entropy

MDPI

Article

Radial Basis Function Finite Difference Method Based on Oseen Iteration for Solving Two-Dimensional Navier–Stokes Equations

Liru Mu and Xinlong Feng *

College of Mathematics and System Sciences, Xinjiang University, Urumqi 830017, China; muliru2021@163.com
* Correspondence: fxlmath@xju.edu.cn; Tel.: +86-991-858-5505

Abstract: In this paper, the radial basis function finite difference method is used to solve two-dimensional steady incompressible Navier–Stokes equations. First, the radial basis function finite difference method with polynomial is used to discretize the spatial operator. Then, the Oseen iterative scheme is used to deal with the nonlinear term, constructing the discrete scheme for Navier–Stokes equation based on the finite difference method of the radial basis function. This method does not require complete matrix reorganization in each nonlinear iteration, which simplifies the calculation process and obtains high-precision numerical solutions. Finally, several numerical examples are obtained to verify the convergence and effectiveness of the radial basis function finite difference method based on Oseen Iteration.

Keywords: Navier–Stokes equation; radial basis function finite difference method; polynomial; Oseen iteration

Citation: Mu, L.; Feng, X. Radial Basis Function Finite Difference Method Based on Oseen Iteration for Solving Two-Dimensional Navier–Stokes Equations. *Entropy* **2023**, *25*, 804. https://doi.org/10.3390/e25050804

Academic Editor: José F. F. Mendes

Received: 20 March 2023
Revised: 16 April 2023
Accepted: 18 April 2023
Published: 16 May 2023

1. Introduction

The Navier–Stokes equations are a set of equations used to describe fluid substance such as liquid and air, and can be used to simulate weather, ocean currents, water flows in pipes, the motion of stars in galaxies, etc. Therefore, they are of significant research value; this paper discusses the numerical solution method of steady incompressible Navier–Stokes equations.

Scholars have studied many numerical methods for Navier–Stokes equations [1]. For unsteady Navier–Stokes problems [2] and steady Navier–Stokes problems [3], a variety of traditional numerical methods have been proposed, including the finite element method, finite difference method, and finite volume method. In response to incompressible conditions, nonlinearity, long-time integration, and other difficulties encountered in numerical solutions of three-dimensional unsteady Navier–Stokes equations, this paper discusses the research status and latest research results of highly efficient and fully discrete finite element methods, which can be used to overcome these difficulties. In addition, this paper illustrates stability and error estimation of finite element-space discrete solutions and optimal error estimation of efficient fully-discrete finite element solutions for solving three-dimensional unsteady Navier–Stokes equations [4]. These methods are mesh-based, and the generation of meshes increases computational cost; moreover, the computational domain and the technology used for mesh quality and stabilization affect the computational accuracy. Especially for high-dimensional problems, it is difficult and costly to generate good meshes. Therefore, in recent years many scholars have introduced meshless methods [5,6] for solving partial differential equations, among which the radial basis function method [7] is increasingly popular.

The radial basis function method has been widely used for solving various partial differential equations, with the radial basis function finite difference method (RBF-FD) proposed considering ill-conditioned problems with a high occurrence rate in the matrix generated by the global radial basis function method. In this way, it is easy to realize

RBF-FD discretization without needing to use a mesh at all. The RBF-FD method can be regarded as an extension of the standard finite difference and local refinement of global radial basis function method. This method uses the radial basis function to create weights for the RBF-FD formula, and the differential matrix obtained in this way is as sparse as that obtained by the standard difference method, which combines many advantages of radial basis function and traditional finite difference approximation. Thus, RBF-FD approximation [8] is an attractive substitute for the global radial basis function method.

In order to control the stagnation error, Natasha Flyer et al. introduced higher-order polynomials under the background of Gaussian (GA) and Polyharmonic Spline (PHS) interpolation [9,10]. PHS eliminates the stagnation error in combination with the polynomial to capture the basic physical properties of the problem, and achieves high-order accuracy without the need to adjust the shape parameters. The present paper chooses the PHS basis function and applies the radial basis function finite difference method with the polynomial to discretize the spatial operator of the equation. It is well known that the iterative method [11] is effective for Navier–Stokes equations with strong nonlinear terms, e.g., the Newton iterative method, Stokes iterative method, and Oseen iterative method. This paper focuses on the Oseen iterative method to deal with the nonlinear terms of the equation. The radial basis function finite difference method based on Oseen iteration proposed in this paper does not need a complete set of matrices in each nonlinear iteration, which is more suitable for nonlinear problems, and obtains high-precision numerical solutions.

2. Problem Setup

This paper studies the following two-dimensional steady incompressible Navier–Stokes equations [12].

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = f \quad &\text{on } \Omega, \\ \nabla\cdot\boldsymbol{u} = \boldsymbol{0} \quad &\text{on } \Omega, \\ \boldsymbol{u} = \boldsymbol{0} \quad &\text{on } \partial\Omega. \end{aligned} \tag{1}$$

where Ω is the bounded region on R^2, $\boldsymbol{u} = (u, v) : \Omega \to R^2$ is the velocity vector, $p : \Omega \to R$ is the pressure, $\nu > 0$ is the viscosity coefficient, and $f = (f_1, f_2)$ is the external force acting on a unit volume of fluid, assuming that the uniqueness of p can be achieved by imposing the condition, $\int_\Omega p\, dx = 0$.

