J. Appl. Comput. Mech., 4(3) (2018) 231-244
DOI: 10.22055/JACM.2017.23557.1163

ISSN: 2383-4536
jacm.scu.ac.ir

Shahid Chamran University of Ahvaz

Journal of Applied and Computational Mechanics

Research Article

Verification and Validation of Common Derivative Terms Approximation in Meshfree Numerical Scheme

Zhibo Ma[1], Yazhou Zhao[2,3,4,5]

[1] Institute of Applied Physics and Computational Mathematics
Beijing, 100094, China, mazhibo@iapcm.ac.cn

[2] Key Laboratory of Shale Gas and Geoengineering, Institute of Geology and Geophysics, Chinese Academy of Sciences
Beijing, 100029, China, asiabuaasa@163.com

[3] Institutions of Earth Science, Chinese Academy of Sciences, Beijing, 100029, China

[4] University of Chinese Academy of Sciences, Beijing, 100049, China

[5] Institute of Geothermal Energy, Zhejiang Loopmaster Energy Technology Corp., Ltd
Hangzhou, 310052, China

Received October 02 2017; Revised November 15 2017; Accepted for publication November 28 2017.
Corresponding author: Yazhou Zhao, asiabuaasa@163.com

Abstract. In order to improve the approximation of spatial derivatives without meshes, a set of meshfree numerical schemes for derivative terms is developed, which is compatible with the coordinates of Cartesian, cylindrical, and spherical. Based on the comparisons between numerical and theoretical solutions, errors and convergences are assessed by a posteriori method, which shows that the approximations for functions and derivatives are of the second accuracy order, and the scale of the support domain has some influences on numerical errors but not on accuracy orders. With a discrete scale h=0.01, the relative errors of the numerical simulation for the selected functions and their derivatives are within 0.65%.

Keywords: Meshfree method; Smoothed particle hydrodynamics; Physics evoked cloud method; Approximation of spatial derivative; Verification and validation.

1. Introduction

Approximate calculation of the function and its spatial derivative is the key to solving the partial differential equation for meshfree numerical method [1-3]. This technique is mainly concerned with the compatibility, stability, and convergence of the numerical schemes, and its credibility can be evaluated in conjunction with code verification process. The main purpose of code verification is to verify the correctness of the numerical algorithm and its corresponding code. Since the mathematical model involved in the numerical calculation is usually more complex, *a priori* analysis method based on mathematical theory would encounter great difficulties; therefore, code validation of the numerical simulation is often completed by means of *a posteriori* analysis method with resort to the contrast information of numerical solution and high confidence reference solution [4-5]. As for the existing meshfree algorithm based on kernel estimation, the derivatives involved are mostly limited to Cartesian coordinates or relatively simple derivative forms. The posterior evaluation is also based on qualitative or semi-quantitative comparison analysis [6 -17].

With the continually expanding application field of meshfree method, the entity model concerned may have different symmetry characteristics, so it requires numerical calculation to be able to handle different coordinate system types. Calculation ability of first order spatial derivatives such as gradient, divergence, and rotation, and second order spatial

Published Online: June 14 2018

derivatives such as Laplacian and dissipative terms for the meshfree numerical schemes is necessary in the numerical simulations of complex processes involving fluid, solid, and heat transfer. In terms of reliability assessment of numerical simulation, the continuous development of verification and validation (V&V) technique has also provided more and more technical support for the comprehensive evaluation based on qualitative and quantitative analysis. Among them, the quantitative evaluation of the convergence accuracy level has become an important technical content of convergence analysis [18-20].

There are three aspects in the engineering application of the meshfree numerical calculation of the function and its derivative: First, compatibility, convergency, and has the required precision degree; Second, has the ability to adapt to different coordinate systems; and Third, simple, efficient, and easy to program. The SPH (Smoothed Particle Hydrodynamics) method, proposed by Gingold and Monaghan, constructs the numerical approximation scheme based on the kernel estimation idea, its structure is simple and independent of the form of the approximated function. Also, it is helpful to quickly solve the differential equations containing multiple unknowns [1]. Liu proposed the RKPM (Reproducing kernel particle method) method to correct the SPH kernel function so as to improve the kernel estimation accuracy of the boundary or interior point based on the volume and position information of the neighboring particles in the supporting domain [8]. Using the first order derivative of the kernel function to approximate the unknown first order derivative, it is generally possible to give a better approximation accuracy. But, if the second order derivative of the kernel function is used directly to estimate the unknown second order derivative, whether using SPH or RKPM kernel function, the approximation accuracy is significantly decreased [16].

For the second order derivative of arbitrary function, inspired by the finite difference idea, Brookshaw proposed an approximation method with comparatively high accuracy based on the first derivative of kernel function and gave the kernel estimation algorithm of the Laplacian operator in Cartesian coordinates [21]. On the basis of similar ideas, Cleary, Zhang, and Basa gave the kernel estimation algorithm for the second order derivatives in the Cartesian coordinates framework including dissipation term in the simulation of heat transfer and viscous flow [22-24]. In order to extend the meshfree method to simulate complex physical processes involving multiple types of materials, using the numerical approximation method of kernel estimation, the author Ma proposes a meshfree method called PECM (Physics Evoked Cloud Method) and illustrates the designing ideas and numerical schemes of this method in the paper [25]. Among them, a meshfree algorithm which is simple, efficient, and adaptable to the Cartesian, cylindrical, and spherical coordinates for the common first and second order derivative form is presented. In addition, based on these new algorithms, a meshfree numerical simulation software HAUC (How Are Universes Cuddling) applicable to complex multi-physics process is developed [25].

In this paper, first of all, the calculation scheme of derivatives in HAUC software is deduced theoretically, then, the compatibility, convergence, and the accuracy grade of convergence of the scheme are evaluated by posterior analysis. Based on the arbitrary given function type, the convergence of the numerical scheme is evaluated qualitatively and semi-quantitatively by the consistency of the numerical and the exact solution and the observation of the error norm decreasing with the discretized scale according to power law. Moreover, on the basis of the error norm, the quantitative evaluation of the accuracy level of the numerical scheme is performed followed by the evaluation results of the code verification given in the conclusion.

2. Meshfree approximation of function and its spatial derivative

2.1 Meshfree numerical scheme

The meshfree method estimates the functions and their derivatives at a certain spatial position by the function information of discrete elements in the support domain. In the field of computational mathematics and applied physics, the commonly used derivatives are generally not higher than the second order, and their meshfree approximate estimation can be expressed as:

Function itself :
$$f\left(\boldsymbol{\xi}\right)\Big|_{\boldsymbol{\xi}={}^{i}\boldsymbol{\xi}} \approx \sum_{j=1}^{N} {}^{j}\psi f\left({}^{j}\boldsymbol{\xi}\right) \tag{1}$$

First order derivative :
$$\frac{\partial f\left(\boldsymbol{\xi}\right)}{\partial \xi_{\alpha}}\Big|_{\boldsymbol{\xi}={}^{i}\boldsymbol{\xi}} \approx \sum_{j=1}^{N} {}^{j}\psi_{\alpha} f\left({}^{j}\boldsymbol{\xi}\right) \tag{2}$$

Second order derivative :
$$\frac{\partial^2 f\left(\boldsymbol{\xi}\right)}{\partial \xi_{\alpha}\partial \xi_{\beta}}\Big|_{\boldsymbol{\xi}={}^{i}\boldsymbol{\xi}} \approx \sum_{j=1}^{N} {}^{j}\psi_{\alpha\beta} f\left({}^{j}\boldsymbol{\xi}\right) \tag{3}$$

The above three formulas also apply to the function of the vector form $f\left({}^{j}\boldsymbol{\xi}\right)$, where the input information on the right hand of the terms are the function values carried by the material points scattered in the space. Here we note that for the discrete element *i*, there are *N* neighbor particles in the support domain, *j* is the label number for them, vector $\boldsymbol{\xi}$ is the coordinate of the discrete element, and α and β are the components representing the coordinate. For the Cartesian coordinate system 1, the three coordinate components ξ_1, ξ_2 and ξ_3 are *x*, *y*, and *z*, respectively. While the three corresponding coordinate components for the cylindrical or spherical coordinate system are *r*, *z* and θ, $(0 \le r < \infty, -\infty < z < \infty, 0 \le \theta < 2\pi)$ and r, θ and φ $(0 \le r < \infty,\ 0 \le \theta < 2\pi,\ 0 \le \phi \le \pi)$, respectively.

According to the technical specifications of V&V, the purpose of code validation is to test whether the meshfree method chooses the correct ${}^{j}\psi$, ${}^{j}\psi_{\alpha}$, and ${}^{j}\psi_{\alpha\beta}$ and correctly represents the discrete expression in the code to ensure that the approximate calculation of the function and derivatives satisfies the expected demand. The original function and derivative

types involved in this paper are scalar functions $f(\boldsymbol{\xi})$, $\lambda(\boldsymbol{\xi})$, and $\phi(\boldsymbol{\xi})$, respectively; and the vector functions $\boldsymbol{V}(\boldsymbol{\xi})$ whose three coordinate components are $V_1(\boldsymbol{\xi})$, $V_2(\boldsymbol{\xi})$, and $V_3(\boldsymbol{\xi})$, gradient function $\nabla f(\boldsymbol{\xi})$, rotation function $\nabla\times\boldsymbol{V}(\boldsymbol{\xi})$, divergence function $\nabla\cdot\boldsymbol{V}(\boldsymbol{\xi})$, Laplacian operator $\Delta f(\boldsymbol{\xi})$, and dissipation term $\nabla\cdot[\lambda(\boldsymbol{\xi})\nabla\phi(\boldsymbol{\xi})]$. For the above functions and derivative forms, Ma in HAUC proposes the following calculation schemes as:

$$\left\langle f\left({}^{i}\boldsymbol{\xi}\right)\right\rangle=\sum_{j=1}^{N}b_j f\left({}^{j}\boldsymbol{\xi}\right)\cdot W_{ij}^{(\mathrm{RKPM})} \tag{4}$$

