



A reduced-order variational multiscale interpolating element free Galerkin technique based on proper orthogonal decomposition for solving Navier–Stokes equations coupled with a heat transfer equation: Nonstationary incompressible Boussinesq equations



Mostafa Abbaszadeh^{a,*}, Mehdi Dehghan^a, Amirreza Khodadadian^{c,d},
Nima Noii^d, Clemens Heitzinger^{c,b}, Thomas Wick^d

^a Department of Applied Mathematics, Faculty of Mathematics and Computer Sciences, Amirkabir University of Technology, No. 424, Hafez Ave., 15914 Tehran, Iran

^b School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85287, USA

^c Institute for Analysis and Scientific Computing, Vienna University of Technology (TU Wien), Wiedner Hauptstraße 8–10, 1040 Vienna, Austria

^d Leibniz University of Hannover, Institute for Applied Mathematics, Welfengarten 1, 30167, Hannover, Germany

ARTICLE INFO

Article history:

Received 15 March 2020

Received in revised form 29 July 2020

Accepted 25 September 2020

Available online 12 October 2020

Keywords:

Interpolating moving least squares

approximation and Meshless methods

Element free Galerkin and proper

orthogonal decomposition methods

Rayleigh–Benard convection problem

Variational multiscale approach

Incompressible Navier–Stokes and

nonstationary incompressible Boussinesq equations

Large scale atmospheric and oceanic flows

ABSTRACT

In the recent decade, meshless methods have been handled for solving some PDEs due to their easiness. One of the most efficient meshless methods is the element free Galerkin (EFG) method. The test and trial functions of the EFG are based upon the special basis. Recently, some modifications have been developed to improve the EFG method. One of these improvements is the variational multiscale EFG (VMEFG) procedure. In the current article, the shape functions of interpolating moving least squares (IMLS) approximation are applied to the variational multiscale EFG technique to numerical study the Navier–Stokes equations coupled with a heat transfer equation such that this model is well-known as two-dimensional nonstationary Boussinesq equations. In order to reduce the computational time of simulation, we employ a reduced order model (ROM) based on the proper orthogonal decomposition (POD) technique. In the current paper, we developed a new reduced order model based on the meshless numerical procedure for solving an important model in fluid mechanics. To illustrate the reduction in CPU time as well as the efficiency of the proposed method, we investigate two-dimensional cases.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

This work devoted to coupling of the incompressible Navier–Stokes equations with a heat conduction problem. The resulting system is the so-called nonstationary Boussinesq approximation [1,2]. The Boussinesq equations can be used for

* Corresponding author.

E-mail addresses: m.abbaszadeh@aut.ac.ir (M. Abbaszadeh), mdehghan@aut.ac.ir (M. Dehghan), amirreza.khodadadian@tuwien.ac.at (A. Khodadadian), noii@ifam.uni-hannover.de (N. Noii), clemens.heitzinger@tuwien.ac.at (C. Heitzinger), thomas.wick@ifam.uni-hannover.de (T. Wick).

<https://doi.org/10.1016/j.jcp.2020.109875>

0021-9991/© 2020 Elsevier Inc. All rights reserved.

modeling large scale atmospheric and oceanic flows that are responsible for cold fronts and the jet stream. Furthermore, the Boussinesq equations have important role in the study of Rayleigh-Benard convection [3]. Thus, we consider

$$\mathbf{u}_t(\mathbf{x}, t) - \varepsilon \Delta \mathbf{u}(\mathbf{x}, t) + (\mathbf{u}(\mathbf{x}, t) \cdot \nabla) \mathbf{u}(\mathbf{x}, t) + \nabla p(\mathbf{x}, t) = w \mathbf{j}, \quad (\mathbf{x}, t) \in \Omega \times I, \quad (1.1)$$

$$\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0, \quad (\mathbf{x}, t) \in \Omega \times I, \quad (1.2)$$

$$w_t(\mathbf{x}, t) - \gamma \Delta w(\mathbf{x}, t) + (\mathbf{u}(\mathbf{x}, t) \cdot \nabla) w(\mathbf{x}, t) = 0, \quad (\mathbf{x}, t) \in \Omega \times I, \quad (1.3)$$

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t), \quad w(\mathbf{x}, t) = h(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \partial\Omega \times I, \quad (1.4)$$

$$\mathbf{u}(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad w(\mathbf{x}, 0) = w_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (1.5)$$

where $\mathbf{j} = (0, 1)$ is unit vector, Ω is the computational domain and $I = (0, T_f)$ such that T_f is the final time. The mathematical model (1.1)-(1.4) has been studied by some numerical techniques for example mixed finite element formulation [4], POD mixed finite volume element procedure [5], POD Galerkin type with error estimation [2] and Crank–Nicolson mixed finite volume–element procedure [1]. Also, the existence and uniqueness of the solutions of model (1.1)-(1.5) are studied in [8]. The extended version of Eqs. (1.1)-(1.5) is [4,5]

$$\frac{\partial u}{\partial t} - \varepsilon \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} = 0, \quad (\mathbf{x}, t) \in \Omega \times I, \quad (1.6)$$

$$\frac{\partial v}{\partial t} - \varepsilon \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} = w, \quad (\mathbf{x}, t) \in \Omega \times I, \quad (1.7)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (\mathbf{x}, t) \in \Omega \times I, \quad (1.8)$$

$$\frac{\partial w}{\partial t} - \gamma \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right) + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} = 0, \quad (\mathbf{x}, t) \in \Omega \times I, \quad (1.9)$$

$$u(\mathbf{x}, t) = f(\mathbf{x}, t), \quad v(\mathbf{x}, t) = g(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \partial\Omega \times I, \quad (1.10)$$

$$w(\mathbf{x}, t) = h(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \partial\Omega \times I, \quad (1.11)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad v(\mathbf{x}, 0) = v_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (1.12)$$

$$w(\mathbf{x}, 0) = w_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (1.13)$$

where

- \mathbf{x} is (x, y) ,
- u and v are the velocity components of the fluid in the x - and y -directions, respectively,
- w is the temperature of the fluid,
- p presents the pressure of the fluid,
- $\varepsilon = \sqrt{\text{Pr}(Re)^{-1}}$,
- Re denotes the Reynolds number,
- Pr interprets the Prandtl number,
- $\gamma = \sqrt{(Re)(Pr)}$,
- f, g and h are the boundary conditions for the velocity in the x - and y -directions and the temperature of the fluid, respectively,
- Furthermore, u_0, v_0 and w_0 are the initial conditions for the velocity in the x - and y -directions and the temperature of the fluid, respectively.

The proper orthogonal decomposition (POD) idea is a method used to construct reduced order models (ROMs) [9,10]. The POD technique can be found in several research papers for solving different physical models. The POD technique is considered by many scholars. The main aim of [11,12] is to evaluate and compare the efficiencies of techniques for constructing reduced-order models for finite difference (FD) and finite element (FE) algorithms obtained via discretizing the systems of unsteady nonlinear partial differential equations (PDEs). A new approach to enhance the accuracy of a novel Proper Orthogonal Decomposition (POD) model applied to moderate Reynolds number flows (of the type typically encountered in ocean models) is developed in [13]. The authors of [14] proposed a non-intrusive reduced order model for general, dynamic partial differential equations based on the proper orthogonal decomposition (POD) and Smolyak sparse grid collocation. Reduced-order models are derived in [15,16] from low-order bases computed by applying proper orthogonal decomposition (POD) on an a priori ensemble of data of the Navier–Stokes model. A non-intrusive model reduction computational method is developed in [17] using hypersurfaces representation for reservoir simulation and further it was applied to 3D fluvial channel problems. Recently, authors of [18] presented a non-intrusive reduced order model based on machine learning.

A non-intrusive reduced order method is employed in [19] to model a solid interacting with compressible fluid flows to simulate crack initiation and propagation. A new reduced order model is proposed in [20] based upon the POD for solving the Navier–Stokes equations as the novelty of the method lies in its treatment of the equation's non-linear operator. The main aim of [21] is to develop a new nonlinear POD Petrov–Galerkin approach for the Navier–Stokes equations. A new non-intrusive model reduction method is proposed in [22] for the Navier–Stokes equations based on the radial basis function (RBF) multi-dimensional interpolation instead of the traditional approach of projecting the equations onto the reduced space. A fast and stabilized meshless method that combines a variational multi-scale element free Galerkin (VMEFG) method and the POD method is developed in [23] to solve convection-diffusion problems. The POD technique is applied for the meshless method in [24] for transient heat conduction problems. A combination of POD method with finite difference technique has been proposed in [25,26] to solve the parabolized Navier–Stokes (PNS) equations. A POD technique is used in [5] for model reduction of mixed finite element (MFE) for the nonstationary Navier–Stokes equations and error estimates between a reference solution and the POD solution of reduced MFE formulation are studied. Authors of [6] proposed a framework for orthogonal decomposition of swirling flows applied to problems originating from turbomachines. A combination of proper orthogonal decomposition with radial basis functions is developed in [7] for solving fluid flow problems.