We define the 2-norm and ∞-norm of the vector x as

$$\|x\|_2 = (\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}}, \qquad \|x\|_\infty = \underset{1\le i\le n}{max} |x_i|.$$

3. Numerical Method

In this part, we focus on the radial basis function finite difference method with polynomial.

3.1. Radial Basis Function Finite Difference Method with Polynomial

First, this part provides a brief introduction to the radial basis function (RBF). The RBF, expressed as $\Phi(r) : R^d \to R$, is a d-dimensional radially symmetric function, and is only dependent on $r = \|x - x_k\|_2$, where $\|\cdot\|_2$ represents the Euclidean distance between two points, $\boldsymbol{x}$ is the point to be solved, and $\boldsymbol{x}_k$ is the center position of the RBF. Common radial basis functions include Gauss (GA) ($e^{-(\varepsilon r)^2}$), multiquadratic function ($\sqrt{1 + (\varepsilon r)^2}$), and Polyharmonic Spline (PHS) ($r^m, m = 1, 3, 5, \cdots$), where ε is the shape parameter that determines the radial basis function. Next, we introduce the radial basis function finite difference method with polynomial. First, L is defined as a linear operator, which can be Δ, $\dfrac{\partial}{\partial x}$, $\dfrac{\partial}{\partial y}$, etc. We can approximate the value of the operator L at node $\boldsymbol{x}_c$ by linear combination of the function value $\{u_k\}_{k=1}^{n}$. For this, we need to select n nodes near the central node $\boldsymbol{x}_c$ to form a node template, which is recorded as $X = \{\boldsymbol{x}_k\}_{k=1}^{n}$, including $\boldsymbol{x}_c$, as follows:

$$(Lu)\mid_{x=x_c}=\sum_{k=1}^{n} w_k u_k, \tag{2}$$

where n is the size of the node template and w_k is the differential weight. In order to calculate the weight w_k, we can specify that the linear combination of function values be accurate for interpolation $s(x)$:

$$s(x)=\sum_{k=1}^{n}\lambda_k\Phi(\|x-x_k\|_2)+\sum_{k=1}^{\ell}\mu_k p_k(x), \tag{3}$$

The constraint condition is

$$\sum_{k=1}^{n}\lambda_k p_j(x_k)=0 \;\; j=1,2\cdots\ell, \tag{4}$$

where $p_l(x)$ is degree l of the binary polynomial and $\Phi(\|\boldsymbol{x}-\boldsymbol{x}_k\|_2)$ is the radial basis function.

When $d=2$ and the degree of the polynomial is $l=1$, Equation (3) is

$$s(x)=\sum_{k=1}^{n}\lambda_k\Phi(\|x-x_k\|_2)+\mu_1+\mu_2 x+\mu_3 y, \tag{5}$$

with the constraint condition (4)

$$\sum_{k=1}^{n}\lambda_k=\sum_{k=1}^{n}\lambda_k x_k=\sum_{k=1}^{n}\lambda_k y_k=0. \tag{6}$$

Suppose that the function value at x_k is $f_k, k=1,\cdots,n$; then, the matrix form of Equations (5) and (6) is

$$\begin{bmatrix}\Phi(\|x_1-x_1\|_2) & \cdots & \Phi(\|x_1-x_n\|_2) & 1 & x_1 & y_1\\ \Phi(\|x_2-x_1\|_2) & \cdots & \Phi(\|x_2-x_n\|_2) & 1 & x_2 & y_2\\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots\\ \Phi(\|x_n-x_1\|_2) & \cdots & \Phi(\|x_n-x_n\|_2) & 1 & x_n & y_n\\ 1 & \cdots & 1 & 0 & 0 & 0\\ x_1 & \cdots & x_n & 0 & 0 & 0\\ y_1 & \cdots & y_n & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_n\\ \mu_1\\ \mu_2\\ \mu_3\end{bmatrix}=\begin{bmatrix}f_1\\ f_2\\ \vdots\\ f_n\\ 0\\ 0\\ 0\end{bmatrix},$$

In this formula, $\hat{A}$ is used to represent the matrix at the left end $(n+3)\times(n+3)$; thus,

$$\begin{aligned}s(x)&=[\Phi(\|x-x_1\|_2)\ldots\Phi(\|x-x_n\|_2)\;1\;x\;y][\lambda_1\ldots\lambda_n\;\mu_1\;\mu_2\;\mu_3]^T\\ &=[\Phi(\|x-x_1\|_2)\ldots\Phi(\|x-x_n\|_2)\;1\;x\;y]\hat{A}^{-1}[f_1\ldots f_n\;0\;0\;0]^T,\end{aligned} \tag{7}$$