$$\left\langle\nabla f\left({}^{i}\boldsymbol{\xi}\right)\right\rangle=\sum_{j=1}^{N}b_j f\left({}^{j}\boldsymbol{\xi}\right)\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})} \tag{5}$$

$$\left\langle\nabla\times\mathbf{V}\left({}^{i}\boldsymbol{\xi}\right)\right\rangle=\sum_{j=1}^{N}b_j\left[\mathbf{V}\left({}^{i}\boldsymbol{\xi}\right)-\mathbf{V}\left({}^{j}\boldsymbol{\xi}\right)\right]\times\nabla_i W_{ij}^{(\mathrm{RKPM})} \tag{6}$$

$$\left\langle\nabla\cdot\mathbf{V}\left({}^{i}\boldsymbol{\xi}\right)\right\rangle=\sum_{j=1}^{N}b_j\left[{}^{2}\boldsymbol{\Omega}_{ij}:\mathbf{V}\left({}^{j}\boldsymbol{\xi}\right)-\mathbf{V}\left({}^{i}\boldsymbol{\xi}\right)\right]\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})} \tag{7}$$

$$\left\langle\Delta f\left({}^{i}\boldsymbol{\xi}\right)\right\rangle=\sum_{j=1}^{N}2b_j\left[f\left({}^{i}\boldsymbol{\xi}\right)-f\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\left({}^{1}\boldsymbol{\Omega}_{ij}:\mathbf{r}_{ij}\right)\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]/\mathbf{r}_{ij}^2 \tag{8}$$

$$\left\langle\nabla\cdot\left[\lambda\left({}^{i}\boldsymbol{\xi}\right)\nabla\varphi\left({}^{i}\boldsymbol{\xi}\right)\right]\right\rangle=\sum_{j=1}^{N}b_j\left[\lambda\left({}^{i}\boldsymbol{\xi}\right)+\lambda\left({}^{j}\boldsymbol{\xi}\right)\right]\cdot\left[\varphi\left({}^{i}\boldsymbol{\xi}\right)-\varphi\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\left({}^{1}\boldsymbol{\Omega}_{ij}:\mathbf{r}_{ij}\right)\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]/\mathbf{r}_{ij}^2 \tag{9}$$

where " **:** " is a specific vector operator in the formulas (7-9); for example, if $\boldsymbol{a}=[a_1,a_2,a_3]$ and $\boldsymbol{b}=[b_1,b_2,b_3]$, then:

$$\boldsymbol{a}:\boldsymbol{b}=\left[a_1b_1,\ \ a_2b_2,\ \ a_3b_3\right] \tag{10}$$

Each variable is defined the same as in [25], where $\boldsymbol{r}_{ij}=\boldsymbol{r}_i-\boldsymbol{r}_j$ is the vector difference between the two discrete elements *i* and *j*. The value of the vector ${}^{m}\boldsymbol{\Omega}_{ij}\ (m=1,\ 2)$ is defined as:

$${}^{m}\boldsymbol{\Omega}_{ij}=\begin{cases}\left[\left(\dfrac{{}^{j}\xi_1}{{}^{i}\xi_1}\right)^{0.0m},\ 1.0,\ 1.0\right] & \text{Cartesian system}\\ \left[\left(\dfrac{{}^{j}\xi_1}{{}^{i}\xi_1}\right)^{0.5m},\ 1.0,\ 0.0\right] & \text{cylindrical system}\\ \left[\left(\dfrac{{}^{j}\xi_1}{{}^{i}\xi_1}\right)^{1.0m},\ 0.0,\ 0.0\right] & \text{spherical system}\end{cases} \tag{11}$$

Formula (11) applies to one-dimensional, two-dimensional, and three-dimensional problems in Cartesian coordinate systems, one-dimensional and two-dimensional problems under cylindrical coordinates, and one-dimensional problems under spherical coordinates. Under cylindrical or spherical coordinates, ξ_1 corresponds to the radial coordinates and satisfies $0\le\xi_1<\infty$.

2.2 Derivation of the numerical scheme

2.2.1 Theoretical basis

Derivation of the first order derivative approximation of the gradient, divergence, and rotation in the SPH formulation has been completed in the early literature. The basic idea is to transform the derivative of the objective function into the derivative of the known kernel function by integral by part and Gaussian formula, combined with the compactness of the kernel function in the support domain. When the RKPM kernel function is used, the corresponding divergence numerical scheme is:

$$\left\langle\nabla\cdot\mathbf{V}\left({}^{i}\boldsymbol{\xi}\right)\right\rangle=\sum_{j=1}^{N}b_j\left[\mathbf{V}\left({}^{j}\boldsymbol{\xi}\right)-\mathbf{V}\left({}^{i}\boldsymbol{\xi}\right)\right]\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})} \tag{12}$$

For operators involving second order derivative, the more accurate Brookshaw scheme can be expressed as:

$$\left\langle\Delta f\left({}^{i}\boldsymbol{\xi}\right)\right\rangle=\sum_{j=1}^{N}2b_j\left[f\left({}^{i}\boldsymbol{\xi}\right)-f\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]/\mathbf{r}_{ij}^2 \tag{13}$$

$$\left\langle\nabla\cdot\left[\lambda\left({}^{i}\boldsymbol{\xi}\right)\nabla\varphi\left({}^{i}\boldsymbol{\xi}\right)\right]\right\rangle=\sum_{j=1}^{N}b_j\left[\lambda\left({}^{i}\boldsymbol{\xi}\right)+\lambda\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\varphi\left({}^{i}\boldsymbol{\xi}\right)-\varphi\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]/\mathbf{r}_{ij}^2 \tag{14}$$

Formulations (4-6) do not change with the type of coordinate system; the derivation can be seen in the early literature [1, 8]. Equations (7-9) involve the calculation of divergence, Laplacian operator, and dissipation term. The numerical scheme is related to the type of the selected orthogonal coordinate system. If only the direction of the cylindrical coordinates ξ_1 is considered, we have:

$$\nabla\cdot\mathbf{V}=\frac{\partial\left(\xi_1 v_{\xi_1}\right)}{\xi_1\partial\xi_1} \tag{15}$$

$$\Delta f = \frac{1}{\xi_1}\frac{\partial}{\partial \xi_1}\left(\xi_1 \frac{\partial f}{\partial \xi_1}\right) \tag{16}$$

$$\nabla \cdot (\lambda \nabla \phi) = \frac{1}{\xi_1}\frac{\partial}{\partial \xi_1}\left(\xi_1 \lambda \frac{\partial \phi}{\partial \xi_1}\right) \tag{17}$$

If only the direction of the spherical coordinates ξ_1 is considered, we have:

$$\nabla \cdot \mathbf{V} = \frac{\partial\left(\xi_1^{\,2} v_{\xi_1}\right)}{\xi_1^{\,2}\partial \xi_1} \tag{18}$$

$$\Delta f = \frac{1}{\xi_1^{\,2}}\frac{\partial}{\partial \xi_1}\left(\xi_1^{\,2} \frac{\partial f}{\partial \xi_1}\right) \tag{19}$$

$$\nabla \cdot (\lambda \nabla \phi) = \frac{1}{\xi_1^{\,2}}\frac{\partial}{\partial \xi_1}\left(\xi_1^{\,2} \lambda \frac{\partial \phi}{\partial \xi_1}\right) \tag{20}$$

When the numerical simulation of practical problem is performed, it is possible to avoid the calculation of the derivative in the direction of ξ_3 under the cylindrical coordinates and in the direction of ξ_2 and ξ_3 under the spherical coordinates by selecting the appropriate coordinate system. Therefore, based on the derivative form in the direction of ξ_1 and ξ_2 under the cylindrical coordinates and in the direction of ξ_3 under the spherical coordinates, the numerical scheme of the coordinates (7-9) under the cylindrical coordinates and the spherical coordinates are deduced.

2.2.2 Deduction of the numerical scheme under the cylindrical coordinates

For the divergence terms, by combining (12) and (15), we have:

$$^{i}\left\langle \nabla \cdot \mathbf{V}(\boldsymbol{\xi}) \right\rangle = \frac{1}{{}^{i}\xi_1}\sum_{j=1}^{N} b_j \left[{}^{j}\xi_1 \mathbf{V}\left({}^{j}\boldsymbol{\xi}\right) - {}^{i}\xi_1 \mathbf{V}\left({}^{i}\boldsymbol{\xi}\right)\right] \cdot \nabla_i W_{ij}^{(\mathrm{RKPM})} = \sum_{j=1}^{N} b_j \left[\left({}^{j}\xi_1 / {}^{i}\xi_1\right)^{0.5\times 2} \mathbf{V}\left({}^{j}\boldsymbol{\xi}\right) - \mathbf{V}\left({}^{i}\boldsymbol{\xi}\right)\right] \cdot \nabla_i W_{ij}^{(\mathrm{RKPM})} \tag{21}$$

For the Laplacian operator, by using the approximate relation of first order small quantity (the approximate error is the second order small), we have:

$$1 + \frac{1}{2}\left({}^{j}\xi_1 - {}^{i}\xi_1\right) / {}^{i}\xi_1 \approx \left[1 + \left({}^{j}\xi_1 - {}^{i}\xi_1\right) / {}^{i}\xi_1\right]^{\frac{1}{2}} = \left({}^{j}\xi_1 / {}^{i}\xi_1\right)^{\frac{1}{2}} \tag{22}$$