The interpolating moving least-squares (IMLS) method based on a nonsingular weight function is used in [27] to construct the approximation function, the weak form of the problem of inhomogeneous swelling of polymer gels is used to obtain the final discretized equations, and penalty method is applied to impose the displacement boundary condition, then an improved element-free Galerkin (IEFG) method for the problem of the inhomogeneous swelling of polymer gels is presented. The improved element-free Galerkin (IEFG) method based on the improved MLS approximation and a nonsingular weight function is proposed in [28] for solving elastoplastic large deformation problems. Improved complex variable moving least-squares (ICVMLS) approximation is applied in [29] to construct the shape function and then modified Galerkin weak form of wave propagation problems is employed for obtaining the final system equations. The improved element-free Galerkin (IEFG) method is presented in [30] based on the improved MLS approximation to solve three-dimensional elastoplasticity. The authors of [31] developed an interpolating element-free Galerkin (IEFG) method for solving three-dimensional potential problems based on the improved interpolating moving least-squares (IIMLS) method. By combining the dimension splitting method and the improved complex variable element-free Galerkin method, the dimension splitting and improved complex variable element-free Galerkin (DS-ICVEFG) method is developed in [32] for solving 3D transient heat conduction problems. Furthermore, authors of [33] combined the dimension splitting method with the improved complex variable element-free Galerkin method to get a hybrid improved complex variable element-free Galerkin (H-ICVEFG) method for solving three-dimensional advection-diffusion problems. The main aim of [34] is to develop a fast and efficient local meshless method based on the POD method and RBF-generated FD technique for solving shallow water equations in one- and two-dimensional cases. The authors of [35] employed the shape functions of the reproducing kernel particle method in the meshless local Petrov–Galerkin procedure for solving two-dimensional nonstationary incompressible Boussinesq equations. The main propose of [36] is to introduce a numerical procedure based on the POD method and local RBF-generated FD formulation to simulate the time dependent incompressible Navier–Stokes equation with variable density. The Oldroyd model as a generalized incompressible Navier–Stokes equation is investigated in [37] via the interpolating stabilized element free Galerkin technique. An upwind local radial basis functions-differential quadrature (RBFs-DQ) technique is developed in [38] to simulate some models arising in water sciences. The authors of [39] developed a meshless numerical procedure based on the interpolating element free Galerkin (IEFG) method to simulate the groundwater equation (GWE). The main aim of [40] is to propose a POD reduced-order discontinuous Galerkin method for solving the generalized Swift–Hohenberg equation with application in biological science and mechanical engineering.

In the current research work, we replace the MLS shape functions with the interpolating MLS shape functions to directly apply the essential conditions. Also, we employ a variational multiscale (VM) approach based on increasing the order of approximation to improve the numerical results. Furthermore, to decrease the computational cost of the new scheme, the POD technique is utilized.

The structure of this paper is: the shape functions of interpolating moving least squares approximation are explained in Section 2, the proper orthogonal decomposition is described in Section 3, the discretization of the temporal variable is developed in Section 4, the variational multiscale element free Galerkin method is explained in Section 5, some numeri-

cal experiments are investigated in Section 6 to show the efficiency and accuracy of the new numerical formulation and conclusion of paper has been noted in Section 7.

2. Shape functions of interpolating MLS approximation

Here, we explain constructing the shape functions of the interpolating MLS (IMLS) approximation. The shape functions of MLS approximation do not have δ -Kronecker property thus the Dirichlet boundary condition cannot be applied, directly. However, the shape functions of IMLS approximation are built based on a singular weight function as according to this alteration, the new shape functions have δ -Kronecker property.

Let $X = \{\zeta_i\}_{i=1}^N$ be a set of distributed nodes in $\Omega \subset \mathbb{R}^n$. The fill distance parameter is

$$h_{X,\Omega} = \sup_{x \in \Omega} \min_{1 \leq j \leq N} \|x - \zeta_j\|_2, \quad q_X = \frac{1}{2} \min_{i \neq j} \|\zeta_i - \zeta_j\|_2. \tag{2.1}$$

Also

$$\mathfrak{N}(\zeta) \triangleq B(\zeta, \delta) = \{\zeta^* \in \mathbb{R}^n : \|\zeta - \zeta^*\| < \delta(\zeta)\}, \tag{2.2}$$

is the influence domain of node ζ and the influence domain of point ζ_i is

$$\mathfrak{N}_i \triangleq \mathfrak{N}(\zeta_i) = \{\zeta^* \in \mathbb{R}^n : \|\zeta_i - \zeta^*\| < \delta_i\}, \tag{2.3}$$

where δ_i is the radius of \mathfrak{N}_i . Also, the following weight function is employed [41]

$$w_i(\mathbf{x}) = \begin{cases} \Phi\left(\frac{\|\mathbf{x} - \zeta_i\|_2}{\delta_i}\right) \left\|\frac{\mathbf{x} - \zeta_i}{\delta_i}\right\|_2^{-\alpha}, & \mathbf{x} \in \mathfrak{N}_i, \\ 0, & \mathbf{x} \notin \mathfrak{N}_i, \end{cases} \tag{2.4}$$

where the function Φ is nonnegative, compactly supported in the unit circle $B(0, 1)$, r -th times continuously differentiable, and its derivatives up to order r are bounded. The function Φ may be constant one or any weight function of MLS approximation. We set

$$\mathbf{p}(\mathbf{x}) = [p_0(\mathbf{x}), p_1(\mathbf{x}), \dots, p_{m-1}(\mathbf{x})]^T, \quad \mathbf{x} \in \Omega, \tag{2.5}$$

where m is the number of polynomials. The shifted and scaled bases are applied as [42]

$$\mathbf{p}(\mathbf{x}) = \left[1, \left(\frac{x - x^*}{h}\right), \left(\frac{x - x^*}{h}\right)^2\right]^T, \quad \text{in } 1D,$$

$$\mathbf{p}(\mathbf{x}) = \left[1, \left(\frac{x - x^*}{h}\right), \left(\frac{y - y^*}{h}\right), \left(\frac{x - x^*}{h}\right)^2, \left(\frac{x - x^*}{h}\right) \times \left(\frac{y - y^*}{h}\right), \left(\frac{y - y^*}{h}\right)^2\right]^T, \quad \text{in } 2D,$$

where (x^*, y^*) is a fixed point. Thus, we use $I_1, I_2, \dots, I_{\delta(\zeta)}$ to describe the sequential sequence numbers of these points and also

$$\mathcal{E}(\zeta) = [I_1, I_2, \dots, I_{\delta(\zeta)}]. \tag{2.6}$$

Consider $\text{span}\{p_0(\zeta), p_1(\zeta), \dots, p_m(\zeta)\}$ and according to [41], we have

$$q_0(\zeta, \bar{\zeta}) = \frac{p_0(\zeta)}{(p_0, p_0)_\zeta^{\frac{1}{2}}} = \frac{1}{\left[\sum_{i \in \mathcal{E}(\zeta)} w_i(\zeta)\right]^{\frac{1}{2}}}, \tag{2.7}$$

where the following inner product is defined

$$(f, g)_\zeta = \sum_{i \in \mathcal{E}(\zeta)} w_i(\zeta) f(\zeta_i) g(\zeta_i). \tag{2.8}$$

Also, we set [41]

$$q_i(\zeta, \bar{\zeta}) = p_i(\bar{\zeta}) - \sum_{l \in \mathcal{E}(\zeta)} v_l(\zeta) p_l(\zeta_l), \tag{2.9}$$

in which

$$v_l(\boldsymbol{\zeta}) = \frac{w_l(\boldsymbol{\zeta})}{\sum_{j \in \mathcal{E}(\boldsymbol{\zeta})} w_j(\boldsymbol{\zeta})}. \tag{2.10}$$

To approximate the unknown function $u(\boldsymbol{\zeta})$ at $\boldsymbol{\zeta}$, $u_h(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}})$ takes the following form [41]

$$u_h(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) = \sum_{i=0}^m q_i(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) a_i(\boldsymbol{\zeta}) = q_0(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) a_0(\boldsymbol{\zeta}) + \mathbf{q}^T(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) \mathbf{a}(\boldsymbol{\zeta}), \tag{2.11}$$

such that $\{a_i(\boldsymbol{\zeta})\}_{i=0}^m$ are the unknown coefficients. According to [41], to approximate a_i , the following minimization problem is defined through:

$$J(\boldsymbol{\zeta}) = \sum_{i \in \mathcal{E}(\boldsymbol{\zeta})} w_i(\boldsymbol{\zeta}) [u(\boldsymbol{\zeta}_i) - u_h(\boldsymbol{\zeta}, \boldsymbol{\zeta}_i)]^2 = \sum_{i \in \mathcal{E}(\boldsymbol{\zeta})} w_i(\boldsymbol{\zeta}) \left[u(\boldsymbol{\zeta}_i) - \sum_{i=0}^m q_i(\boldsymbol{\zeta}, \boldsymbol{\zeta}_i) a_i(\boldsymbol{\zeta}) \right]^2. \tag{2.12}$$

According to relation (2.8), Eq. (2.12) can be rewritten as follows

$$(u(\cdot) - u_h(\boldsymbol{\zeta}, \cdot), q_i(\boldsymbol{\zeta}, \cdot))_{\boldsymbol{\zeta}} = 0, \quad 0 \leq i \leq m, \tag{2.13}$$

such that [41]

$$a_0(\boldsymbol{\zeta}) = (u - q_0(\boldsymbol{\zeta}, \cdot))_{\boldsymbol{\zeta}}, \tag{2.14}$$

$$\sum_{i=1}^m (q_i(\boldsymbol{\zeta}, \cdot), q_j(\boldsymbol{\zeta}, \cdot))_{\boldsymbol{\zeta}} a_i(\boldsymbol{\zeta}) = (u, q_j(\boldsymbol{\zeta}, \cdot))_{\boldsymbol{\zeta}}, \quad j = 1, 2, \dots, m. \tag{2.15}$$