The linear operator L is used to calculate the result at $x=x_c$:

$$\begin{aligned}Ls(x)|_{x=x_c}=&[L\Phi(\|x-x_1\|_2)|_{x=x_c}\ldots L\Phi(\|x-x_n\|_2)|_{x=x_c}\\ &L1|_{x=x_c}\;Lx|_{x=x_c}\;Ly|_{x=x_c}]\hat{A}^{-1}[f_1\ldots f_n\;0\;0\;0]^T.\end{aligned} \tag{8}$$

If $f_i=1$ and $f_j=0$, then $j\neq i$; thus, $Ls(x)|_{x=x_c}=w_i$ and $i=1,\cdots,n$, according to Formula (2). Therefore,

$$[w_1,\cdots,w_n]=[L\Phi(\|x-x_1\|_2)|_{x=x_c}\ldots L\Phi(\|x-x_n\|_2)|_{x=x_c}$$

$$L1|_{x=x_c}\;Lx|_{x=x_c}\;Ly|_{x=x_c}]\hat{A}^{-1}\begin{bmatrix}D\\ O\end{bmatrix}_{(n+3)\times n},$$

where D is the identity matrix of $(n\times n)$ and O is the null matrix of $(3\times n)$. An identity matrix is formed by adding three columns to the right matrix, which can be ignored. Thus, the left-hand matrix is added with $[w_{n+1}, w_{n+2}, w_{n+3}]$. The right-hand matrix is multiplied by $\hat{A}$ and then transposed, obtaining the following linear equations:

$$\begin{bmatrix} \Phi(\|x_1 - x_1\|_2) & \cdots & \Phi(\|x_1 - x_n\|_2) & 1 & x_1 & y_1 \\ \Phi(\|x_2 - x_1\|_2) & \cdots & \Phi(\|x_2 - x_n\|_2) & 1 & x_2 & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \Phi(\|x_n - x_1\|_2) & \cdots & \Phi(\|x_n - x_n\|_2) & 1 & x_n & y_n \\ 1 & \cdots & 1 & 0 & 0 & 0 \\ x_1 & \cdots & x_n & 0 & 0 & 0 \\ y_1 & \cdots & y_n & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \\ w_{n+1} \\ w_{n+2} \\ w_{n+3} \end{bmatrix} = \begin{bmatrix} L\Phi(\|x - x_1\|_2)\ |_{x=x_c} \\ L\Phi(\|x - x_2\|_2)\ |_{x=x_c} \\ \vdots \\ L\Phi(\|x - x_n\|_2)\ |_{x=x_c} \\ L1\ |_{x=x_c} \\ Lx\ |_{x=x_c} \\ Ly\ |_{x=x_c} \end{bmatrix}.$$

Therefore, the weights $w_1 \cdots w_n$ are obtained from the above equations. In the following numerical examples, the resulting linear system changes with the varying degree of the polynomial. In this paper, the PHS radial basis function is mainly considered; $\Phi(r) = r^m, m = 1, 3, 5, \cdots$, PHS is a piecewise smooth function, where $r = \|\mathbf{x} - \mathbf{x}_k\|_2$, and it is unnecessary to select a shape parameter ε .

3.2. *Radial Basis Function Finite Difference Method Based on Oseen Iteration*

In this part, we introduce discretization of the Navier–Stokes equations using the radial basis function finite difference method based on Oseen iteration. First, we use the radial basis function finite difference method with polynomial to discretize the component equations. Then, we linearize the equations based on Oseen iteration starting from this scheme, and finally obtain the fully discrete scheme of the equations.

3.2.1. Discrete Scheme of Equation

First, the radial basis function finite difference method with polynomial is used to discretize the spatial operator of the equation. Equation (1) is expressed as the following components:

$$-\nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} = f_1, \tag{9}$$

$$-\nu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}) + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} = f_2, \tag{10}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0. \tag{11}$$

For Equation (9), the differential value at a point x_c is

$$Lu(x_c) = \sum_{j=1}^{n} w_j u(x_{cj}), \tag{12}$$

where $\{x_{cj}\}_j^n$ is the n points around x_c and $\{w_j\}_j^n$ is the weight coefficient corresponding to these points obtained by applying the interpolation condition $f(x_{cj}) = f_j$.

First, we assume that there are N nodes in a zone. At a certain point x_c, Formula (9) is discretized according to RBF-FD interpolation and the following form is obtained:

$$\begin{aligned} -\nu \sum_{j=1}^{n_1} w_j^{\Delta} u(x_{cj}) + u(x_c) \sum_{j=1}^{n_1} w_j^{\nabla_x} u(x_{cj}) + v(x_c) \sum_{j=1}^{n_1} w_j^{\nabla_y} u(x_{cj}) + \sum_{j=1}^{n_2} w_j^{\nabla_x} p(x_{cj}) \\ = f_1(x_c), \end{aligned} \tag{13}$$