By combining (14) and (16), we get:

$$\begin{aligned}
&^{i}\left\langle \Delta f\left(\boldsymbol{\xi}\right)\right\rangle = {}^{i}\left\langle \frac{1}{\xi_1}\nabla \cdot \left[\xi_1 \nabla f\left(\boldsymbol{\xi}\right)\right]\right\rangle \\
&= \frac{1}{{}^{i}\xi_1}\sum_{j=1}^{N} b_j \left({}^{i}\xi_1 + {}^{j}\xi_1\right)\left[f\left({}^{i}\boldsymbol{\xi}\right) - f\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\mathbf{r}_{ij} \cdot \nabla_i W_{ij}^{(\mathrm{RKPM})}\right] / \mathbf{r}_{ij}^2 \\
&= \sum_{j=1}^{N} 2b_j \left[1 + \frac{1}{2}\left({}^{j}\xi_1 - {}^{i}\xi_1\right) / {}^{i}\xi_1\right]\left[f\left({}^{i}\boldsymbol{\xi}\right) - f\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\mathbf{r}_{ij} \cdot \nabla_i W_{ij}^{(\mathrm{RKPM})}\right] / \mathbf{r}_{ij}^2 \\
&\approx \sum_{j=1}^{N} 2b_j \left({}^{j}\xi_1 / {}^{i}\xi_1\right)^{\frac{1}{2}}\left[f\left({}^{i}\boldsymbol{\xi}\right) - f\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\mathbf{r}_{ij} \cdot \nabla_i W_{ij}^{(\mathrm{RKPM})}\right] / \mathbf{r}_{ij}^2 \\
&= \sum_{j=1}^{N} 2b_j \left[f\left({}^{i}\boldsymbol{\xi}\right) - f\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\left(\xi_1 / {}^{i}\xi_1\right)^{0.5\times 1} \mathbf{r}_{ij} \cdot \nabla_i W_{ij}^{(\mathrm{RKPM})}\right] / \mathbf{r}_{ij}^2
\end{aligned} \tag{23}$$

For the dissipation term, ignore the second order small quantity $\left[\lambda\left({}^{j}\boldsymbol{\xi}\right) - \lambda\left({}^{i}\boldsymbol{\xi}\right)\right]\left[{}^{j}\boldsymbol{\xi} - {}^{i}\boldsymbol{\xi}\right] / {}^{i}\boldsymbol{\xi}$ and approximate the coefficient in front of the first order small quantity as follows (the resulting approximation error is second order small):

$$\lambda\left({}^{j}\boldsymbol{\xi}\right) / \left[\lambda\left({}^{i}\boldsymbol{\xi}\right) + \lambda\left({}^{j}\boldsymbol{\xi}\right)\right] \approx 1/2 \tag{24}$$

By combining (14) and (17), we get:

$$\begin{aligned}
&^{i}\left\langle \nabla \cdot \left[\lambda\left(\boldsymbol{\xi}\right)\nabla \varphi\left(\boldsymbol{\xi}\right)\right]\right\rangle = {}^{i}\left\langle \frac{1}{\xi_1}\nabla \cdot \left[\xi_1 \lambda\left(\boldsymbol{\xi}\right)\nabla \varphi\left(\boldsymbol{\xi}\right)\right]\right\rangle \\
&= \frac{1}{{}^{i}\xi_1}\sum_{j=1}^{N} b_j \left[{}^{i}\xi_1 \lambda\left({}^{i}\boldsymbol{\xi}\right) + {}^{j}\xi_1 \lambda\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\varphi\left({}^{i}\boldsymbol{\xi}\right) - \varphi\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\mathbf{r}_{ij} \cdot \nabla_i W_{ij}^{(\mathrm{RKPM})}\right] \\
&\approx \sum_{j=1}^{N} b_j \left[\lambda\left({}^{i}\boldsymbol{\xi}\right) + \lambda\left({}^{j}\boldsymbol{\xi}\right) + \lambda\left({}^{j}\boldsymbol{\xi}\right)\left({}^{j}\xi_1 - {}^{i}\xi_1\right) / {}^{i}\xi_1\right]\left[\varphi\left({}^{i}\boldsymbol{\xi}\right) - \varphi\left({}^{j}\boldsymbol{\xi}\right)\right]\left[\mathbf{r}_{ij} \cdot \nabla_i W_{ij}^{(\mathrm{RKPM})}\right]
\end{aligned}$$

$$\begin{aligned}&\approx \sum_{j=1}^{N} b_j \left[1+\frac{1}{2}\left({}^{j}\xi_1-{}^{i}\xi_1\right)/{}^{i}\xi_1\right]\left[\lambda\left({}^{i}\xi\right)+\lambda\left({}^{j}\xi\right)\right]\left[\varphi\left({}^{i}\xi\right)-\varphi\left({}^{j}\xi\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]\\&\approx \sum_{j=1}^{N} b_j \left[\lambda\left({}^{i}\xi\right)+\lambda\left({}^{j}\xi\right)\right]\left[\varphi\left({}^{i}\xi\right)-\varphi\left({}^{j}\xi\right)\right]\left[\left({}^{j}\xi_1/{}^{i}\xi_1\right)^{0.5\times 1}\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]\end{aligned} \tag{25}$$

Considering that the derivative calculation of ξ_2 under the cylindrical coordinates is the same as the one under the Cartesian coordinates, and the derivative of ξ_3 is not involved, the specific expressions (7-9) corresponding to the cylindrical coordinates can be further obtained on the basis of (21), (23), and (25).

2.2.3 Deduction of the numerical scheme under the spherical coordinates

For the divergence terms, by combining (12) and (18), we have:

$${}^{i}\left\langle\nabla\cdot\mathbf{V}(\xi)\right\rangle=\frac{1}{{}^{i}\xi_1^2}\sum_{j=1}^{N} b_j\left[{}^{j}\xi_1^2\mathbf{V}\left({}^{j}\xi\right)-{}^{i}\xi_1^2\mathbf{V}\left({}^{i}\xi\right)\right]\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}=\sum_{j=1}^{N} b_j\left[\left({}^{j}\xi_1/{}^{i}\xi_1\right)^{1.0\times 2}\mathbf{V}\left({}^{j}\xi\right)-\mathbf{V}\left({}^{i}\xi\right)\right]\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})} \tag{26}$$

For the Laplacian operator, also using the approximate relation of first order small quantity (the approximate error is the second order small) yields:

$$1+\frac{1}{2}\left({}^{j}\xi_1^2-{}^{i}\xi_1^2\right)/{}^{i}\xi_1^2\approx\left[1+\left({}^{j}\xi_1^2-{}^{i}\xi_1^2\right)/{}^{i}\xi_1^2\right]^{\frac{1}{2}}=\left[{}^{j}\xi_1^2/{}^{i}\xi_1^2\right]^{\frac{1}{2}} \tag{27}$$

By combining (13) and (19), we get:

$$\begin{aligned}&{}^{i}\left\langle\Delta f\left(\xi\right)\right\rangle={}^{i}\left\langle\frac{1}{\xi_1^2}\nabla\cdot\left[\xi_1^2\nabla f\left(\xi\right)\right]\right\rangle\\&=\frac{1}{{}^{i}\xi_1^2}\sum_{j=1}^{N} b_j\left[{}^{i}\xi_1^2+{}^{j}\xi_1^2\right]\left[f\left({}^{i}\xi\right)-f\left({}^{j}\xi\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]/\mathbf{r}_{ij}^2\\&=\sum_{j=1}^{N} 2b_j\left[1+\frac{1}{2}\left({}^{j}\xi_1^2-{}^{i}\xi_1^2\right)/{}^{i}\xi_1^2\right]\left[f\left({}^{i}\xi\right)-f\left({}^{j}\xi\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]/\mathbf{r}_{ij}^2\\&\approx\sum_{j=1}^{N} 2b_j\left({}^{j}\xi_1^2/{}^{i}\xi_1^2\right)^{\frac{1}{2}}\left[f\left({}^{i}\xi\right)-f\left({}^{j}\xi\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]/\mathbf{r}_{ij}^2\\&=\sum_{j=1}^{N} 2b_j\left[f\left({}^{i}\xi\right)-f\left({}^{j}\xi\right)\right]\left[\left({}^{j}\xi_1/{}^{i}\xi_1\right)^{1.0\times 1}\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]/\mathbf{r}_{ij}^2\end{aligned} \tag{28}$$

For the dissipation term, by combining (14) and (20) and following the derivation of equation (25), we get:

$$\begin{aligned}&{}^{i}\left\langle\nabla\cdot\left[\lambda\left(\xi\right)\nabla\varphi\left(\xi\right)\right]\right\rangle={}^{i}\left\langle\frac{1}{\xi_1^2}\nabla\cdot\left[\xi_1^2\lambda\left(\xi\right)\nabla\varphi\left(\xi\right)\right]\right\rangle\\&=\frac{1}{{}^{i}\xi_1^2}\sum_{j=1}^{N} b_j\left[{}^{i}\xi_1^2\lambda\left({}^{i}\xi\right)+{}^{j}\xi_1^2\lambda\left({}^{j}\xi\right)\right]\left[\varphi\left({}^{i}\xi\right)-\varphi\left({}^{j}\xi\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]\\&\approx\sum_{j=1}^{N} b_j\left[\lambda\left({}^{i}\xi\right)+\lambda\left({}^{j}\xi\right)+\lambda\left({}^{j}\xi\right)\left({}^{j}\xi_1^2-{}^{i}\xi_1^2\right)/{}^{i}\xi_1^2\right]\left[\varphi\left({}^{i}\xi\right)-\varphi\left({}^{j}\xi\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]\\&\approx\sum_{j=1}^{N} b_j\left[1+\frac{1}{2}\left({}^{j}\xi_1^2-{}^{i}\xi_1^2\right)/{}^{i}\xi_1^2\right]\left[\lambda\left({}^{i}\xi\right)+\lambda\left({}^{j}\xi\right)\right]\left[\varphi\left({}^{i}\xi\right)-\varphi\left({}^{j}\xi\right)\right]\left[\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]\\&\approx\sum_{j=1}^{N} b_j\left[\lambda\left({}^{i}\xi\right)+\lambda\left({}^{j}\xi\right)\right]\left[\varphi\left({}^{i}\xi\right)-\varphi\left({}^{j}\xi\right)\right]\left[\left({}^{j}\xi_1/{}^{i}\xi_1\right)^{1.0\times 1}\mathbf{r}_{ij}\cdot\nabla_i W_{ij}^{(\mathrm{RKPM})}\right]\end{aligned} \tag{29}$$

Considering that the derivative calculation of ξ_2 and ξ_3 is not involved, the specific expressions (7-9) corresponding to the spherical coordinates can be further obtained on the basis of (26), (28), and (29).