Thus, Eq. (2.15) can be written as

$$\mathbf{A}(\boldsymbol{\zeta}) \mathbf{a}(\boldsymbol{\zeta}) = \mathbf{B}(\boldsymbol{\zeta}) \mathbf{u}, \tag{2.16}$$

where

$$\mathbf{u} = \left[u(\boldsymbol{\zeta}_{l_1}) \quad u(\boldsymbol{\zeta}_{l_2}) \quad \dots \quad u(\boldsymbol{\zeta}_{l_{\eta(\boldsymbol{\zeta})}}) \right]^T, \quad \mathbf{A}(\boldsymbol{\zeta}) = \mathbf{B}(\boldsymbol{\zeta}) \mathbf{Q}(\boldsymbol{\zeta}), \tag{2.17}$$

$$\mathbf{Q}(\boldsymbol{\zeta}) = \left[\mathbf{q}(\boldsymbol{\zeta}, \boldsymbol{\zeta}_{l_1}) \quad \mathbf{q}(\boldsymbol{\zeta}, \boldsymbol{\zeta}_{l_2}) \quad \dots \quad \mathbf{q}(\boldsymbol{\zeta}, \boldsymbol{\zeta}_{l_{\eta(\boldsymbol{\zeta})}}) \right], \tag{2.18}$$

and also [41]

$$\mathbf{B}_{ij}(\boldsymbol{\zeta}) = \begin{cases} w_{l_j}(\boldsymbol{\zeta}) q_i(\boldsymbol{\zeta}, \boldsymbol{\zeta}_{l_j}), & \boldsymbol{\zeta} \neq \boldsymbol{\zeta}_{l_j}, \\ \sum_{k \in \mathcal{E}(\boldsymbol{\zeta}), k \neq j} w_k(\boldsymbol{\zeta}) [p_i(\boldsymbol{\zeta}_{l_j}) - p_i(\boldsymbol{\zeta}_k)], & \boldsymbol{\zeta} = \boldsymbol{\zeta}_{l_j}. \end{cases} \tag{2.19}$$

Hence, by using (2.16), we approximate \mathbf{a} , as follows

$$\mathbf{a}(\boldsymbol{\zeta}) = \mathbf{A}^{-1}(\boldsymbol{\zeta}) \mathbf{B}(\boldsymbol{\zeta}) \mathbf{u}. \tag{2.20}$$

Now, from the above formulation, we can achieve [41]

$$q_0(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) a_0(\boldsymbol{\zeta}) = q_0(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) (u, q_0(\boldsymbol{\zeta}, \cdot))_{\boldsymbol{\zeta}} = \sum_{i \in \mathcal{E}(\boldsymbol{\zeta})} v_i(\boldsymbol{\zeta}) u(\boldsymbol{\zeta}_i) = \boldsymbol{\beta}^T(\boldsymbol{\zeta}) \mathbf{u}, \tag{2.21}$$

in which

$$\boldsymbol{\beta}(\boldsymbol{\zeta}) = \left[v_{l_1}(\boldsymbol{\zeta}) \quad v_{l_2}(\boldsymbol{\zeta}) \quad \dots \quad v_{l_{\eta(\boldsymbol{\zeta})}}(\boldsymbol{\zeta}) \right]^T. \tag{2.22}$$

Applying Eqs. (2.20) and (2.21) into Eq. (2.11) yields

$$u_h(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) = \boldsymbol{\beta}^T(\boldsymbol{\zeta}) \mathbf{u} + \mathbf{q}^T(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) \mathbf{A}^{-1}(\boldsymbol{\zeta}) \mathbf{B}(\boldsymbol{\zeta}) \mathbf{u}. \tag{2.23}$$

Thus, we have [41]

$$u(\boldsymbol{\zeta}) \approx u_h(\boldsymbol{\zeta}) = u_h(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}})|_{\bar{\boldsymbol{\zeta}}=\boldsymbol{\zeta}} = \left[\boldsymbol{\beta}^T(\boldsymbol{\zeta}) + \mathbf{q}^T(\boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}) \mathbf{A}^{-1}(\boldsymbol{\zeta}) \mathbf{B}(\boldsymbol{\zeta}) \right] \mathbf{u}, \tag{2.24}$$

where the IMLS shape functions are [41]

$$\phi_i(\boldsymbol{\zeta}) = \begin{cases} v_i(\boldsymbol{\zeta}) + \sum_{j=1}^m q_j(\boldsymbol{\zeta}, \boldsymbol{\zeta})[\mathbf{A}^{-1}(\boldsymbol{\zeta})\mathbf{B}(\boldsymbol{\zeta})]_{jk}, & i = I_k \in \mathcal{E}(\boldsymbol{\zeta}), \\ 0, & i \notin \mathcal{E}(\boldsymbol{\zeta}). \end{cases} \quad (2.25)$$

Remark 2.1. As mentioned previously, in this technique, we can apply the Dirichlet boundary condition exactly i.e. this boundary condition is imposed by some changes in the final coefficient matrix and also in the right hand side vector of the final algebraic system of equations. Consider the following general problem

$$\begin{cases} \mathcal{L}u = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega \end{cases} \quad (2.26)$$

in which \mathcal{L} is linear differential operator and also functions f and g are known. After constructing the discretization of main equations, based on the interior and boundary integrals, we get a system of algebraic equations as follows

$$\mathbf{AU} = \mathbf{F}. \quad (2.27)$$

Now, to apply the Dirichlet boundary condition the following steps must be done:

1. Find rows of matrix \mathbf{A} associated to the boundary nodes for example i th-row of matrix \mathbf{A} is related to the i th-node on $\partial\Omega$. Set

$$\begin{aligned} A(i, \cdot) &= 0, \quad \text{i.e. } A(i, j) = 0 \text{ for every } j = 1, 2, \dots, \dim(\mathbf{A}), \\ A(i, i) &= 1. \end{aligned}$$

2. Put the value of boundary condition in i th-element of right hand side vector F i.e.

$$F(i) = g(\mathbf{x}_i).$$

According to the above steps, the Dirichlet boundary conditions can be applied, exactly and also without interpolation error.

3. Construction of the POD basis

Let $p_r^{n_i}$ be known for $1 \leq n_1 < n_2 < \dots < n_L \leq N$ and $1 \leq i \leq L$, we define

$$\mathcal{V} = \text{span} \{p_r^{n_1}, p_r^{n_2}, \dots, p_r^{n_L}\}, \quad (3.1)$$

and also $\{\Phi\}_{j=1}^l$ is an orthogonal basis of \mathcal{V} such that

$$p_r^{n_i} = \sum_{j=1}^l (p_r^{n_i}, \Phi_j)_\omega \Phi_j, \quad i = 1, 2, \dots, L, \quad (3.2)$$

in which

$$(p_r^{n_i}, \Phi_j)_\omega = (\nabla p_r^{n_i}, \nabla \Phi_j). \quad (3.3)$$

Definition 3.1. [4,5] In the POD idea, we want to find an orthogonal basis Φ_j such that for every $1 \leq d \leq l$

$$\begin{cases} \min_{\{\Phi\}_{j=1}^d} \frac{1}{L} \sum_{i=1}^L \left\| p_r^{n_i} - \sum_{j=1}^d (p_r^{n_i}, \Phi_j)_\omega \Phi_j \right\|_\omega^2 \\ \text{subject to: } (\Phi_i, \Phi_j)_\omega = \delta_{ij}, \quad 1 \leq i \leq d, \quad 1 \leq j \leq d, \end{cases} \quad (3.4)$$

where

$$\|p_r^{n_i}\|_\omega^2 = \|\nabla p_r^{n_i}\|^2. \quad (3.5)$$

On other hand, problem (3.4) is equivalent to [4,5]

$$\max_{(\Phi, \Phi)_\omega = \|\nabla\Phi\|_{L^2(\Omega)}^2} \frac{1}{L} \sum_{i=1}^L |(p_r^{n_i}, \Phi)_\omega|^2. \tag{3.6}$$

We use trial function Φ as follows [4]

$$\Phi = \sum_{i=1}^L \beta_i p_r^{n_i}, \tag{3.7}$$

where the coefficient β_i must be calculated such that Φ is a maximizer for Eq. (3.6) [4]. Now, we define

$$F((x, y), (\eta, \xi)) = \frac{1}{L} \sum_{i=1}^L p_r^{n_i}(x, y) p_r^{n_i}(\eta, \xi), \tag{3.8}$$

and

$$I\Phi = \int_{\Omega} \nabla' F((x, y), (\eta, \xi)) \nabla' \Phi(\eta, \xi) d\eta d\xi, \tag{3.9}$$

where $I : \omega \rightarrow \omega$ and ∇' is the gradient with respect to (η, ξ) . Thus, [4]

$$(I\Phi, \Phi)_\omega = \frac{1}{L} \sum_{i=1}^L |(p_r^{n_i}, \Phi)_\omega|^2, \tag{3.10}$$

$$(I\Phi, \phi)_\omega = (\Phi, I\phi)_\omega, \quad \forall \Phi, \phi \in \omega. \tag{3.11}$$

According to the above explanations, problem (3.6) reduces to find the largest eigenvalue of

$$I\Phi = \lambda \nabla\Phi, \quad \text{and} \quad \|\nabla\Phi\|_{L^2(\Omega)} = 1, \tag{3.12}$$

or equivalently

$$\int_{\Omega} \nabla' \nabla F((x, y), (\eta, \xi)) \nabla' \Phi(\eta, \xi) d\eta d\xi = \lambda \nabla\Phi, \quad \text{and} \quad \|\nabla\Phi\|_{L^2(\Omega)} = 1. \tag{3.13}$$