where n_1 represents the number of nearest nodes at the central node $\boldsymbol{x_c}$, used for approximating the weight coefficients of Δu and ∇u, and n_2 represents the number of nodes at the central node $\boldsymbol{x_c}$, used to form a template of nearest nodes for approximating the weight coefficients of ∇p; moreover, w_j^{Δ} is a weight coefficient for approximating Δu, $w_j^{\nabla_x}$ is a weight coefficient for approximating operator $\frac{\partial}{\partial x}$, and $w_j^{\nabla_y}$ is a weight coefficient for approximating operator $\frac{\partial}{\partial y}$. Next, we use the Oseen iterative scheme to deal with the nonlinear terms of the steady Navier–Stokes equation. Iterating according to Discrete Equation (13), we obtain the following form:

$$\begin{aligned}-\nu\sum_{j=1}^{n_1} w_j^{\Delta} u(x_{cj}^{k+1}) + u(x_c^k)\sum_{j=1}^{n_1} w_j^{\nabla_x} u(x_{cj}^{k+1}) + v(x_c^k)\sum_{j=1}^{n_1} w_j^{\nabla_y} u(x_{cj}^{k+1}) + \sum_{j=1}^{n_2} w_j^{\nabla_x} p(x_{cj}^{k+1})& \\ = f_1(x_c^{k+1}),&\end{aligned} \tag{14}$$

then

$$\left(-\nu\sum_{j=1}^{n_1} w_j^{\Delta} + u(x_c^k)\sum_{j=1}^{n_1} w_j^{\nabla_x} + v(x_c^k)\sum_{j=1}^{n_1} w_j^{\nabla_y}\right) u(x_{cj}^{k+1}) + \sum_{j=1}^{n_2} w_j^{\nabla_x} p(x_{cj}^{k+1}) = f_1(x_c^{k+1}), \tag{15}$$

where k denotes the number of iterations, $u(x_c^k)$ denotes the value of u at the kth iteration, $u(x_c^{k+1})$ denotes the value of u at the (k + 1)th iteration, and the initial value of the velocity u^0 is obtained from the corresponding Stokes equation. In Equation (15), the value of u at the kth level is known; thus, the nonlinear term in Equation (9) is linearized. Likewise, there are similar discrete forms for Equations (10) and (11):

$$\left(-\nu\sum_{j=1}^{n_1} w_j^{\Delta} + u(x_c^k)\sum_{j=1}^{n_1} w_j^{\nabla_x} + v(x_c^k)\sum_{j=1}^{n_1} w_j^{\nabla_y}\right) v(x_{cj}^{k+1}) + \sum_{j=1}^{n_2} w_j^{\nabla_y} p(x_{cj}^{k+1}) = f_2(x_c^{k+1}), \tag{16}$$

$$\sum_{j=1}^{n_1} w_j^{\nabla_x} u(x_{cj}^{k+1}) + \sum_{j=1}^{n_1} w_j^{\nabla_y} v(x_{cj}^{k+1}) = 0, \tag{17}$$

There are N nodes in a zone including N_B boundary nodes and N_I internal nodes, where $N = N_I + N_B$. The central node x_c in Formulas (15) and (16) is considered on N_I internal nodes, and the central node x_c in Formula (17) is considered on N total nodes. Considering the boundary condition $\boldsymbol{u} = 0$, Equations (15)–(17) and the boundary conditions form the following equations:

$$\begin{bmatrix} W^{\Delta+\nabla}_{N_I\times N_I} & W^{\Delta+\nabla}_{N_I\times N_B} & O_{N_I\times N_I} & O_{N_I\times N_B} & W^{\nabla_x}_{N_I\times N} \\ O_{N_B\times N_I} & E_{N_B\times N_B} & O_{N_B\times N_I} & O_{N_B\times N_B} & O_{N_B\times N} \\ O_{N_I\times N_I} & O_{N_I\times N_B} & W^{\Delta+\nabla}_{N_I\times N_I} & W^{\Delta+\nabla}_{N_I\times N_B} & W^{\nabla_y}_{N_I\times N} \\ O_{N_B\times N_I} & O_{N_B\times N_B} & O_{N_B\times N_I} & E_{N_B\times N_B} & O_{N_B\times N} \\ W^{\nabla_x}_{N\times N_I} & W^{\nabla_x}_{N\times N_B} & W^{\nabla_y}_{N\times N_I} & W^{\nabla_y}_{N\times N_B} & O_{N\times N} \end{bmatrix} \begin{bmatrix} U_I \\ U_B \\ V_I \\ V_B \\ P \end{bmatrix} = \begin{bmatrix} F_1 \\ 0 \\ F_2 \\ 0 \\ 0 \end{bmatrix}, \tag{18}$$

We can now write (18) in the form of a block matrix:

$$\begin{bmatrix} W_u^{\Delta+\nabla} & O & W_p^{\nabla_x} \\ O & W_v^{\Delta+\nabla} & W_p^{\nabla_y} \\ W_u^{\nabla_x} & W_v^{\nabla_y} & O \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ 0 \end{bmatrix}. \tag{19}$$