3. Convergence analysis method

Since the approximation of the function and its derivative does not require iterative computation and time advance solution, there is no instability problem in the computational process. Therefore, the compatibility and convergence of the numerical scheme are equivalent here. Convergence analysis is the main content validated by this method. The algorithm involved in this paper does not include the complete solution to the differential equations. Therefore, convergence of the numerical scheme here means that the exact solution of the discrete equation can be approximated to the exact solution of the mathematical model with the discrete scale tending to zero. If h refers to the discrete scale, u_h represents the exact solution of the numerical scheme, and u_* be the exact solution of the mathematical model, then, the discrete error ε_h is:

$$\varepsilon_h = u_h - u_* \tag{30}$$

Convergence of the numerical scheme can be expressed as:

$$\lim_{h\to 0}\varepsilon_h = 0 \tag{31}$$

Numerical schemes of this paper don't involve iterative calculations, and rounding errors are negligible. Therefore, the calculation results in accordance with the numerical scheme and mathematical model are referred to as numerical solution u_h and exact solution u_*. According to the literature [4], the numerical solution can be based on the convergence limit for Taylor expansion:

$$u_h = u_{h=0} + \frac{\partial u}{\partial h}\Big|_{h=0} h + \frac{\partial^2 u}{\partial h^2}\Big|_{h=0}\frac{h^2}{2} + \frac{\partial^3 u}{\partial h^3}\Big|_{h=0}\frac{h^3}{6} + O\left(h^4\right) \tag{32}$$

For the convergent numerical scheme, $u_{h=0} = u_*$, assuming that the convergence accuracy level is p, and the error term below p is zero, then:

$$\varepsilon_h = g_p h^p + O\left(h^{p+1}\right) \tag{33}$$

For the coarse and fine discrete scale, the scale ratio is:

$$\eta = h_{\text{coarse}} / h_{\text{fine}} \tag{34}$$

The posteriori of the convergence accuracy level can be obtained from Eq. (33) as:

$$\hat{p} \approx \ln\left(\varepsilon_{\eta h} / \varepsilon_h\right) / \ln(\eta) \tag{35}$$

Assume Λ is the maximum absolute value of the objective function or derivative in the computational domain. The error magnitude and the accuracy level are evaluated by the error norm $\|\varepsilon_h\|$ and the relative error $\|\tilde{\varepsilon}_h\| = \|\varepsilon_h\| / \Lambda$ in this paper. If there are M discrete elements in the computational domain, then:

$$\|\varepsilon_h\| = \frac{1}{M}\sum_{i=1}^{M}\left|{}^i u_h - {}^i u_*\right| \tag{36}$$

In the asymptotic convergence region, the approximate linear relationship can be obtained from Eq. (33) by substituting $\|\varepsilon_h\|$ for ε_h and subtracting the higher order term as:

$$\ln\|\varepsilon_h\| \approx \ln\|g_p\| + p\ln h \tag{37}$$

By using the error norm instead of the discrete error in the above formula, a general expression of convergence accuracy level is obtained as:

$$\hat{p} \approx \ln\left(\|\varepsilon_{\eta h}\| / \|\varepsilon_h\|\right) / \ln(\eta) \tag{38}$$

Only when the discrete scale is small to a certain extent to assure that the numerical calculation is performed in the asymptotic convergence region, and the main extension on the right end of the equation (32) does not change in the sign, the analytical results based on Equations (37) and (38) may be reasonable.

Convergence analysis can be performed by the following three technical ways:

1) Select the different function form and the appropriate discrete scale, and verify the correctness of the algorithm by the consistency of the numerical solution and the exact solution.

2) The convergence of the calculation scheme is judged by the linear relationship between $\ln\|\varepsilon_h\|$ and $\ln h$ when the discrete scale is small enough.

3) Obtain the convergence accuracy level based on the error norms and scale ratio of the two discrete scales in the asymptotic convergence region.

During code validation process, it is necessary to try a variety of functions so that sufficient evidence could be got to illustrate the correctness of the numerical algorithm and the program code. Due to paragraph constraints, only the quantitative analyses of specific functions are listed in the paper.

4. One-dimensional model validation of the code

4.1. Original function and calculation parameters

For a one-dimensional model, the value of the coordinate variable is ranged between $0.0 \le \xi_1 \le 1.0$, and the selected original function is as follows (39-42):

$$f(\xi_1) = \xi_1^2 \sin(2.5\pi\xi_1) \tag{39}$$

$$\lambda(\xi_1) = \sin(6\pi\xi_1) \tag{40}$$

$$\phi(\xi_1) = \cos(2.5\pi\xi_1) \tag{41}$$

$$V_1(\xi_1) = \xi_1^2 \sin(2.5\pi\xi_1) \tag{42}$$

where $V_1(\xi_1)$ represents the first component of the vector function $\mathbf{V}(\xi_1, \xi_2, \xi_3)$ in the one-dimensional case. In this paper,

the kernel estimation meshfree method is used to apply the lowest order correction function to the kernel function; that is, the correction order n = 0. The discrete scales of the numerical calculation h_1, h_2, h_3, and h_4 correspond to the discrete numbers of 50, 100, 200, and 400, respectively; l_1, l_2, l_3, and l_4 are the corresponding support domain scales. A smaller support domain scale factor $\delta = l_i / h_i = 1.0$, i=1~4 is adopted here, and only two adjacent elements fall into the support region on the left and right. The calculation parameters are shown in Table 1.

Table 1. Computation parameters for 1-dimensional models

n	δ	h_1	h_2	h_3	h_4	l_1	l_2	l_3	l_4
0	1.0	0.02	0.01	0.005	0.0025	δh_1	δh_2	δh_3	δh_4

4.2. Consistency evaluation

Figures (1-3) depict the numerical results based on the selected original function in addition to the discrete scale corresponding to h_1 and other calculation parameters in Table 1. Numerical experiments for any original function show that the numerical solution and the exact solution of the one-dimensional model agree well at the appropriate discrete scale.