Employing function F and Eq. (3.7) in relation (3.13), gives [4]

$$\sum_{k=1}^L \left(\frac{1}{L} \int_{\Omega} \nabla' p_r^{n_i}(\eta, \xi) \cdot \nabla' p_r^{n_k}(\eta, \xi) d\eta d\xi \right) \beta_k = \lambda \beta_i, \quad i = 1, 2, \dots, L. \tag{3.14}$$

Therefore, Eq. (3.13) is converted to the eigenvalue problem

$$\bar{\mathbf{A}}\bar{\boldsymbol{\beta}} = \lambda\bar{\boldsymbol{\beta}}, \tag{3.15}$$

in which

$$\bar{\mathbf{A}}_{ik} = \int_{\Omega} \nabla' p_r^{n_i}(\eta, \xi) \cdot \nabla' p_r^{n_k}(\eta, \xi) d\eta d\xi, \quad \bar{\boldsymbol{\beta}} = [\beta_1, \beta_2, \dots, \beta_L]^T. \tag{3.16}$$

Moreover, $\bar{\mathbf{A}}$ has a complete set of orthogonal eigenvectors

$$\boldsymbol{\beta}^1 = [\beta_1^1, \beta_2^1, \dots, \beta_L^1]^T, \quad \boldsymbol{\beta}^2 = [\beta_1^2, \beta_2^2, \dots, \beta_L^2]^T, \dots, \boldsymbol{\beta}^l = [\beta_1^l, \beta_2^l, \dots, \beta_L^l]^T, \tag{3.17}$$

with the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_l > 0$. The solution of problem (3.4) is [4]

$$\Phi_1 = \frac{1}{\sqrt{L\lambda_1}} \sum_{i=1}^L \beta_i^1 p_r^{n_i}. \tag{3.18}$$

The remaining POD basis is

$$\Phi_k = \frac{1}{\sqrt{L\lambda_k}} \sum_{i=1}^L \beta_i^k p_r^{n_i}, \quad k = 2, 3, \dots, l. \tag{3.19}$$

The obtained POD basis has the following property [4]

$$(\Phi_k, \Phi_i)_\omega = \frac{1}{\sqrt{\lambda_k \lambda_i}} \lambda_i \beta^k \beta^i = \begin{cases} 1, & k = i, \\ 0, & k \neq i. \end{cases} \tag{3.20}$$

Theorem 3.2. [4,5] Let (λ_i, β^i) be eigenvalue and eigenvector of matrix $\bar{\mathbf{A}}$ such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_l > 0$. Then, the POD basis can be computed as

$$\Phi_i = \frac{1}{\sqrt{L\lambda_i}} \sum_{j=1}^L \beta_j^i p_r^{n_j}, \quad 1 \leq i \leq d \leq l. \tag{3.21}$$

Also, the following error formula holds

$$\frac{1}{L} \sum_{i=1}^L \left\| p_r^{n_i} - \sum_{j=1}^d (p_r^{n_i}, \Phi_j)_\omega \Phi_j \right\|_\omega^2 = \sum_{j=d+1}^l \lambda_j. \tag{3.22}$$

As shown in [43], we can use the following relation to obtain the number of POD basis [43]

$$I(m) = \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^l \lambda_i}. \tag{3.23}$$

4. Discretization of the temporal variable

We define $t_k = kdt$ for $k = 0, 1, \dots, N$, where $dt = T/N$. To approximate the time-derivative the backward finite difference method is used, thus we have

$$\frac{\partial u(x, y, t_k)}{\partial t} = \frac{u^{k+1}(x, y) - u^k(x, y)}{dt}, \tag{4.1}$$

$$\frac{\partial v(x, y, t_k)}{\partial t} = \frac{v^{k+1}(x, y) - v^k(x, y)}{dt}. \tag{4.2}$$

According to the main problem (1.6)-(1.9), we can write the following time discretization

$$\frac{u^{k+1} - u^k}{dt} - \varepsilon \left(\frac{\partial^2 u^k}{\partial x^2} + \frac{\partial^2 u^k}{\partial y^2} \right) + u^k \frac{\partial u^k}{\partial x} + v^k \frac{\partial u^k}{\partial y} + \frac{\partial p^k}{\partial x} = 0, \tag{4.3}$$

$$\frac{v^{k+1} - v^k}{dt} - \varepsilon \left(\frac{\partial^2 v^k}{\partial x^2} + \frac{\partial^2 v^k}{\partial y^2} \right) + u^k \frac{\partial v^k}{\partial x} + v^k \frac{\partial v^k}{\partial y} + \frac{\partial p^k}{\partial y} = w^k, \tag{4.4}$$

$$\frac{\partial u^{k+1}}{\partial x} + \frac{\partial v^{k+1}}{\partial y} = 0, \tag{4.5}$$

$$\frac{w^{k+1} - w^k}{dt} - \gamma \left(\frac{\partial^2 w^k}{\partial x^2} + \frac{\partial^2 w^k}{\partial y^2} \right) + u^k \frac{\partial w^k}{\partial x} + v^k \frac{\partial w^k}{\partial y} = 0. \tag{4.6}$$

Simplifying relations (4.3)-(4.6), yields

$$u^{k+1} = u^k + dt \left[\varepsilon \left(\frac{\partial^2 u^k}{\partial x^2} + \frac{\partial^2 u^k}{\partial y^2} \right) - u^k \frac{\partial u^k}{\partial x} - v^k \frac{\partial u^k}{\partial y} - \frac{\partial p^k}{\partial x} \right], \tag{4.7}$$

$$v^{k+1} = v^k + dt \left[\varepsilon \left(\frac{\partial^2 v^k}{\partial x^2} + \frac{\partial^2 v^k}{\partial y^2} \right) - u^k \frac{\partial v^k}{\partial x} - v^k \frac{\partial v^k}{\partial y} - \frac{\partial p^k}{\partial y} \right] + dt w^k, \tag{4.8}$$

$$\frac{\partial u^{k+1}}{\partial x} + \frac{\partial v^{k+1}}{\partial y} = 0, \tag{4.9}$$

$$w^{k+1} = w^k + dt \left[\gamma \left(\frac{\partial^2 w^k}{\partial x^2} + \frac{\partial^2 w^k}{\partial y^2} \right) - u^k \frac{\partial w^k}{\partial x} - v^k \frac{\partial w^k}{\partial y} \right]. \tag{4.10}$$

Now, we define

$$\mathbb{M}^k = u^k + dt \left[\varepsilon \left(\frac{\partial^2 u^k}{\partial x^2} + \frac{\partial^2 u^k}{\partial y^2} \right) - u^k \frac{\partial u^k}{\partial x} - v^k \frac{\partial u^k}{\partial y} \right], \tag{4.11}$$

$$\mathbb{N}^k = v^k + dt \left[\varepsilon \left(\frac{\partial^2 v^k}{\partial x^2} + \frac{\partial^2 v^k}{\partial y^2} \right) - u^k \frac{\partial v^k}{\partial x} - v^k \frac{\partial v^k}{\partial y} + w^k \right], \tag{4.12}$$

$$\mathbb{P}^k = w^k + dt \left[\gamma \left(\frac{\partial^2 w^k}{\partial x^2} + \frac{\partial^2 w^k}{\partial y^2} \right) - u^k \frac{\partial w^k}{\partial x} - v^k \frac{\partial w^k}{\partial y} \right], \tag{4.13}$$

then, Eqs. (4.6), (4.8) and (4.10) can be rewritten as follows

$$u^{k+1} = \mathbb{M}^k - dt \frac{\partial p^k}{\partial x}, \tag{4.14}$$

$$v^{k+1} = \mathbb{N}^k - dt \frac{\partial p^k}{\partial y}, \tag{4.15}$$

$$w^{k+1} = \mathbb{P}^k. \tag{4.16}$$

Next, the first-order derivative with respect to the x and Y for Eqs. (4.14) and (4.15), gives

$$\frac{\partial u^{k+1}}{\partial x} = \frac{\partial \mathbb{M}^k}{\partial x} - dt \frac{\partial^2 p^k}{\partial x^2}, \tag{4.17}$$

$$\frac{\partial v^{k+1}}{\partial y} = \frac{\partial \mathbb{N}^k}{\partial y} - dt \frac{\partial^2 p^k}{\partial y^2}, \tag{4.18}$$

respectively. Substituting Eqs. (4.17) and (4.18) in Eq. (4.9), results in

$$\frac{\partial^2 p^k}{\partial x^2} + \frac{\partial^2 p^k}{\partial y^2} = \frac{1}{dt} \left(\frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right), \tag{4.19}$$

such that Eq. (4.19) is a Poisson equation. We note that Poisson problem (4.19) has been solved by boundary element method [44,45] and element free Galerkin method [46,47].