The stabilization parameter ϵ is added to Equation (19) to obtain a system of equations in the form of BU = F:

$$\begin{bmatrix} W_u^{\Delta+\nabla} & O & W_p^{\nabla_x} \\ O & W_v^{\Delta+\nabla} & W_p^{\nabla_y} \\ W_u^{\nabla_x} & W_v^{\nabla_y} & \epsilon D \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ 0 \end{bmatrix}. \tag{20}$$

where B is the left-end weight matrix, $W_u^{\Delta+\nabla}$ and $W_v^{\Delta+\nabla}$ are $(N \times N)$ matrices composed of weight coefficients and boundary condition coefficients used to approximate u and v in Equations (15) and (16), respectively, $W_u^{\nabla_x}$ and $W_v^{\nabla_y}$ are a matrix of $(N \times N)$ composed of weighted coefficients for approximating $\frac{\partial u}{\partial x}$ and $\frac{\partial v}{\partial y}$, $W_p^{\nabla_x}$ and $W_p^{\nabla_y}$ are a matrix of $(N \times N)$ composed of weighted coefficients for approximating $\frac{\partial p}{\partial x}$ and $\frac{\partial p}{\partial y}$, F_1 and F_2 are known right-end terms, $\mathbf{U}$ is the solution of velocity u on point set X, $\mathbf{V}$ is the solution of velocity v on point set X, $\mathbf{P}$ is the solution of pressure p on point set X, D is the unit matrix of $(N \times N)$, and ϵ is the stabilization parameter, $\epsilon = 1.0 \times 10^{-6}$.

3.2.2. Selection of Degrees of Added Polynomial and Number of Template Nodes

According to [13,14], without depending on the dimension, when using radial basis function PHS: $\Phi(r) = r^p$, when p is an odd number and the l-degree polynomials are supplemented to approximate the k-order derivative the convergence rate is not determined by the order of PHS, and is instead determined by the highest degree of the polynomial used. The solution accuracy is $O(h^{l-k+1})$. For the PDE with only the first-order spatial derivative, the convergence speed can reach $O(h^l)$, while for the PDE with the second-order spatial derivative, the convergence speed can reach $O(h^{l-1})$. The number of terms of the l-degree polynomial is $\frac{(l+1)(l+2)}{2}$. When approximating the value of an operator acting on the center point of the template, symmetry of the node templates is beneficial, as symmetrical node templates provide uniform information. In this paper, the numerical examples are considered in two layouts, namely, a right-angle node layout and a hexagonal node layout, with the node layouts shown in Figure 1. The right-angled node layout has completely symmetrical templates on numbers 5, 9, 13, 21, 25, 29, 37, 45, etc., while the hexagonal node layout has completely symmetrical templates on numbers 7, 13, 19, 31, 37, 43, 55, 61, etc. In this paper, the minimum distance between two points under the right-angled node layout is $h = \frac{1}{\sqrt{N}-1}$, while N is the total number of nodes.

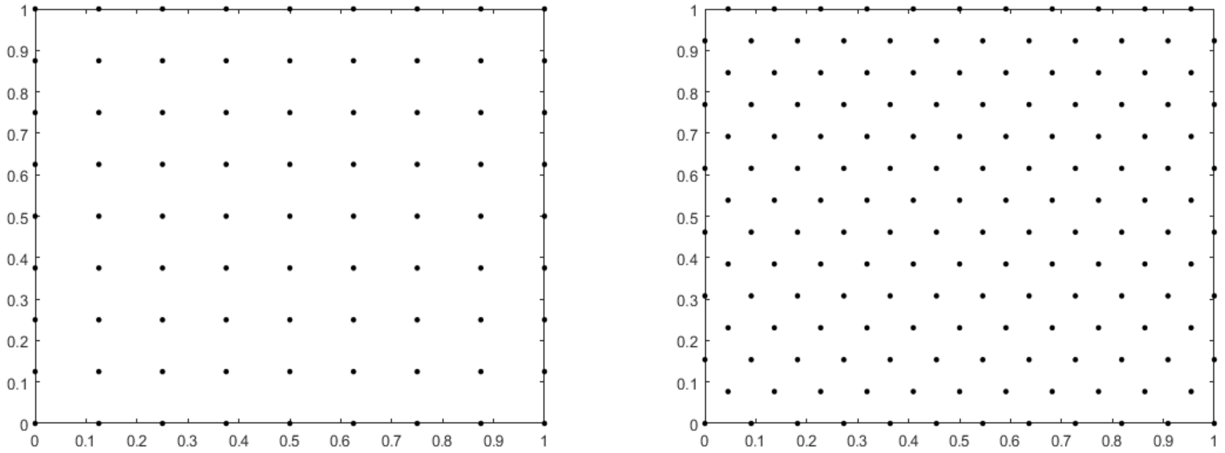

Figure 1. Right-angled node layout (**left**) and hexagonal node layout (**right**).