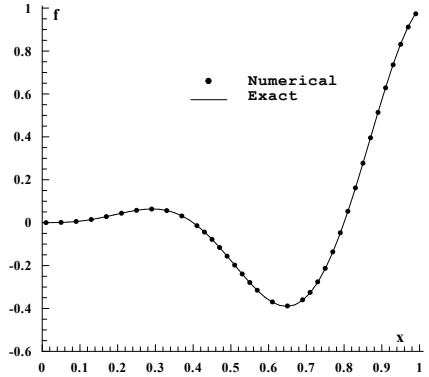


Fig. 1(a). Numerical results of $f(r)$

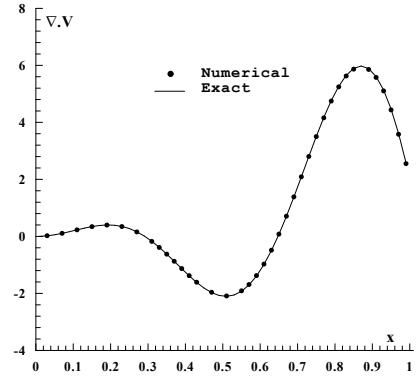


Fig. 1(b). Numerical results of $\nabla \cdot \boldsymbol{V}(r)$

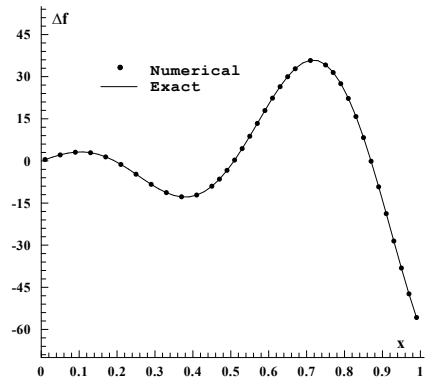


Fig. 1(c). Numerical results of $\Delta f(r)$

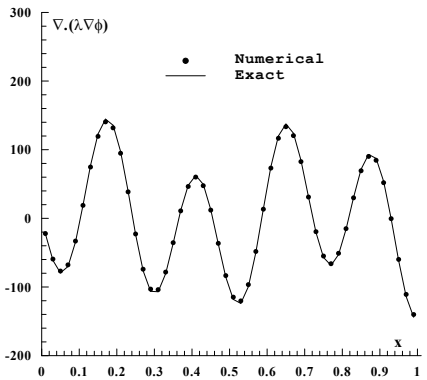


Fig. 1(d). Numerical results of $\nabla \cdot [\lambda(r) \nabla \phi(r)]$

Fig. 1. Numerical results with Cartesian coordinates

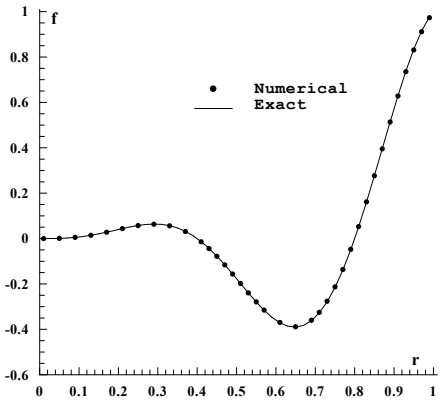


Fig. 2(a). Numerical results of $f(r)$

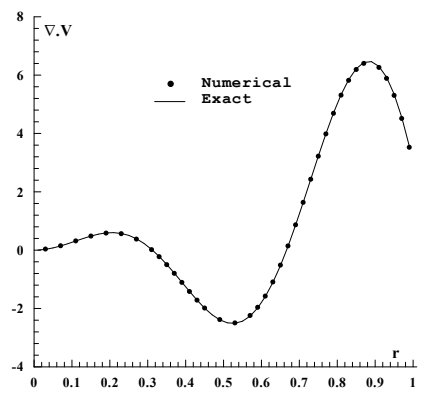


Fig. 2(b). Numerical results of $\nabla \cdot \boldsymbol{V}(r)$

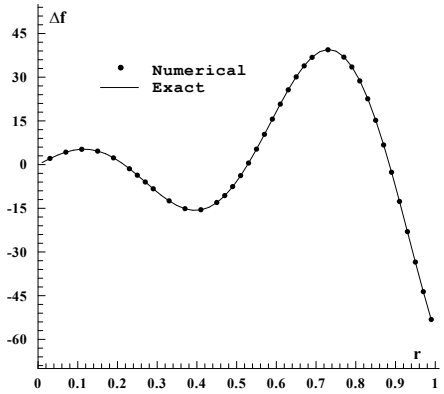


Fig. 2(c). Numerical results of $\Delta f(r)$

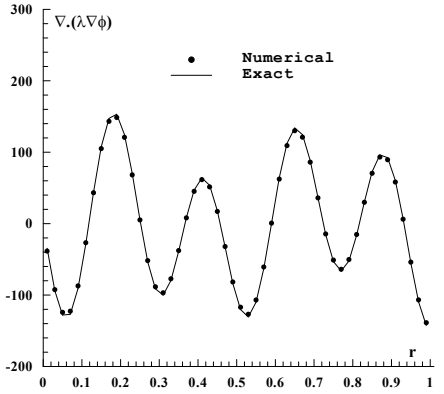


Fig. 2(d). Numerical results of $\nabla\cdot[\lambda(r)\nabla\phi(r)]$

Fig. 2. Numerical results with cylindrical coordinates

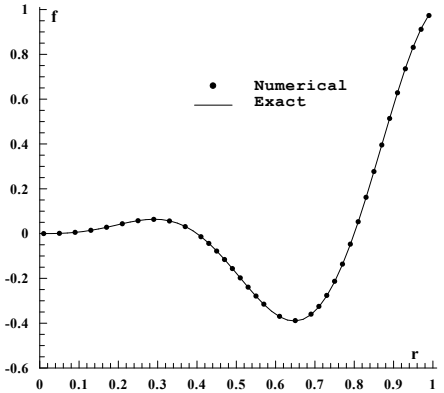


Fig. 3(a). Numerical results of $f(r)$

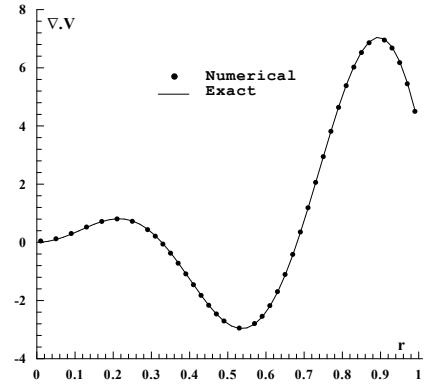


Fig. 3(b). Numerical results of $\nabla\cdot\boldsymbol{V}(r)$

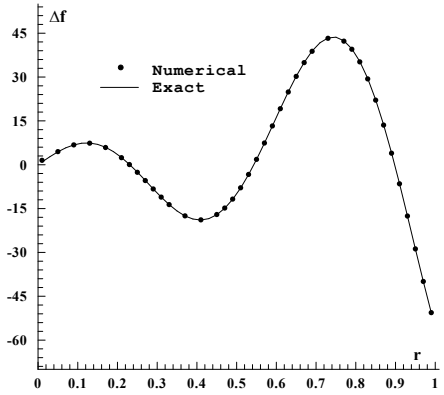


Fig. 3(c). Numerical results of $\Delta f(r)$

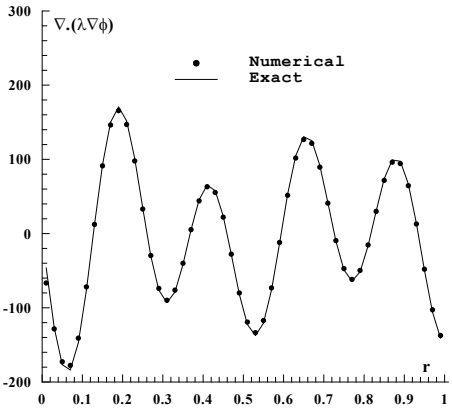


Fig. 3(d). Numerical results of $\nabla\cdot[\lambda(r)\nabla\phi(r)]$

Fig. 3. Numerical results with spherical coordinates

4.3. Convergence evaluation

Convergence evaluation consists of two basic aspects: one is to evaluate whether the numerical scheme converges, and the other is to evaluate the convergence accuracy level. The discrete scale of the numerical calculation can never be zero, but if the discrete scale is small enough to a certain degree, and the numerical calculation is in the asymptotic convergence region, logarithm of the error norm and the discrete scale would show the linear relationship as shown in Eq. (37); therefore, convergence of the computational scheme can be judged, and the accuracy level of the numerical convergence is quantified by Eq. (38).

As can be seen from Fig. 4, the error norm and the discrete scale of the one-dimensional model are approximately linear in logarithmic coordinates. Table 2 lists the quantization results for the error norm $\|\varepsilon_h\|$ and the convergence accuracy level $\hat{p}$, where the calculation information comes from $\|\varepsilon_{h=0.005}\|$, $\|\varepsilon_{h=0.0025}\|$, and $\eta=2$.

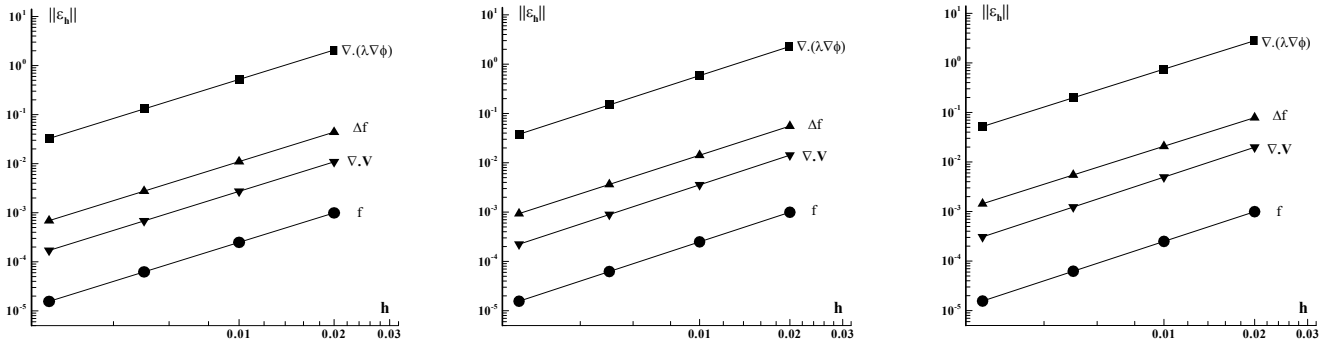


Fig. 4(a). Cartesian **Fig. 4(b).** Cylindrical **Fig. 4(c).** Spherical

Fig. 4. Error norms vs. discretization scales in 1-dimensional models

Table 2. Discretization errors and accuracy orders of 1-dimensional models

Object	Coordinates	$\|\varepsilon_{h=0.02}\|$	$\|\varepsilon_{h=0.01}\|$	$\|\varepsilon_{h=0.005}\|$	$\|\varepsilon_{h=0.0025}\|$	$\|\tilde{\varepsilon}_{h=0.01}\|$	$\hat{p}$
f	Cartesian	9.925e-4	2.493e-4	6.234e-5	1.559e-5	0.025%	1.99954
	Cylindrical	9.925e-4	2.493e-4	6.234e-5	1.559e-5	0.025%	1.99954
	Spherical	9.925e-4	2.493e-4	6.234e-5	1.559e-5	0.025%	1.99954
Δf	Cartesian	4.398e-2	1.103e-2	2.759e-3	6.900e-4	0.019%	1.99948
	Cylindrical	5.551e-2	1.424e-2	3.646e-3	9.328e-4	0.025%	1.96667
	Spherical	7.855e-2	2.077e-2	5.469e-3	1.435e-3	0.038%	1.93023
$\nabla \cdot \boldsymbol{V}$	Cartesian	1.091e-2	2.728e-3	6.821e-4	1.705e-4	0.046%	2.00021
	Cylindrical	1.425e-2	3.567e-3	8.919e-4	2.230e-4	0.055%	1.99984
	Spherical	1.985e-2	4.940e-3	1.231e-3	3.072e-4	0.070%	2.00258
$\nabla \cdot (\lambda \nabla \phi)$	Cartesian	2.075e+0	5.212e-1	1.304e-1	3.262e-2	0.359%	1.99912
	Cylindrical	2.286e+0	5.848e-1	1.492e-1	3.794e-2	0.377%	1.97546
	Spherical	2.806e+0	7.482e-1	1.978e-1	5.199e-2	0.438%	1.92773

5. Two-dimensional model validation of the code

5.1. Original function and calculation parameters

The two-dimensional model selects the original function shown in (43-47), and the range of the coordinate variables are $0.0 \le \xi_1 \le 1.0$ and $0.0 \le \xi_2 \le 0.5$.Two kinds of coordinate systems, Cartesian and cylindrical, are involved in the numerical calculation. The calculation parameters are shown in Table 3.

$$f\left(\xi_1,\xi_2\right)=\cos\left(2\pi\xi_1\right)\sin\left(6\pi\xi_2\right) \tag{43}$$

$$\lambda\left(\xi_1,\xi_2\right)=\sin\left[4\pi\left(\xi_1+\xi_2\right)\right] \tag{44}$$

$$\varphi\left(\xi_1,\xi_2\right)=\cos\left(2\pi\xi_1\xi_2\right) \tag{45}$$

$$V_1\left(\xi_1,\xi_2\right)=\sin\left(2\pi\xi_1\right)\cos\left(2\pi\xi_2\right) \tag{46}$$

$$V_2\left(\xi_1,\xi_2\right)=\cos\left(2\pi\xi_1\right)\sin\left(6\pi\xi_2\right) \tag{47}$$

where $V_1\left(\xi_1,\xi_2\right)$ and $V_2\left(\xi_1,\xi_2\right)$ represent the first two components of the vector function $\boldsymbol{V}\left(\xi_1,\xi_2,\xi_3\right)$ in the two-dimensional case.

Table 3. Computation parameters for 2-dimensional models

n	δ	h_1	h_2	h_3	h_4	l_1	l_2	l_3	l_4
0	1.5	0.02	0.01	0.0067	0.005	δh_1	δh_2	δh_3	δh_4

5.2. Consistency evaluation

Figures (5-6) present the numerical results based on the selected original function in addition to the discrete scale corresponding to h1 and other calculation parameters in Table 3. In the case of two-dimensional model, consistency of the numerical solution and the exact solution could be represented by the approximation degree of the contour. Numerical experiments for any original function show that the numerical solution and the exact solution of the two-dimensional model agree well at the appropriate discrete scale.