4.1. Numerical procedure in the temporal direction

According to the above explanations, we present the following steps:

Step 1: Calculate the velocity field and the temperature as follows

$$\mathbb{M}^k := u^k + dt \left[\varepsilon \left(\frac{\partial^2 u^k}{\partial x^2} + \frac{\partial^2 u^k}{\partial y^2} \right) - u^k \frac{\partial u^k}{\partial x} - v^k \frac{\partial u^k}{\partial y} \right], \tag{4.20}$$

$$\mathbb{N}^k := v^k + dt \left[\varepsilon \left(\frac{\partial^2 v^k}{\partial x^2} + \frac{\partial^2 v^k}{\partial y^2} \right) - u^k \frac{\partial v^k}{\partial x} - v^k \frac{\partial v^k}{\partial y} + w^k \right], \tag{4.21}$$

$$\mathbb{P}^k := w^k + dt \left[\gamma \left(\frac{\partial^2 w^k}{\partial x^2} + \frac{\partial^2 w^k}{\partial y^2} \right) - u^k \frac{\partial w^k}{\partial x} - v^k \frac{\partial w^k}{\partial y} \right]. \tag{4.22}$$

Step 2: Compute the pressure by using the following equation

$$\begin{cases} \frac{\partial^2 p^k}{\partial x^2} + \frac{\partial^2 p^k}{\partial y^2} = \frac{1}{dt} \left(\frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right), & \text{in } \Omega, \\ p^k|_{\Gamma_D} = p_0^k, \quad \nabla p^k \cdot \mathbf{n}|_{\Gamma_N} = q_0^k. & \text{on } \partial\Omega, \end{cases} \tag{4.23}$$

Step 3: Upgrade the velocity field and the temperature term by using the below relation

$$u^{k+1} = \mathbb{M}^k - dt \frac{\partial p^k}{\partial x}, \tag{4.24}$$

$$v^{k+1} = \mathbb{N}^k - dt \frac{\partial p^k}{\partial y}, \tag{4.25}$$

$$w^{k+1} = \mathbb{P}^k. \tag{4.26}$$

5. Variational multiscale interpolating EFG procedure

In the mid 90's Hughes [48,49] reviewed the stabilization schemes for the two-scale problems which are commonly known as the variational multiscale (VM) method. There are several research papers that the VM idea is combined with finite element method such as multiscale/stabilized (FEM) formulations for solving the incompressible Navier–Stokes equations [50], the advection-diffusion equation [51], the heat transfer problem [52], the Darcy flow model [53], the Fokker–Planck equation [54]. Also, Franca et al. [55,56] developed a two-level FEM for solving the convection–diffusion problem in [55] and the incompressible Navier–Stokes equations [56].

In these investigations, there exists a main assumption that the fine scale solutions vanish identically over the element boundaries although non-zero within the elements. Hughes [48] remarked that it is a rather strong assumption and it may be not valid for many cases of practical interest. Zhang et al. [57–59] followed the variational multiscale FEM and they applied it for meshfree methods and proposed the variational multiscale EFG technique for solving several practice problems in mechanics and electromagnetic applications. Recently, the non-intrusive version of the VM method which is so-called the global-local approach, is applied to the localized solid mechanics PDE; but not yet for the EFG setting [60–62]. Furthermore, Yeon and his co-authors [63,64] combined VM and meshless methods for studying elastoplastic solids.

5.1. The VMEFG method for Burgers' equation

In this section, we explain the variational multiscale element free Galerkin method for the Burgers' equation in the one-dimensional case. The used time-discrete scheme is as follows

$$\frac{u^n - u^{n-1}}{dt} + \mu u^{n-1} \frac{\partial u^{n-1}}{\partial x} = \frac{\varepsilon}{2} \left(\frac{\partial^2 u^n}{\partial x^2} + \frac{\partial^2 u^{n-1}}{\partial x^2} \right), \quad x \in [a, b], \quad n = 1, 2, \dots \tag{5.1}$$

Simplifying Eq. (5.1), gives

$$u^n - \frac{dt\varepsilon}{2} \frac{\partial^2 u^n}{\partial x^2} = u^{n-1} + \frac{dt\varepsilon}{2} \frac{\partial^2 u^{n-1}}{\partial x^2} - dt\mu u^{n-1} \frac{\partial u^{n-1}}{\partial x}, \quad x \in [a, b], \quad n = 1, 2, \dots \tag{5.2}$$

Let the unknown scalar solution be decomposed as follows

$$u = \tilde{u} + \bar{u}, \tag{5.3}$$

where \tilde{u} and \bar{u} are the coarse and fine scales terms, respectively. The trial function spaces for each scale are

$$\tilde{U} = \{u \mid u \in H^1(\Omega), \quad u = g \text{ at } \partial\Omega\}, \quad \bar{U} = \{u \mid u \in H^1(\Omega), \quad u = 0 \text{ at } \partial\Omega\}, \tag{5.4}$$

and $\tilde{u} \in \tilde{U}, \bar{u} \in \bar{U}$ such that $U = \tilde{U} \oplus \bar{U}$. Similar to the trial function and trial function space, for test function we have

$$v = \tilde{v} + \bar{v}, \tag{5.5}$$

in which \tilde{v} and \bar{v} are the coarse and fine scale terms, respectively and also

$$\tilde{V} = \{v \mid v \in H^1(\Omega), \quad v = 0 \text{ at } \partial\Omega\}, \quad \bar{V} = \{u \mid v \in H^1(\Omega), \quad v = 0 \text{ at } \partial\Omega\}, \tag{5.6}$$

with $V = \tilde{V} \oplus \bar{V}$. Considering Eq. (5.2), linearizing and substituting the trial solutions Eq. (5.3) and the weighting functions Eq. (5.5) into the standard variational form, we arrive at

$$\begin{aligned} \left\langle \tilde{v} + \bar{v}, \frac{\tilde{u}^{n+1} + \bar{u}^{n+1}}{dt} \right\rangle + \mu \left\langle \tilde{v} + \bar{v}, (\tilde{u}^n + \bar{u}^n) \frac{\partial (\tilde{u}^{n+1} + \bar{u}^{n+1})}{\partial x} \right\rangle \\ + \varepsilon \left\langle \frac{\partial (\tilde{v} + \bar{v})}{\partial x}, \frac{\partial (\tilde{u}^{n+1} + \bar{u}^{n+1})}{\partial x} \right\rangle = \left\langle \tilde{v} + \bar{v}, \frac{u^n}{dt} \right\rangle, \end{aligned} \tag{5.7}$$

in which (\bullet, \bullet) is the inner product. Using the linearity of the weighting function slot, we can split Eq. (5.7) into the coarse and the fine scale equations as follows

$$\tilde{V} : \left\langle \tilde{v}, \frac{\tilde{u}^{n+1} + \bar{u}^{n+1}}{dt} \right\rangle + \mu \left\langle \tilde{v}, (\tilde{u}^n + \bar{u}^n) \frac{\partial (\tilde{u}^{n+1} + \bar{u}^{n+1})}{\partial x} \right\rangle + \varepsilon \left\langle \frac{\partial \tilde{v}}{\partial x}, \frac{\partial (\tilde{u}^{n+1} + \bar{u}^{n+1})}{\partial x} \right\rangle = \left\langle \tilde{v}, \frac{u^n}{dt} \right\rangle, \tag{5.8}$$

$$\bar{V} : \left\langle \bar{v}, \frac{\tilde{u}^{n+1} + \bar{u}^{n+1}}{dt} \right\rangle + \mu \left\langle \bar{v}, (\tilde{u}^n + \bar{u}^n) \frac{\partial (\tilde{u}^{n+1} + \bar{u}^{n+1})}{\partial x} \right\rangle + \varepsilon \left\langle \frac{\partial \bar{v}}{\partial x}, \frac{\partial (\tilde{u}^{n+1} + \bar{u}^{n+1})}{\partial x} \right\rangle = \left\langle \bar{v}, \frac{u^n}{dt} \right\rangle. \tag{5.9}$$

Rewriting relations (5.8) and (5.9) yields

$$\begin{aligned} \tilde{V} : \langle \tilde{v}, \tilde{u}^{n+1} \rangle + dt \mu \left\langle \tilde{v}, u^n \frac{\partial \tilde{u}^{n+1}}{\partial x} \right\rangle + dt \varepsilon \left\langle \frac{\partial \tilde{v}}{\partial x}, \frac{\partial \tilde{u}^{n+1}}{\partial x} \right\rangle = dt \langle \tilde{v}, u^n \rangle - \langle \tilde{v}, \bar{u}^{n+1} \rangle \\ - dt \mu \left\langle \tilde{v}, u^n \frac{\partial \bar{u}^{n+1}}{\partial x} \right\rangle - dt \varepsilon \left\langle \frac{\partial \tilde{v}}{\partial x}, \frac{\partial \bar{u}^{n+1}}{\partial x} \right\rangle, \end{aligned} \tag{5.10}$$

$$\begin{aligned} \bar{V} : \langle \bar{v}, \bar{u}^{n+1} \rangle + dt \mu \left\langle \bar{v}, u^n \frac{\partial \bar{u}^{n+1}}{\partial x} \right\rangle + dt \varepsilon \left\langle \frac{\partial \bar{v}}{\partial x}, \frac{\partial \bar{u}^{n+1}}{\partial x} \right\rangle = dt \langle \bar{v}, u^n \rangle - \langle \bar{v}, \tilde{u}^{n+1} \rangle \\ - dt \mu \left\langle \bar{v}, u^n \frac{\partial \tilde{u}^{n+1}}{\partial x} \right\rangle - dt \varepsilon \left\langle \frac{\partial \bar{v}}{\partial x}, \frac{\partial \tilde{u}^{n+1}}{\partial x} \right\rangle. \end{aligned} \tag{5.11}$$

Now, we consider a set of functions whose sum equals to the unity on the whole domain such that it is covered with a set of open domains Ω_i i.e.

$$\text{supp}\{\phi_i\} = \Omega_i, \quad \forall x \in \Omega, \quad \sum_i \phi_i = 1. \tag{5.12}$$

The space of local enrichment basis functions is defined as follows

$$\vartheta_i(\Omega_i) = \text{span} \left\{ V_i^j \right\}. \tag{5.13}$$

Multiplying the partition of unity (PU) functions and the local approximation functions, the used space of functions for the approximation is as follows

$$\vartheta(\Omega) = \text{span} \left\{ \phi_i V_i^j \right\}. \tag{5.14}$$

Then the approximation of the unknown scalar field at point \mathbf{x} is applied by

$$u^h(\mathbf{x}) = \sum_i \sum_{V_i^j \in \vartheta_i} \phi_i V_i^j(\mathbf{x}) u_{i,j}. \tag{5.15}$$

Often for an open domain Ω_i , the polynomial basis functions are selected as the local enrichment basis V_i^j . In the sequel, we introduce some of them.