4. Numerical Method

In this section, several numerical examples are used to demonstrate the effectiveness of the proposed methods; the relative error L_2 and error L_∞ are applied for comparison of the numerical solution and true solution. The relative error L_2 is defined as $error_{L_2} = \frac{\|u-u_h\|_2}{\|u\|_2}$, while the error L_∞ is defined as $error_{L_\infty} = \|u - u_h\|_{L_\infty(\Omega)} = \max\limits_{\mathbf{x}\in X} |u(\mathbf{x}) - u_h(\mathbf{x})|$, where u_h is the obtained numerical solution and u is the true solution $x \in X$.

4.1. Convergence Test

This example studies the two-dimensional Navier–Stokes equations under the layout of right-angled nodes. We assume that Equation (1) has the true solutions

$$\begin{aligned} u &= \left(x^2 - 2x^3 + x^4\right)\left(2y - 6y^2 + 4y^3\right) \quad \text{on } \Omega, \\ v &= -\left(y^2 - 2y^3 + y^4\right)\left(2x - 6x^2 + 4x^3\right) \quad \text{on } \Omega, \\ p &= \cos(\pi x)\sin(\pi y) \quad \text{on } \Omega. \end{aligned} \tag{21}$$

where $\Omega = [0,1] \times [0,1]$, the viscosity coefficient $\nu = 1$, and the nodes are uniformly distributed on Ω as shown in Figure 1 left. For purpose of the convergence test, the basis function $\Phi(r) = r^7$ is applied to add third-order and second-order polynomials for approximation of the velocity $\mathbf{u} = (u, v)$, while the basis function $\Phi(r) = r^5$ is applied to add second-order and first-order polynomials for approximation of the pressure p. For the number of node templates approximated at a central node of the velocity and pressure in Equations (15) and (16), we select $n_1 = 21$ and $n_2 = 9$, respectively.

Figure 2 left shows the addition of the third-order polynomial for approximation of the velocity and the error results for pressure approximation when adding the second-order polynomial. The figure shows the relative error L_2 when the number of nodes N is 25, 81, 289, 1089, and 4225. The error decreases with increasing number of nodes, and the convergence order of velocity and pressure reaches third-order and second-order. When the number of nodes N is 4225, the velocity error is 4.4583×10^{-5} and the pressure error is 2.2254×10^{-2}. The right0hand side of Figure 2 shows the error results of the approximation when adding the second-order polynomial to the velocity and adding the first-order polynomial to the pressure. When the number of nodes N is 4225, the relative errors L_2 of the velocity and pressure are 4.2393×10^{-5} and 2.8000×10^{-2}, respectively.

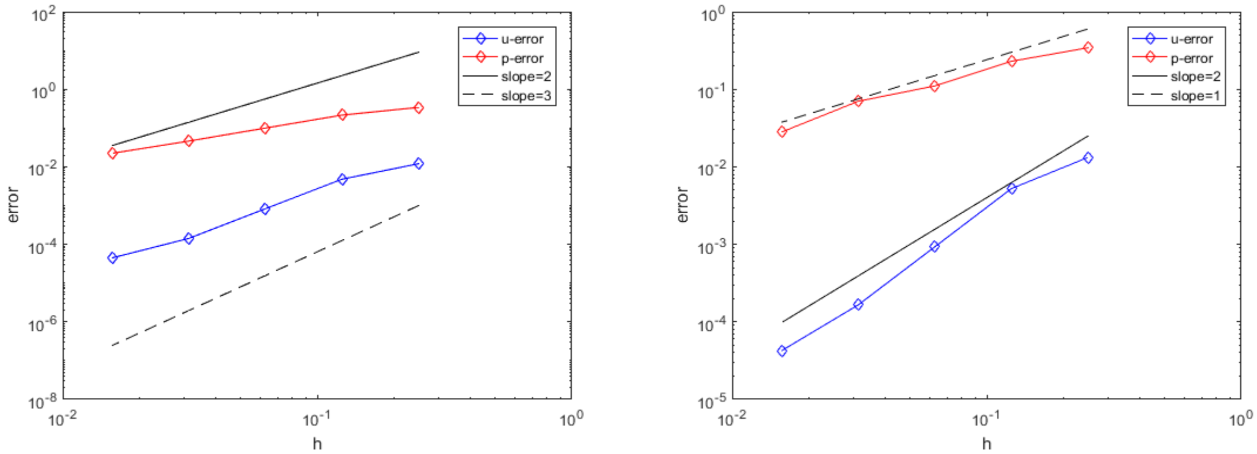

Figure 2. Relative error L_2 of the velocity u and pressure p (right-angled node layout).

By comparing the two figures, the error between the numerical solution and the true solution obtained from approximation by adding third-order and second-order polynomials to the velocity and pressure is far less than that obtained from approximation by adding second-order and first-order polynomials to the velocity and pressure. When the number of nodes is less than 1089, the former reaches the corresponding convergence order. As the number of nodes approaches 4225, the order of velocity is less than 2 due to the reduction in point density. This example illustrates that under the right-angled node layout, the radial basis function difference method based on Oseen iteration is convergent and effective for solving incompressible Navier–Stokes equations.

