and L. L. Schumaker, 131-138.

Franke, R. (1982): Scattered data interpolation: tests of some methods, *Math. Comp.* **38** 181-199.

Han Z. D.; Liu H. T.; Rajendran A. M.; Atluri S. N. (2006): The Applications of Meshless Local Petrov-Galerkin (MLPG) Approaches in High-Speed Impact, Penetration and Perforation Problems, *CMES: Computer Modeling in Engineering & Science*s, 14 (2) 119-128.

Hardy, R. L. (1971): Multiquadric equations of topography and other irregular surfaces, *J. Geophys. Res.* **76** 1905-1915.

Hon, Y. C.; Schaback, R. (2001): On unsymmetric collocation by radial basis functions. *Applied Mathematics and Computation*, **119** 177-186.

Johnson, T. A.; Patel, V. C. (1999): Flow past a sphere up to a Reynolds number of 300, *J. Fluid Mech.* **378** 19-70.

Kansa, E. J. (1990a): Multiquadrics—A scattered data approximation scheme with application to computational fluid dynamics—I. Surface approximations and partial derivative estimates, *Comput. Math. Appl.* **19** 127-145.

Kansa, E. J. (1990b): Multiquadrics—A scattered data approximation scheme with application to computational fluid dynamics—II. Solutions to parabolic, hyperbolic, and elliptic partial differential equations, *Comput. Math. Appl.* **19** 147-161.

Larsson, E.; Fornberg, B. (2003): A numerical study of some radial basis function based solution methods for elliptic PDEs, *Comput. Math. Appl.* **46** 891-902.

Ling, L.; Kansa, E. J. (2005): A least-squares preconditioner for radial basis functions collocation methods, *Adv. Comput. Math.* **23** 31-54.

Mai-Duy, N.; Mai-Cao, L.; Tran-Cong, T. (2007): Computation of transient viscous flows using indirect radial basis function networks, *CMES: Computer Modeling in Engineering & Sciences*, Vol. 18, No. 1, pp. 59-78.

Mai-Duy, N.; Khennane, A.; Tran-Cong, T. (2007): Computation of Laminated Composite Plates using Integrated Radial Basis Function

Networks, *CMC: Computers, Materials, & Continua*, Vol. 5, No. 1, pp. 63-78.

Micchelli, C. A. (1986): Interpolation of scattered data: Distance matrices and conditionally positive definite functions, *Constr. Approx.* **2** 11-22.

Sarler, B.; Vertnik, R. (2006): Meshfree explicit local radial basis function collocation method for diffusion problems, *Computers and Mathematics with Applications*, **51** 1269-1282.

Shu, C; Ding, H.; Yeo, K. S. (2003): Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Engrg*. **192** 941-954.

Shu, C.; Ding, H.; Chen, H. T.; Wang T. G. (2005): An upwind local RBF-DQ method for simulation of inviscid compressible flows, *Comput. Methods Appl. Mech. Engrg*. **194** 2001-2017.

Shu, C.; Ding, H.; Yeo, K. S. (2005): Computation of incompressible Navier-Stokes equations by local RBF-based differential quadrature method, *CMES-Computer Modeling in Engineering and Sciences*, **7** 195-205.

Wen, P. H.; Hon, Y. C. (2007): Geometrically Nonlinear Analysis of Reissner-Mindlin Plate by Meshless Computation, *CMES: Computer Modeling in Engineering & Scienc*es, Vol. 21, No. 3, pp. 177-192.