1. **First order** ($p = 1$):

- $\left\{ V_i^j \right\} = \left\{ V_i^1 \right\} = \{1\}$, 1D,
- $\left\{ V_i^j \right\} = \left\{ V_i^1 \right\} = \{1\}$, 2D.

2. **Second order** ($p = 2$):

- $\left\{ V_i^j \right\} = \left\{ V_i^1, V_i^2 \right\} = \{1, (x - x_i)^2\}$, 1D,
- $\left\{ V_i^j \right\} = \left\{ V_i^1, V_i^2, V_i^3 \right\} = \{1, (x - x_i)^2, (y - y_i)^2\}$, 2D.

3. **Third order** ($p = 3$):

- $\left\{ V_i^j \right\} = \left\{ V_i^1, V_i^2, V_i^3 \right\} = \{1, (x - x_i)^2, (x - x_i)^3\}$, 1D,
- $\left\{ V_i^j \right\} = \left\{ V_i^1, V_i^2, V_i^3, V_i^4, V_i^5, V_i^6, V_i^7 \right\} = \{1, (x - x_i)^2, (y - y_i)^2, (x - x_i)^3, (x - x_i)^2(y - y_i), (x - x_i)(y - y_i)^2, (y - y_i)^3\}$, 2D.

4. **Fourth order** ($p = 4$):

- $\left\{ V_i^j \right\} = \left\{ V_i^1, V_i^2, V_i^3 \right\} = \{1, (x - x_i)^2, (x - x_i)^3, (x - x_i)^4\}$, 1D,
- $\left\{ V_i^j \right\} = \left\{ V_i^1, V_i^2, V_i^3, V_i^4, V_i^5, V_i^6, V_i^7, V_i^8, V_i^9, V_i^{10}, V_i^{11}, V_i^{12} \right\} = \{1, (x - x_i)^2, (y - y_i)^2, (x - x_i)^3, (x - x_i)^2(y - y_i), (x - x_i)(y - y_i)^2, (y - y_i)^3, (x - x_i)^4, (x - x_i)^3(y - y_i), (x - x_i)^2(y - y_i)^2, (x - x_i)(y - y_i)^3, (y - y_i)^4\}$, 2D.

In the current paper, we consider fourth-order polynomial basis functions in one-dimension, thus we have

$$u^h(x) = \sum_i \phi_i \left(u_{i,0} + (x - x_i)^2 u_{i,1} + (x - x_i)^3 u_{i,2} + (x - x_i)^4 u_{i,3} \right), \tag{5.16}$$

where ϕ_i is shape functions of IMLS approximation. Rewriting relation (5.16) arrives

$$u^h(x) = \sum_i \phi_i u_{i,0} + \sum_i \phi_i (x - x_i)^2 u_{i,1} + \sum_i \phi_i (x - x_i)^3 u_{i,2} + \sum_i \phi_i (x - x_i)^4 u_{i,3}. \tag{5.17}$$

We distribute the coarse and fine scales as follows

$$\tilde{u}^h(x) = \sum_i \phi_i u_{i,0}, \tag{5.18}$$

$$\bar{u}^h(x) = \sum_i \phi_i (x - x_i)^2 u_{i,1} + \sum_i \phi_i (x - x_i)^3 u_{i,2} + \sum_i \phi_i (x - x_i)^4 u_{i,3}. \tag{5.19}$$

Now, substituting relations (5.18) and (5.19) in relations (5.10) and (5.11), respectively, we can write

$$\mathbf{M}_1 \tilde{u}^{n+1} = \mathbf{F}_1 + \mathbf{G}_1 (\bar{u}^{n+1}), \tag{5.20}$$

$$\mathbf{M}_2 \bar{u}^{n+1} = \mathbf{F}_2 + \mathbf{G}_2 (\tilde{u}^{n+1}), \tag{5.21}$$

in which \mathbf{M}_i for $i = 1, 2$ are coefficient matrices. For obtaining the acceptable results, the coarse scale problem (5.20) and the fine scale problem Eq. (5.21) must be solved iteratively. The following procedure has been employed:

(1) **Set $\tilde{\mathbf{u}}^{n+1,0} = \mathbf{u}^0$;**
 (2) **Solve the fine scale problem through the known the coarse scale solution $\tilde{\mathbf{u}}^{n+1}$ in the right-hand side to determine $\bar{\mathbf{u}}^{n+1,i+1}$**

$$\mathbf{M}_2^{n+1,i+1} \bar{\mathbf{u}}^{n+1,i+1} = \mathbf{F}_2^n + \mathbf{G}_2(\tilde{\mathbf{u}}^{n+1,i}).$$

 (3) **Solve the coarse scale problem to determine $\tilde{\mathbf{u}}^{n+1,i+1}$;**

$$\mathbf{M}_1^{n+1,i+1} \tilde{\mathbf{u}}^{n+1,i+1} = \mathbf{F}_1^n + \mathbf{G}_1(\bar{\mathbf{u}}^{n+1,i}).$$

 (4) **Compute $\mathbf{u}^{n+1,i+1}$ and $\mathbf{u}^{n+1,i}$ and calculate**

$$\text{Error} = \text{norm}(\mathbf{u}^{n+1,i+1} - \mathbf{u}^{n+1,i}, \text{inf}),$$

 (5) **Check if Error $\leq 10^{-5}$ then $\mathbf{u}^{n+1,i+1} \rightarrow \mathbf{u}^n$, $n + 1 \rightarrow n$ and go to (1) else go to (2).**

Now, we are ready to describe the presented method for the main mathematical model. For Eq. (4.23) in the numerical procedure, the unknown solution must be divided to the coarse and the fine scales solutions as

$$p = \tilde{p} + \bar{p}, \tag{5.22}$$

in which \tilde{p} and \bar{p} denote the coarse and fine scales terms, respectively. We define the following functional spaces

$$\tilde{\mathbf{P}} = \left\{ p^k \mid p^k \in H^1(\Omega), p^k = g \text{ on } \partial\Omega \right\}, \quad \bar{\mathbf{P}} = \left\{ p^k \mid p^k \in H^1(\Omega), p^k = 0 \text{ on } \partial\Omega \right\}. \tag{5.23}$$

Similarly, for the test functional spaces, we set $\chi = \tilde{\chi} + \bar{\chi}$ and

$$\tilde{\mathcal{V}} = \left\{ \chi \mid \chi \in H^1(\Omega), \chi = 0 \text{ on } \partial\Omega \right\}, \quad \bar{\mathcal{V}} = \left\{ \chi \mid \chi \in H^1(\Omega), \chi = 0 \text{ on } \partial\Omega \right\}, \tag{5.24}$$

so $\mathcal{V} = \tilde{\mathcal{V}} + \bar{\mathcal{V}}$. Substituting the above test and trial functions in Eq. (4.23), yields [65]

$$\left(\tilde{\chi} + \bar{\chi}, \frac{\partial^2(\tilde{p}^{k+1} + \bar{p}^{k+1})}{\partial x^2} + \frac{\partial^2(\tilde{p}^{k+1} + \bar{p}^{k+1})}{\partial y^2} \right) = \frac{1}{dt} \left(\tilde{\chi} + \bar{\chi}, \frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right), \tag{5.25}$$

or

$$-\left(\nabla \cdot (\tilde{\chi} + \bar{\chi}), \nabla \cdot (\tilde{p}^{k+1} + \bar{p}^{k+1}) \right) = \frac{1}{dt} \left(\tilde{\chi} + \bar{\chi}, \frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right). \tag{5.26}$$

Now, Eq. (4.17) will be transferred to

$$\tilde{\mathcal{V}} : -\left(\nabla \cdot \tilde{\chi}, \nabla \cdot \tilde{p}^k \right) = \frac{1}{dt} \left(\tilde{\chi}, \frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right) + \left(\nabla \cdot \tilde{\chi}, \nabla \cdot \tilde{p}^k \right), \tag{5.27}$$

$$\bar{\mathcal{V}} : -\left(\nabla \cdot \bar{\chi}, \nabla \cdot \bar{p}^k \right) = \frac{1}{dt} \left(\bar{\chi}, \frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right) + \left(\nabla \cdot \bar{\chi}, \nabla \cdot \bar{p}^k \right). \tag{5.28}$$

We must consider a set of functions such that their summation is equal to one, e.g.,

$$\text{supp}\{\phi_i\} = \Omega_i, \quad \forall x \in \Omega, \quad \sum_i \phi_i = 1. \tag{5.29}$$

The space of local enrichment basis functions is

$$\xi_i(\Omega_i) = \text{span} \left\{ \mathcal{J}_i^j \right\}. \tag{5.30}$$

The used space of functions for the approximate solution is

$$\xi(\Omega) = \text{span} \left\{ \phi_i \mathcal{J}_i^j \right\}. \tag{5.31}$$