4.2. Small Viscosity Problem

In order to test the suitability of the method proposed in this paper for small viscosity problems, we begin by assuming that Equation (1) has a true solution (21). By setting the viscosity coefficient $\nu = 0.00001$, the nodes are distributed in a right-angled node layout. In this example, the basis function $\Phi(r) = r^7$ is applied to add the third-order polynomial for approximation of the velocity $u = (u, v)$, and the basis function $\Phi(r) = r^5$ is applied to add the second-order polynomial for approximation of the pressure p. Table 1 shows the error results when ν is 0.00001 and number of nodes N is 25, 81, 289, 1089, and 4225. When the number of nodes N is 4225, the relative error L_2 of the velocity is 6.7100×10^{-5} and the relative error L_2 of the pressure is 2.0271×10^{-2}. Figure 3 shows the corresponding error order.

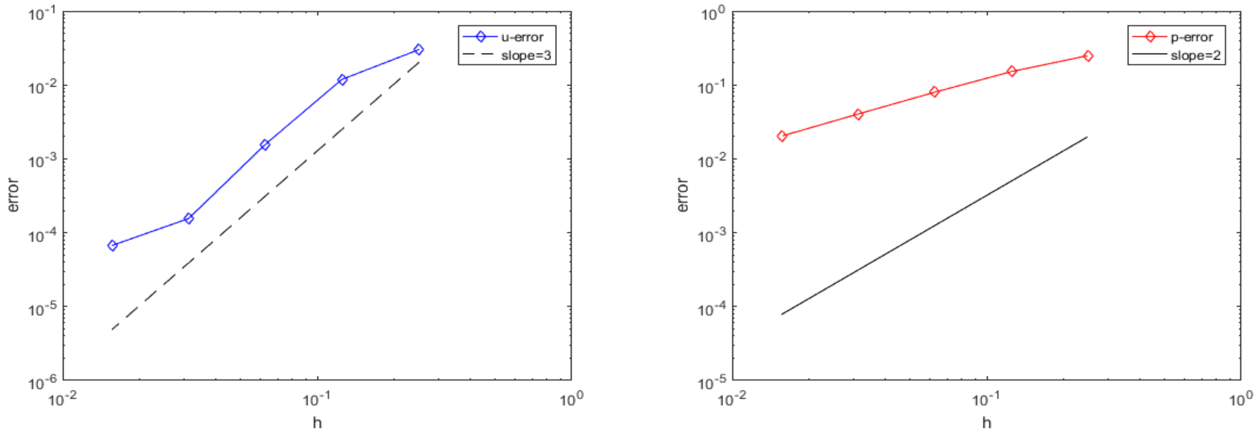

Figure 3. Relative error L_2 of the velocity u (**left**) and the pressure p (**right**) ($\nu = 0.00001$).

Table 1. Relative error L_2 and error L_∞ between numerical solution and true solution of the velocity u and pressure p ($\nu = 0.00001$).

N	Relative Error L_2 of u	Relative Error L_2 of p	L_∞ Error of u	L_∞ Error of p
25	2.9942×10^{-2}	2.4962×10^{-1}	9.9977×10^{-1}	1.0191×10^{-1}
81	1.1883×10^{-2}	1.5148×10^{-1}	1.9150×10^{-1}	3.3872×10^{-2}
289	1.5680×10^{-3}	7.9769×10^{-2}	1.2605×10^{-2}	9.4008×10^{-3}
1089	1.5564×10^{-4}	4.0410×10^{-2}	6.2549×10^{-4}	2.4502×10^{-3}
4225	6.7100×10^{-5}	2.0271×10^{-2}	1.3483×10^{-4}	6.2380×10^{-4}

4.3. Hexagonal Node Layout

In this section, in order to test the suitability of the method proposed in this paper for different node layouts, we consider the two-dimensional equation under the hexagonal node layout, as shown in Figure 1 right, assuming that Equation (1) has a true solution (21) where $\Omega = [0, 1] \times [0, 1]$ and the viscosity coefficient $\nu = 1$. In this example, the basis function $\Phi(r) = r^7$ is applied to add the third-order polynomial for approximation of the velocity u and the basis function $\Phi(r) = r^5$ is applied to add the second-order polynomial for approximation of the pressure p. For the velocity and pressure in a discrete equation, the number of node templates approximated at a central node (n_1 and n_2) are separately selected as $n_1 = 31$ and $n_2 = 13$. When number of nodes is 4699, the relative error L_2 of the velocity is 2.8638×10^{-5} and the relative error L_2 of the pressure is 1.3900×10^{-2}. Figure 4 shows the relative error L_2 between the numerical solution and the true solution when the number of nodes N is 68, 279, 1166, and 4699.