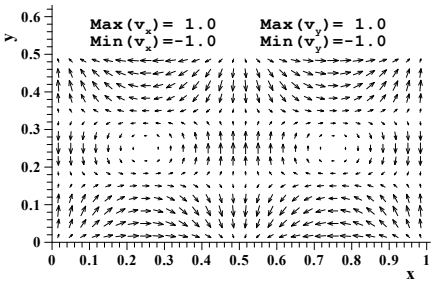


Fig. 5(a). Numerical results of $\boldsymbol{V}(x,y)$

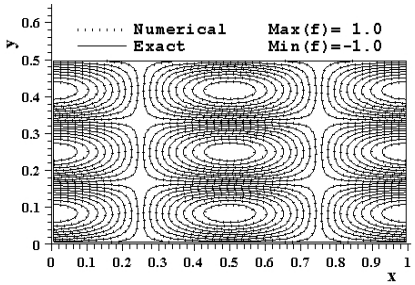


Fig. 5(b). Numerical results of $f(x,y)$

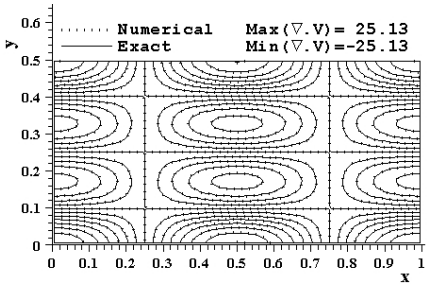


Fig. 5(c). Numerical results of $\nabla\cdot\boldsymbol{V}(x,y)$

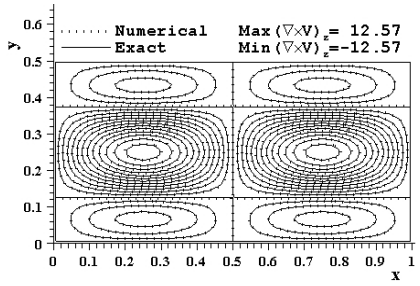


Fig. 5(d). Numerical results of $\left[\nabla\times\boldsymbol{V}(x,y)\right]_z$

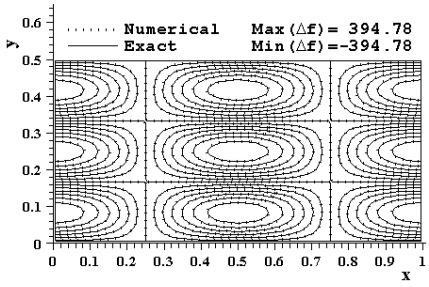


Fig. 5(e). Numerical results of $\Delta f(x,y)$

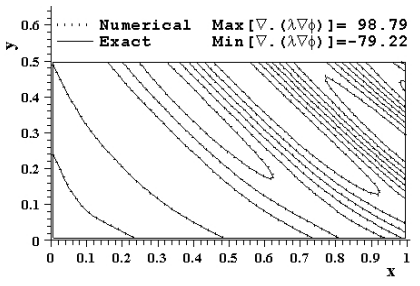


Fig. 5(f). Numerical results of $\nabla\cdot\left[\lambda(x,y)\nabla\phi(x,y)\right]$

Fig. 5. Numerical results with Cartesian coordinates

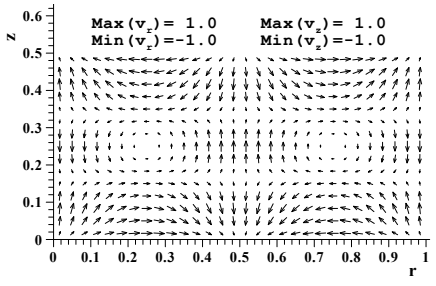


Fig. 6(a). Numerical results of $\boldsymbol{V}(x,y)$

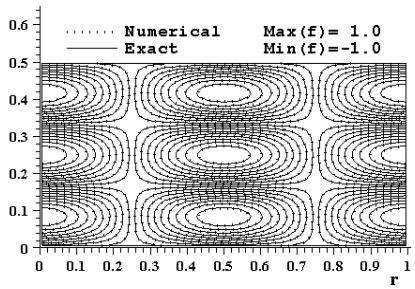


Fig. 6(b). Numerical results of $f(x,y)$

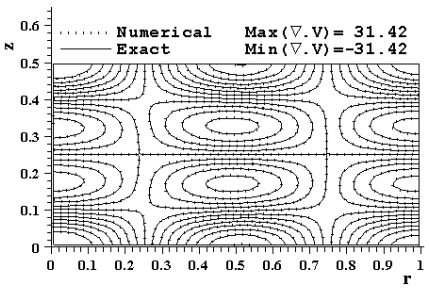


Fig. 6(c). Numerical results of $\nabla\cdot\boldsymbol{V}(x,y)$

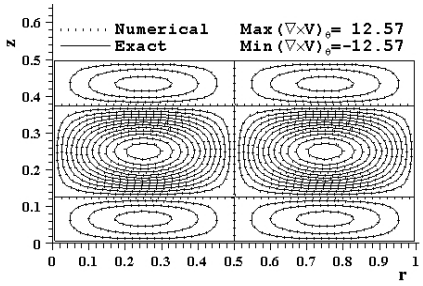


Fig. 6(d). Numerical results of $\left[\nabla\times\boldsymbol{V}(x,y)\right]_z$

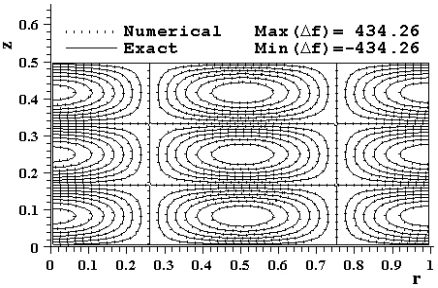


Fig. 6(e). Numerical results of $\Delta f(x,y)$

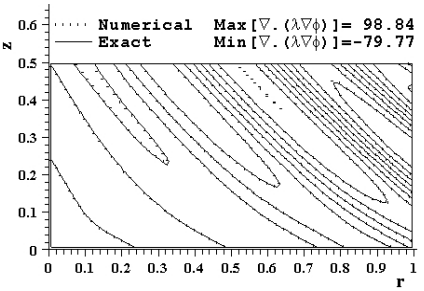


Fig. 6(f). Numerical results of $\nabla\cdot\left[\lambda(x,y)\nabla\phi(x,y)\right]$

Fig. 6. Numerical results with cylindrical coordinates

5.3. Convergence evaluation

As can be seen from Fig. 7, the error norm and the discrete scale of the two-dimensional model are approximately linear in logarithmic coordinates. Table 4 lists the quantization results for the error norm $\|\varepsilon_h\|$ and the convergence accuracy level $\hat{p}$, where the calculation information comes from $\|\varepsilon_{h=0.0067}\|$, $\|\varepsilon_{h=0.005}\|$ and $\eta=1.34$.

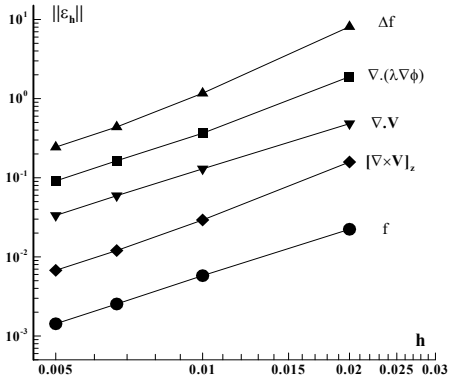


Fig. 7(a). Cartesian

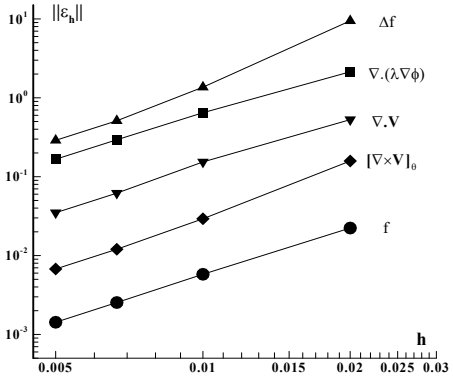


Fig. 7(b). Cylindrical

Fig. 7. Discretization scales vs. error norms in 2-dimensional models

6. Three-dimensional model validation of the code

6.1. Original function and calculation parameters

The three-dimensional model selects the original function shown in (48-53), and the range of the coordinate variables are $0.0 \le \xi_1 \le 1.0$, $0.0 \le \xi_2 \le 1.0$, and $0.0 \le \xi_3 \le 1.0$ .Only Cartesian coordinate system is involved in the numerical calculation, and the calculation parameters are shown in Table 5, where there are two support domain scale factors.

$$f\left(\xi_1,\xi_2,\xi_3\right)=\sin\left(2\pi\xi_1\xi_2\xi_3\right) \tag{48}$$

$$\lambda\left(\xi_1,\xi_2,\xi_3\right)=\sin\left[2\pi\left(\xi_1+\xi_2+\xi_3\right)\right] \tag{49}$$

$$\phi\left(\xi_1,\xi_2,\xi_3\right)=\sin\left(2\pi\xi_1\xi_2\xi_3\right) \tag{50}$$

$$V_1\left(\xi_1,\xi_2,\xi_3\right)=\sin\left[2\pi\left(\xi_1+\xi_2\right)\right] \tag{51}$$

$$V_2\left(\xi_1,\xi_2,\xi_3\right)=\sin\left[2\pi\left(\xi_2+\xi_3\right)\right] \tag{52}$$

$$V_3\left(\xi_1,\xi_2,\xi_3\right)=\sin\left[2\pi\left(\xi_1+\xi_3\right)\right] \tag{53}$$

where $V_1\left(\xi_1,\xi_2,\xi_3\right)$, $V_2\left(\xi_1,\xi_2,\xi_3\right)$, and $V_3\left(\xi_1,\xi_2,\xi_3\right)$ represent the three components of the vector function $\boldsymbol{V}\left(\xi_1,\xi_2,\xi_3\right)$ in the three-dimensional case.