Thus the approximation solution will be

$$p^k(\mathcal{S}) = \sum_i \sum_{\mathcal{J}_i^j \in \xi_i} \phi_i \mathcal{J}_i^j(\mathcal{S}) p_{i,j}^k. \tag{5.32}$$

In the current study, we consider the second-order ($p = 2$) basis [65]

$$\{\mathcal{J}_i^j\} = \{\mathcal{J}_i^1, \mathcal{J}_i^2, \mathcal{J}_i^3\} = \{1, (x - x_i)^2, (y - y_i)^2\}.$$

In the following, we assume

$$p^k(x, y) = \sum_i \phi_i \left(p_{i,0} + (x - x_i)^2 p_{i,1} + (y - y_i)^2 p_{i,2} \right), \tag{5.33}$$

in which ϕ_i are shape functions of IMLS approximation. So Eq. (5.33) changes to

$$p^k(x, y) = \sum_i \phi_i p_{i,0}^k + \sum_i \phi_i (x - x_i)^2 p_{i,1}^k + \sum_i \phi_i (y - y_i)^2 p_{i,2}^k. \tag{5.34}$$

Let the approximate solution of the coarse and fine scales be

$$\tilde{p}^k(x, y) = \sum_i \phi_i p_{i,0}^k, \tag{5.35}$$

$$\bar{p}^k(x, y) = \sum_i \phi_i (x - x_i)^2 p_{i,1}^k + \sum_i \phi_i (y - y_i)^2 p_{i,2}^k. \tag{5.36}$$

Substituting Eqs. (5.35) and (5.36) in Eqs. (5.27) and (5.28), respectively, gives

$$\mathbf{A}_1 \tilde{\mathbf{p}}^k = \mathbf{F}_1, \tag{5.37}$$

$$\mathbf{A}_2 \bar{\mathbf{p}}^k = \mathbf{F}_2, \tag{5.38}$$

in which

$$\mathbf{A}_1 = \begin{bmatrix} -(\nabla \cdot \phi_1, \nabla \cdot \phi_1) & -(\nabla \cdot \phi_1, \nabla \cdot \phi_2) & \dots & -(\nabla \cdot \phi_1, \nabla \cdot \phi_N) \\ -(\nabla \cdot \phi_2, \nabla \cdot \phi_1) & -(\nabla \cdot \phi_2, \nabla \cdot \phi_2) & \dots & -(\nabla \cdot \phi_2, \nabla \cdot \phi_N) \\ \vdots & \vdots & \ddots & \vdots \\ -(\nabla \cdot \phi_N, \nabla \cdot \phi_1) & -(\nabla \cdot \phi_N, \nabla \cdot \phi_2) & \dots & -(\nabla \cdot \phi_N, \nabla \cdot \phi_2) \end{bmatrix}_{N \times N}, \tag{5.39}$$

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{A}_2^{(1)} & \mathbf{A}_2^{(2)} \\ \mathbf{A}_2^{(3)} & \mathbf{A}_2^{(4)} \end{bmatrix}_{2N \times 2N}, \tag{5.40}$$

such that

$$\left(\mathbf{A}_2^{(1)}\right)_{i,j} = -\left(\nabla \cdot \left(\phi_i(x - x_i)^2\right), \nabla \cdot \left(\phi_j(x - x_j)^2\right)\right) \quad 1 \leq i, j \leq N, \tag{5.41}$$

$$\left(\mathbf{A}_2^{(2)}\right)_{i,j} = -\left(\nabla \cdot \left(\phi_i(x - x_i)^2\right), \nabla \cdot \left(\phi_j(y - y_j)^2\right)\right), \quad 1 \leq i, j \leq N, \tag{5.42}$$

$$\left(\mathbf{A}_2^{(3)}\right)_{i,j} = -\left(\nabla \cdot \left(\phi_i(y - y_i)^2\right), \nabla \cdot \left(\phi_j(x - x_j)^2\right)\right), \quad 1 \leq i, j \leq N, \tag{5.43}$$

$$\left(\mathbf{A}_2^{(4)}\right)_{i,j} = -\left(\nabla \cdot \left(\phi_i(y - y_i)^2\right), \nabla \cdot \left(\phi_j(y - y_j)^2\right)\right), \quad 1 \leq i, j \leq N, \tag{5.44}$$

$$\tilde{\mathbf{p}}^k = \left[p_{1,0}^k \quad p_{2,0}^k \quad \dots \quad p_{N,0}^k \right]^T, \tag{5.45}$$

$$\bar{\mathbf{p}}^k = \left[p_{1,1}^k \quad p_{2,1}^k \quad \dots \quad p_{N,1}^k \quad p_{1,2}^k \quad p_{2,2}^k \quad \dots \quad p_{N,2}^k \right]^T, \tag{5.46}$$

$$(\mathbf{F}_1)_i = \frac{1}{dt} \left(\phi_i, \frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right), \quad 1 \leq i \leq N, \tag{5.47}$$

$$(\mathbf{F}_2)_i = \frac{1}{dt} \left(\phi_i(x - x_i)^2, \frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right), \quad 1 \leq i \leq N, \tag{5.48}$$

$$(\mathbf{F}_2)_{i+N} = \frac{1}{dt} \left(\phi_i(y - y_i)^2, \frac{\partial \mathbb{M}^k}{\partial x} + \frac{\partial \mathbb{N}^k}{\partial y} \right), \quad 1 \leq i \leq N. \tag{5.49}$$

We define the following functional spaces

$$\tilde{\mathbf{U}} = \left\{ u^k \mid u^k \in H^1(\Omega), \quad u^k = g \text{ at } \partial\Omega \right\}, \quad \bar{\mathbf{U}} \equiv \left\{ u^k \mid u^k \in H^1(\Omega), \quad u^k = 0 \text{ at } \partial\Omega \right\}. \tag{5.50}$$

The variational weak form will be

$$(\tilde{\chi} + \bar{\chi}, \tilde{u}^{k+1} + \bar{u}^{k+1}) = (\tilde{\chi} + \bar{\chi}, \mathbb{M}^k) - dt \left(\tilde{\chi} + \bar{\chi}, \frac{\partial p^k}{\partial x} \right), \tag{5.51}$$

$$(\tilde{\chi} + \bar{\chi}, \tilde{v}^{k+1} + \bar{v}^{k+1}) = (\tilde{\chi} + \bar{\chi}, \mathbb{N}^k) - dt \left(\tilde{\chi} + \bar{\chi}, \frac{\partial p^k}{\partial y} \right), \tag{5.52}$$

$$(\tilde{\chi} + \bar{\chi}, \tilde{w}^{k+1} + \bar{w}^{k+1}) = (\tilde{\chi} + \bar{\chi}, \mathbb{P}^k). \tag{5.53}$$

Now, the above relations can be transferred to the coarse scale problem

$$(\tilde{\chi}, \tilde{u}^{k+1}) = (\tilde{\chi}, \mathbb{M}^k) - dt \left(\tilde{\chi}, \frac{\partial p^k}{\partial x} \right) - (\tilde{\chi}, \bar{u}^{k+1}), \tag{5.54}$$

$$(\tilde{\chi}, \tilde{v}^{k+1}) = (\tilde{\chi}, \mathbb{N}^k) - dt \left(\tilde{\chi}, \frac{\partial p^k}{\partial y} \right) - (\tilde{\chi}, \bar{v}^{k+1}), \tag{5.55}$$

$$(\tilde{\chi}, \tilde{w}^{k+1}) = (\tilde{\chi}, \mathbb{P}^k) - (\tilde{\chi}, \bar{w}^{k+1}), \tag{5.56}$$

as well as the fine scale problem

$$(\bar{\chi}, \bar{u}^{k+1}) = (\bar{\chi}, \mathbb{M}^k) - dt \left(\bar{\chi}, \frac{\partial p^k}{\partial x} \right) - (\bar{\chi}, \tilde{u}^{k+1}), \tag{5.57}$$

$$(\bar{\chi}, \bar{v}^{k+1}) = (\bar{\chi}, \mathbb{N}^k) - dt \left(\bar{\chi}, \frac{\partial p^k}{\partial y} \right) - (\bar{\chi}, \tilde{v}^{k+1}), \tag{5.58}$$

$$(\bar{\chi}, \bar{w}^{k+1}) = (\bar{\chi}, \mathbb{P}^k) - (\bar{\chi}, \tilde{w}^{k+1}). \tag{5.59}$$

Let

$$u^k(x, y) = \sum_i \phi_i \left(u_{i,0} + (x - x_i)^2 u_{i,1} + (y - y_i)^2 u_{i,2} \right), \tag{5.60}$$

$$v^k(x, y) = \sum_i \phi_i \left(v_{i,0} + (x - x_i)^2 v_{i,1} + (y - y_i)^2 v_{i,2} \right), \tag{5.61}$$

$$w^k(x, y) = \sum_i \phi_i \left(w_{i,0} + (x - x_i)^2 w_{i,1} + (y - y_i)^2 w_{i,2} \right). \tag{5.62}$$