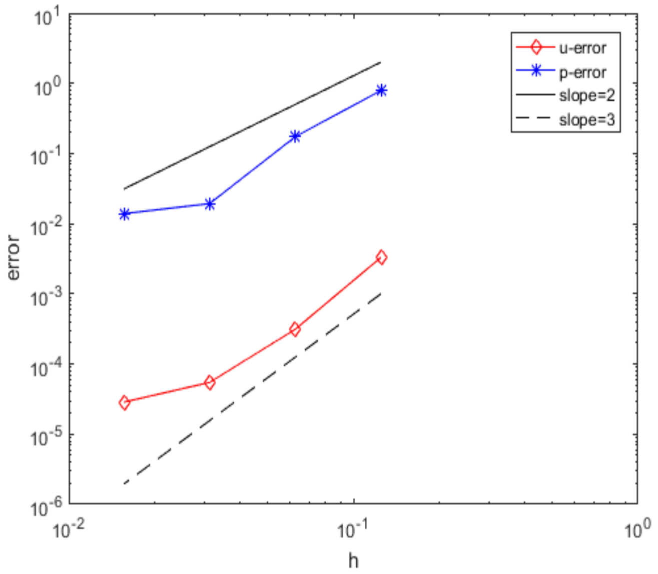

Figure 4. Relative error L_2 of the velocity u and pressure p (hexagonal node layout).

In summation, these numerical experiments show that the radial basis function finite difference method based on Oseen iteration has good performance for solving Navier–Strokes equations under the hexagonal node layout.

5. Conclusions

This paper proposes a radial basis function finite difference method based on Oseen iteration for solving two-dimensional steady Navier–Stokes equations with discretization of the spatial operators of the Navier–Stokes equations using the radial basis function finite difference method with polynomial, then linearizing the equation based on Oseen iteration. In this paper, we provide numerical solutions for the Navier–Stokes equations under right-angled and hexagonal node layouts, analyze and compare the influence of the number of polynomial additions on the accuracy and convergence of the solution, and verify the effectiveness of the proposed method. Under the two node layouts studied in this paper, the proposed method offers high-precision numerical solutions for solving Navier–Stokes equations while demonstrating good performance. Its extension to unsteady problems will be the focus of our future research work.

Author Contributions: Conceptualization, X.F.; methodology, L.M.; software, L.M.; validation, X.F.; formal analysis, L.M.; investigation, L.M.; resources, X.F.; data curation, L.M.; writing—original draft preparation, L.M.; writing—review and editing, X.F. and L.M.; visualization, L.M.; supervision, X.F.; project administration, X.F.; funding acquisition, X.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Teman, R. *Navier-Stokes Equations: Theory and Numerical Analysis*; Springer: Amsterdam, The Netherland; New York, NY, USA; Oxford, UK, 1984.
2. He, Y.; Miao, H.; Mattheij, R. Numerical analysis of a modified finite element nonlinear Galerkin method. *Numer. Math.* **2004**, *97*, 725–756. [CrossRef]

3. He, Y.; Wang, A.; Mei, L. Stabilized finite-element method for the stationary Navier-Stokes equations. *J. Eng. Math.* **2005**, *51*, 367–380. [CrossRef]
4. He, Y.; Feng, X. Research progress of efficient fully discrete finite element method for 3D unsteady Navier-Stokes equations. *J. Xinjiang Univ.* **2022**, *39*, 257–265.
5. Raissi, M.; Perdikaris, P.; Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
6. Bayona, V.; Moscoso, M.; Carretero, M. RBF-FD formulas and convergence properties. *J. Comput. Phys.* **2010**, *229*, 8281–8295. [CrossRef]
7. Bayona, V.; Moscoso, M.; Kindelan, M. Optimal constant shape parameter for multiquadric based RBF-FD method. *J. Comput. Phys.* **2011**, *230*, 7384–7399. [CrossRef]
8. Bayona, V.; Kindelan, M. Propagation of premixed laminar flames in 3D narrow open ducts using RBF-generated finite differences. *Combust. Theory Model.* **2013**, *17*, 789–803. [CrossRef]
9. Bayona, V.; Flyer, N.; Fornberg, B. On the role of polynomials in RBF-FD approximations, II: Numerical solution of elliptic PDEs. *J. Comput. Phys.* **2017**, *332*, 257–273. [CrossRef]
10. Bayona, V.; Flyer, N.; Fornberg, B. On the role of polynomials in RBF-FD approximations, III: Behavior near domain boundaries. *J. Comput. Phys.* **2019**, *38* , 378–399. [CrossRef]
11. He, Y. Stability and convergence of iterative methods related to viscosities for the 2D/3D steady Navier-Stokes equations. *J. Math. Anal. Appl.* **2015**, *423*, 1129–1149. [CrossRef]
12. He, Y.; Li, J. Convergence of three iterative methods based on the finite element discretization for the stationary Navier-Stokes equations. *Comput. Methods Appl. Mech. Eng.* **2009**, *198*, 1351–1359. [CrossRef]
13. Flyer, N.; Fornberg, B.; Bayona, V. On the role of polynomials in RBF-FD approximations, I: Interpolation and accuracy. *J. Comput. Phys.* **2016**, *321*, 21–38. [CrossRef]
14. Flyer, N.; Barnett, G.; Wicker, L. Enhancing finite differences with radial basis functions: Experiments on the Navier-Stokes equations. *J. Comput. Phys.* **2016**, *316*, 39–62. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.