Table 4. Discretization errors and accuracy orders of 2-dimensional models

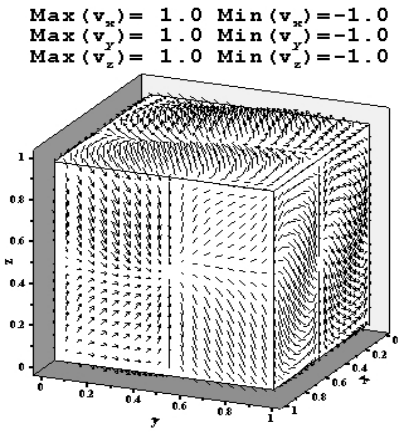
Object	Coordinates	$\|\varepsilon_{h=0.02}\|$	$\|\varepsilon_{h=0.01}\|$	$\|\varepsilon_{h=0.0067}\|$	$\|\varepsilon_{h=0.005}\|$	$\|\tilde{\varepsilon}_{h=0.01}\|$	$\hat{p}$
f	Cartesian	2.234e-2	5.792e-3	2.539e-3	1.428e-3	0.597%	2.00045
	Cylindrical	2.234e-2	5.792e-3	2.539e-3	1.428e-3	0.597%	2.00045
Δf	Cartesian	8.113e-0	1.169e-0	4.385e-1	2.438e-1	0.296%	2.04049
	Cylindrical	9.461e-0	1.364e-0	5.116e-1	2.893e-1	0.314%	1.98165
$\nabla\cdot\boldsymbol{V}$	Cartesian	4.829e-1	1.296e-1	5.933e-2	3.340e-2	0.516%	1.99722
	Cylindrical	5.288e-1	1.537e-1	6.184e-2	3.494e-2	0.489%	1.98472
$[\nabla\times\boldsymbol{V}]_{\xi_3}$	Cartesian	1.581e-1	2.929e-2	1.203e-2	6.765e-3	0.233%	2.00097
	Cylindrical	1.581e-1	2.929e-2	1.203e-2	6.765e-3	0.233%	2.00097
$\nabla\cdot(\lambda\nabla\varphi)$	Cartesian	1.892e-0	3.649e-1	1.638e-1	9.121e-2	0.369%	2.03519
	Cylindrical	2.137e+0	6.469e-1	2.939e-1	1.664e-1	0.650%	1.97751

Table 5. Computation parameters for 3-dimensional models

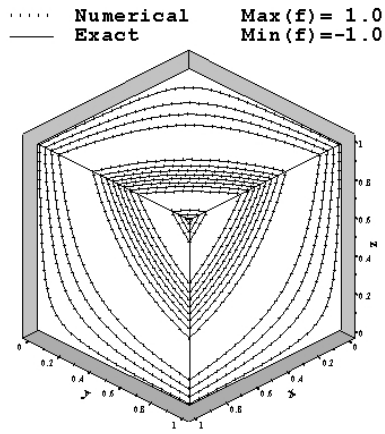
n_c	δ_1	δ_2	δ_3	h_1	h_2	h_3	h_4	l_1	l_2	l_3	l_4
0	1.5	2.0	2.5	0.02	0.0125	0.01	0.008	δh_1	δh_2	δh_3	δh_4

6.2. Consistency evaluation

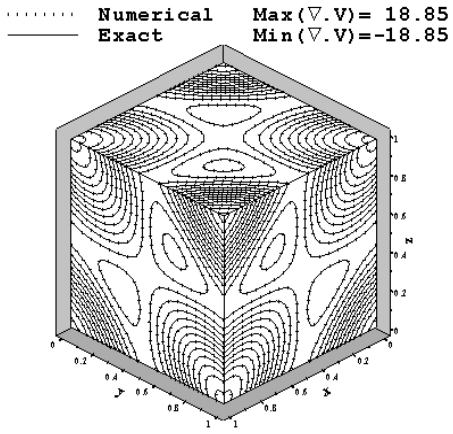
Figures 8(a-k) show the comparison results of numerical and accurate solutions based on the selected original function in addition to the discrete scale corresponding to h=h_1, $\delta=\delta_1$, and other calculation parameters in Table 5. Numerical experiments for any original function show that the numerical solution and the exact solution of the three-dimensional model agree well at the appropriate discrete scale.



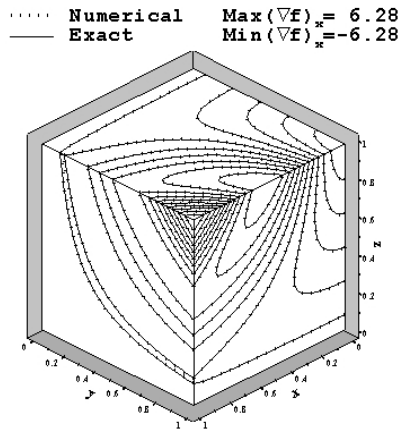
(a). Numerical results of $\boldsymbol{V}(x,y,z)$



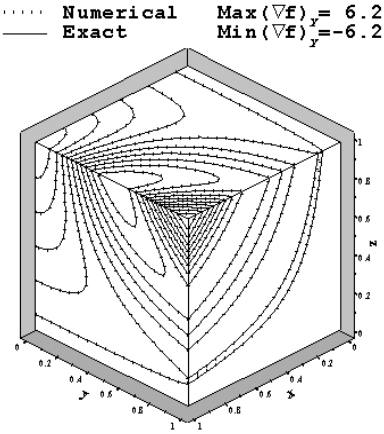
(b). Numerical results of $f(x,y,z)$



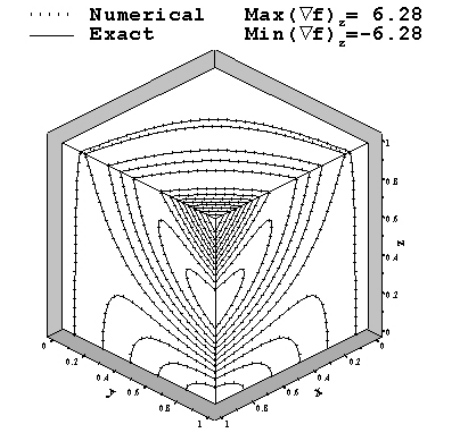
(c). Numerical results of $\nabla\cdot\boldsymbol{V}(x,y,z)$



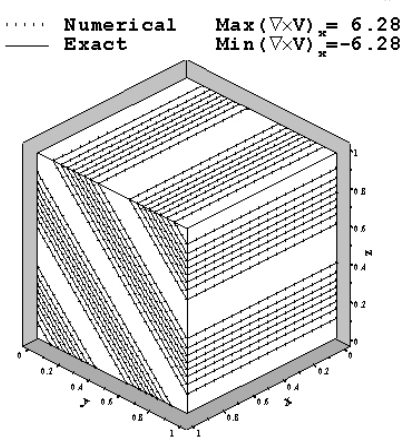
(d). Numerical results of $\left[\nabla f(x,y,z)\right]_x$



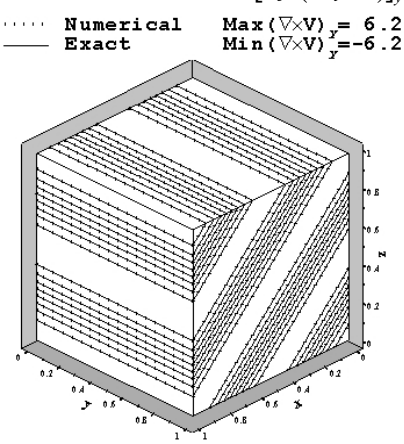
(e). Numerical results of $\left[\nabla f(x,y,z)\right]_y$



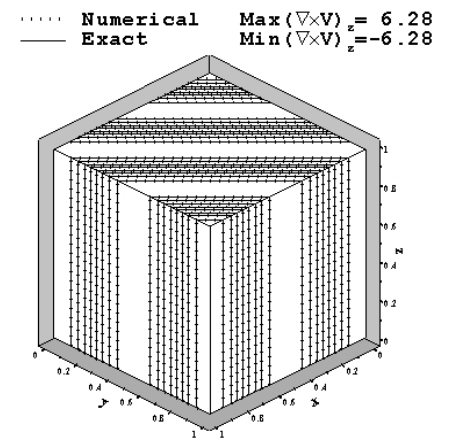
(f). Numerical results of $\left[\nabla f(x,y,z)\right]_z$



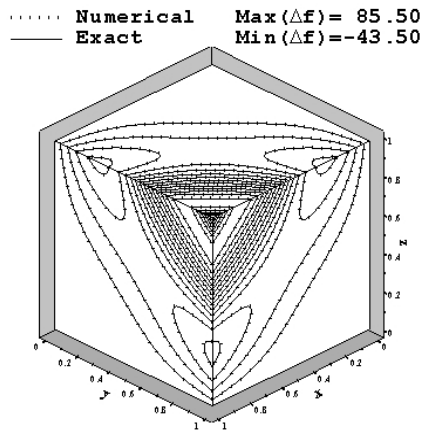
(g). Numerical results of $\left[\nabla\times\boldsymbol{V}(x,y,z)\right]_x$



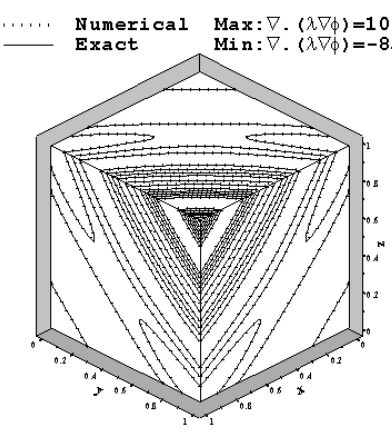
(h). Numerical results of $\left[\nabla\times\boldsymbol{V}(x,y,z)\right]_y$



(i). Numerical results of $\left[\nabla\times\boldsymbol{V}(x,y,z)\right]_z$



(j). Numerical results of $\Delta f(x,y,z)$



(k). Numerical results of $\nabla\cdot\left[\lambda(x,y,z)\nabla\phi(x,y,z)\right]$

Fig. 8. Numerical results with Cartesian coordinate

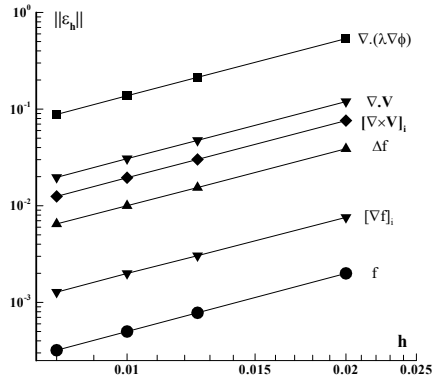


Fig. 9(a). $\delta = 2.0$

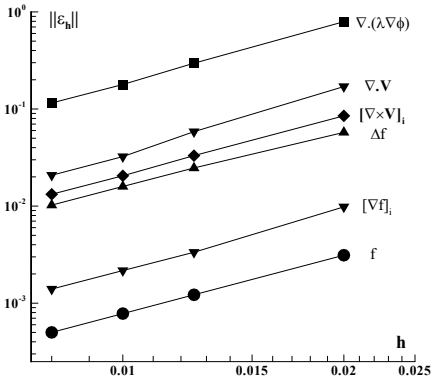


Fig. 9(b). $\delta = 2.5$

Fig. 9. Discretization scales vs. error norms in 3-dimensional models

Table 6. Discretization errors and accuracy orders of 3-dimensional models

Object	δ	$\|\varepsilon_{h=0.02}\|$	$\|\varepsilon_{h=0.0125}\|$	$\|\varepsilon_{h=0.01}\|$	$\|\varepsilon_{h=0.008}\|$	$\|\tilde{\varepsilon}_{h=0.01}\|$	$\hat{p}$
f	2.0	1.997e-003	7.809e-004	4.999e-004	3.200e-004	0.050%	1.99910
	2.5	3.119e-003	1.220e-003	7.814e-004	5.002e-004	0.078%	1.99907
Δf	2.0	3.888e-002	1.542e-002	1.001e-002	6.486e-003	0.012%	1.94466
	2.5	5.744e-002	2.469e-002	1.593e-002	1.024e-002	0.019%	1.98035
$[\nabla f]_i$	2.0	7.582e-003	3.043e-003	1.993e-003	1.276e-003	0.032%	1.99831
	2.5	9.824e-003	3.344e-003	2.168e-003	1.398e-003	0.035%	1.96628
$\nabla\cdot\mathbf{V}$	2.0	1.197e-001	4.748e-002	3.071e-002	1.966e-002	0.163%	1.99872
	2.5	1.703e-001	5.842e-002	3.233e-002	2.079e-002	0.172%	1.97865
$[\nabla\times\boldsymbol{V}]_i$	2.0	7.604e-002	3.014e-002	1.950e-002	1.249e-002	0.311%	1.99641
	2.5	8.502e-002	3.308e-002	2.053e-002	1.324e-002	0.327%	1.96575
$\nabla\cdot(\lambda\nabla\phi)$	2.0	5.363e-001	2.132e-001	1.375e-001	8.809e-002	0.127%	1.99542
	2.5	7.929e-001	2.961e-001	1.788e-001	1.155e-001	0.165%	1.95837

6.3 Convergence evaluation

As can be seen from Fig. 9, the error norm and the discrete scale of the three-dimensional model are approximately linear in logarithmic coordinates. Table 6 lists the quantization results for the error norm $\|\varepsilon_h\|$ and the convergence accuracy level $\hat{p}$, where the calculation information comes from $\|\varepsilon_{h=0.01}\|$, $\|\varepsilon_{h=0.008}\|$ and $\eta = 1.25$.

7. Conclusion

There are three factors influencing the reliability of the calculation results when applying the meshfree method to approximate the function and its derivative: the first is the numerical calculation scheme, the second is the discrete scale, and the third is the support domain scale factor. Numerical experiments show that the code can lead the calculation results to be consistent with the original function and its first and second order derivative functions for arbitrary function form. Moreover, the correctness of the calculation scheme and the program code is verified. In the case of different coordinate systems and spatial dimensions, when the discrete scale is selected as h = 0.01, the relative errors of the selected function and its first and second order derivative values are not more than 0.65%. According to the analysis of the varying trend of numerical results under different discrete scales, the result of observation with second order convergence accuracy is obtained. Numerical experiments also show that both the order of the RKPM correction function and the value of the support domain scale factor have an effect on the error of the derivative estimation, but not on the convergence accuracy level. There is an optimal range of values for the support domain scale factor in terms of different function forms and spatial dimensions, the larger one can provide more profound function distribution information, while the convergence of the calculation scheme can be better displayed when the discrete scale is further reduced. Although smaller support domain scale factor corresponds to smaller error, if the discrete scale is small enough to a certain extent, the error may not be decreasing in accordance with the convergence order, and the convergence block is found in the discrete scale of a certain range. The calculation scheme in this paper only applies the zero-order correction function of the RKPM method.

Since the method constructs the correction function based on the function rather than the derivative estimation principle of the derivative, when the support domain scale factor is smaller, neighborhood particles in the support domain are less, and approximate estimation of the spatial derivatives may deviate from the conditions required for strict compatibility which results in convergence block. By improving the approximation method of the kernel function gradient term, the convergence accuracy level can be further improved, and the convergence block phenomenon would also be eliminated.

References

[1] Gingold, R.A, Monaghan, J.J., Smoothed particle hydrodynamics: Theory and application to non-spherical stars, *Monthly Notices of the Royal Astronomical Society*, 181(3), 1977, 375-389.
[2] Lucy, L.B., A numerical approach to the testing of the fission hypothesis, *Astronomical Journal*, 82, 1977, 1013-1024.
[3] Oñate, E., Idelsohn, S., Zienkiewicz, O.C., et al. A finite point method in computational mechanics-application to convective transport and fluid flow International, *Journal for Numerical Methods in Engineering*, 39(22), 1996, 3839-3866.
[4] Oberkampf, W.L., Roy, C.J., *Verification and validation in science computing*, Cambridge University Press, 2010, 1-767.
[5] Roy, C.J., Oberkampf, W.L., A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing, *Computer Methods in Applied Mechanics and Engineering*, 200(25-28), 2011, 2131-2144.
[6] Monaghan, J.J., Gingold, R.A., Shock simulation by the particle method SPH, *Journal of Computational Physics*, 52(2), 1983, 374-389.
[7] Petschek, A.G., Libersky, L.D., Cylindrical smoothed particle hydrodynamics, *Journal of Computational Physics*, 109(1), 1993, 76-83.
[8] Liu, W.K., Jun, S., Li, S.F., et al. Reproducing kernel particle methods for structural dynamics, *International Journal for Numerical Methods in Engineering*, 38(10), 1995, 1655-1679.
[9] Liu, W.K., Jun, S., Sihling, D.T., et al. Multiresolution reproducing kernel particle methods for computational fluid dynamics, *International Journal for Numerical Methods in Fluids*, 24(12), 1997, 1391-1415.
[10] Chen, J.K., Beraun, J.E., Jih, C.K., An improvement for tensile instability in smoothed particle hydrodynamics, *Computational Mechanics*, 23(4), 1999, 279-287.
[11] Campbell, J., Vignjevic, R., Libersky, L., A contact algorithm for smoothed particle hydrodynamics, *Computer Methods in Applied Mechanics and Engineering*, 184(1), 2000, 49-65.
[12] Inutsuka, S., Reformulation of smoothed particle hydrodynamics with Riemann solver, *Journal of Computational Physics*, 179(1), 2002, 238-267.
[13] Zhang, G.M., Batra, R.C., Modified smoothed particle hydrodynamics method and its application to transient problems, *Computational Mechanics*, 34(2), 2004, 137-146.
[14] Rafiee, A., Thiagarajan, K.P., An SPH projection method for simulating fluid-hypoelastic structure interaction, *Computer Methods in Applied Mechanics and Engineering*, 198(33-36), 2009, 2785-2795.
[15] Xu, R., Stansby, P., Laurence, D., Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach, *Journal of Computational Physics*, 228(18), 2009, 6703-6725.
[16] Yin, J.W., Ma, Z.B., Reproducing kernel particle method in smoothed particle hydrodynamics, *Chinese Journal of Computational Physics*, 26(4), 2009, 553-558.
[17] Fedir, V.S., Jack, J.Y., A smoothed particle hydrodynamics method with approximate Riemann solvers for simulation of strong explosions, *Computers & Fluids*, 88(15), 2013, 418-429.
[18] Liu, W.K., Jun, S., Li, S.F., et al. Reproducing kernel particle methods, *International Journal for Numerical Methods in Fluids*, 20(8-9), 1995, 1081-1106.
[19] The American Society of Mechanical Engineers. V&V 20-2009, Standard for verification and validation in computational fluid dynamics and heat transfer, New York, 2009, 1-87.
[20] The American Society of Mechanical Engineers. V&V 10.1-2012, An illustration of the concepts of verification and validation in computational solid mechanics, New York, 2012, 1-36.
[21] Brookshaw, L., A method of calculating radiative heat diffusion in particle simulations, *Astronomical Society of Australia Proceedings*, 6(2), 1985, 207-210.
[22] Cleary, P.W, Monaghan, J.J., Conducting modelling using smoothed particle hydrodynamics, *Journal of Computational Physics*, 148(1), 1999, 227-264.
[23] Zhang, S., Morita, K., Fukuda, K., et al. An improved MPS method for numerical simulations of convective heat transfer problems, *International Journal for Numerical Methods in Fluids*, 51(1), 2006, 31-47.
[24] Basa, M., Quilan, N., Lastiwka, M., Robustness and accuracy of SPH formulations for viscous flow, *International Journal for Numerical Method in Fluids*, 60(10), 2009, 1127-1148.
[25] Ma, Z., Physics Evoked Cloud Method: A Versatile Systematic Method for Numerical Simulations, *Chinese Journal of Computational Physics*, 34(3), 2017, 261-272.