Now, we can split Eqs. (5.60), (5.61) and (5.62) into coarse scale and fine scale problems as

$$\begin{cases} \tilde{u}^k(x, y) = \sum_i \phi_i u_{i,0}^k, \\ \bar{u}^k(x, y) = \sum_i \phi_i (x - x_i)^2 u_{i,1}^k + \sum_i \phi_i (y - y_i)^3 u_{i,2}^k, \end{cases} \tag{5.63}$$

$$\begin{cases} \tilde{v}^k(x, y) = \sum_i \phi_i v_{i,0}^k, \\ \bar{v}^k(x, y) = \sum_i \phi_i (x - x_i)^2 v_{i,1}^k + \sum_i \phi_i (y - y_i)^3 v_{i,2}^k. \end{cases} \tag{5.64}$$

$$\begin{cases} \tilde{w}^k(x, y) = \sum_i \phi_i w_{i,0}^k, \\ \bar{w}^k(x, y) = \sum_i \phi_i (x - x_i)^2 w_{i,1}^k + \sum_i \phi_i (y - y_i)^3 w_{i,2}^k. \end{cases} \tag{5.65}$$

Substituting Eqs. (5.63)-(5.65) in relations (5.54)-(5.59), gives

$$\begin{cases} \mathbf{A}_1 \tilde{\mathbf{u}}^{k+1} = \mathbf{J}_1, & \mathbf{B}_1 \tilde{\mathbf{v}}^{k+1} = \mathbf{G}_1, & \mathbf{C}_1 \tilde{\mathbf{w}}^{k+1} = \mathbf{H}_1, \\ \mathbf{A}_2 \bar{\mathbf{u}}^{k+1} = \mathbf{J}_2, & \mathbf{B}_2 \bar{\mathbf{v}}^{k+1} = \mathbf{G}_2, & \mathbf{C}_2 \bar{\mathbf{w}}^{k+1} = \mathbf{H}_2, \end{cases} \tag{5.66}$$

in which matrices and vectors in relation (5.66) can be achieved in a similar way to the first stage. In the final relation, we can see that the difference between traditional EFG method and the variational EFG method is two more terms.

Algorithm 1 Combination of VMIEFG method with POD approach.

- 1: Compute the derivative matrices in x- and y-direction, according to Eqs. (5.39) and (5.40).
 - 2: Consider some collocation nodes in the computational domain based on the regular or irregular distributed points.
 - 3: Evaluate relations (4.20), (4.21) and (4.22).
 - 4: Obtain solutions of relation (5.66) according to previous step.
 - 5: Obtain the snapshot $\Phi = [\Phi_k(\mathbf{x}, t_k)] \in R^{n_{xy} \times n_k}$ from the full discretization.
 - 6: Apply SVD for $\Phi = [\Phi_k(\mathbf{x}, t_k)]$, to get the singular values.
 - 7: Investigate relation (3.23) to derive suitable POD basis to solve linear system of equations (5.66).
-

6. Numerical argument

The numerical results are carried out using MATLAB 2018b software on an Intel Core i7 machine with 16 GB of memory.

6.1. Example 1 (Accuracy test problem)

To check the accuracy of the proposed technique, we investigate the following model [66]

$$\frac{\partial u}{\partial t} - \varepsilon \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} = 0, \quad (\mathbf{x}, t) \in \Omega \times I, \tag{6.1}$$

$$\frac{\partial v}{\partial t} - \varepsilon \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} = w, \quad (\mathbf{x}, t) \in \Omega \times I, \tag{6.2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \quad (\mathbf{x}, t) \in \Omega \times I, \tag{6.3}$$

$$\frac{\partial w}{\partial t} - \gamma \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right) + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} = 0, \quad (\mathbf{x}, t) \in \Omega \times I, \tag{6.4}$$

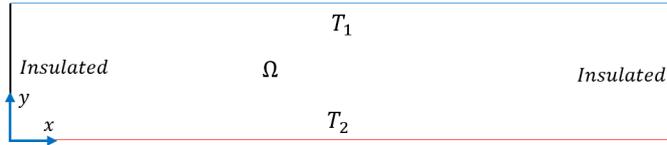


Fig. 1. A computational domain for Example 2.

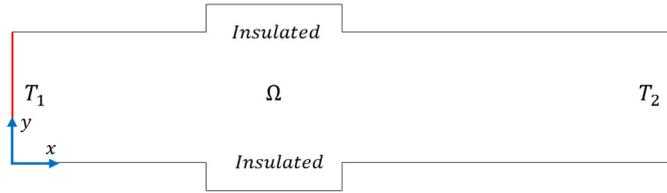


Fig. 2. A computational domain for Example 3.

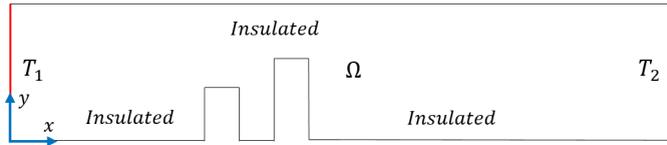


Fig. 3. A computational domain for Example 3.

with following exact solution

$$\begin{cases} u(x, y, t) = -P(t) \sin(kx) \cos(y), \\ v(x, y, t) = kA(t) \cos(kx) \sin(y), \\ w(x, y, t) = \lambda_k B(t) \sin(kx) \sin(y), \end{cases}$$

in which $P(t)$, $A(t)$ and $B(t)$ can be obtained by solving the following ODEs [66]

$$\begin{cases} \frac{dP(t)}{dt} = \nu \lambda_k P(t) - Ri \frac{k}{\lambda_k} A(t), \\ \frac{dA(t)}{dt} = kP(t) + \kappa \lambda_k A(t) - 2k B(t) P(t), \\ \frac{dB(t)}{dt} = -4\kappa B(t) - kP(t) A(t), \end{cases} \tag{6.5}$$

and also $k = 1$, $\lambda_k = -(k^2 + 1)$, $\nu = \kappa = 0.001$ and $Ri = 1$ and also $\Omega = [0, \pi]^2$. We use a fourth-order Runge-Kutta method to compute $P(t)$, $A(T)$ and $B(t)$ functions in relation (6.5). Tables 1, 2 and 3 show the error obtained for components u , v and w at final time $T_f = 10$ and $dt = 10^{-5}$ and based on the different values of distributed nodes in the computational domain for Example 1. Tables 1, 2 and 3 illustrate the efficiency and accuracy of the proposed technique.

To display that the POD plan can approximate the full model, there are some idea such as

- Root mean squares error (RMSE) [43]

$$RMES^n = \sqrt{\frac{\sum_{i=1}^N (U_i^{n(full)} - U_i^{n(POD)})^2}{N}}, \tag{6.6}$$

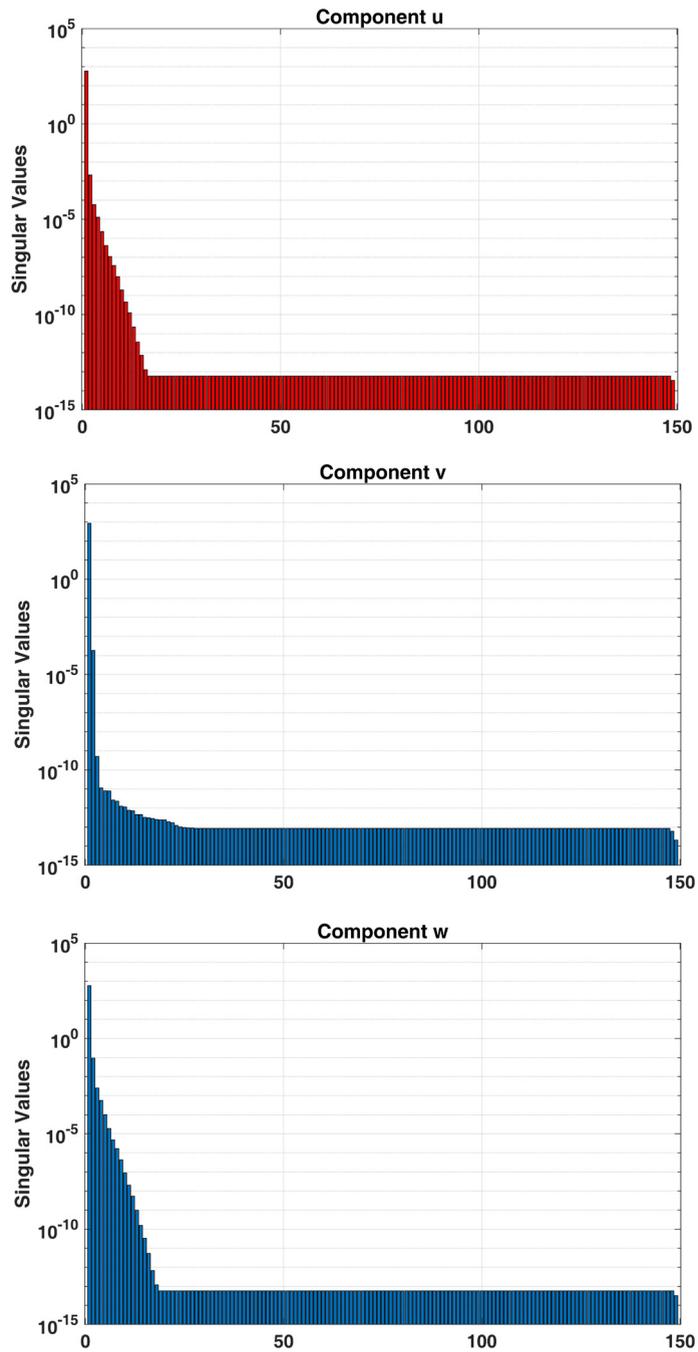


Fig. 4. Singular values based on different snapshots for Example 1.

- Energy relation [17,43]

$$I(m) = \frac{\sum_{i=1}^d \lambda_i}{\sum_{i=1}^l \lambda_i}. \tag{6.7}$$

- Singular values of snapshot matrix [17,43].

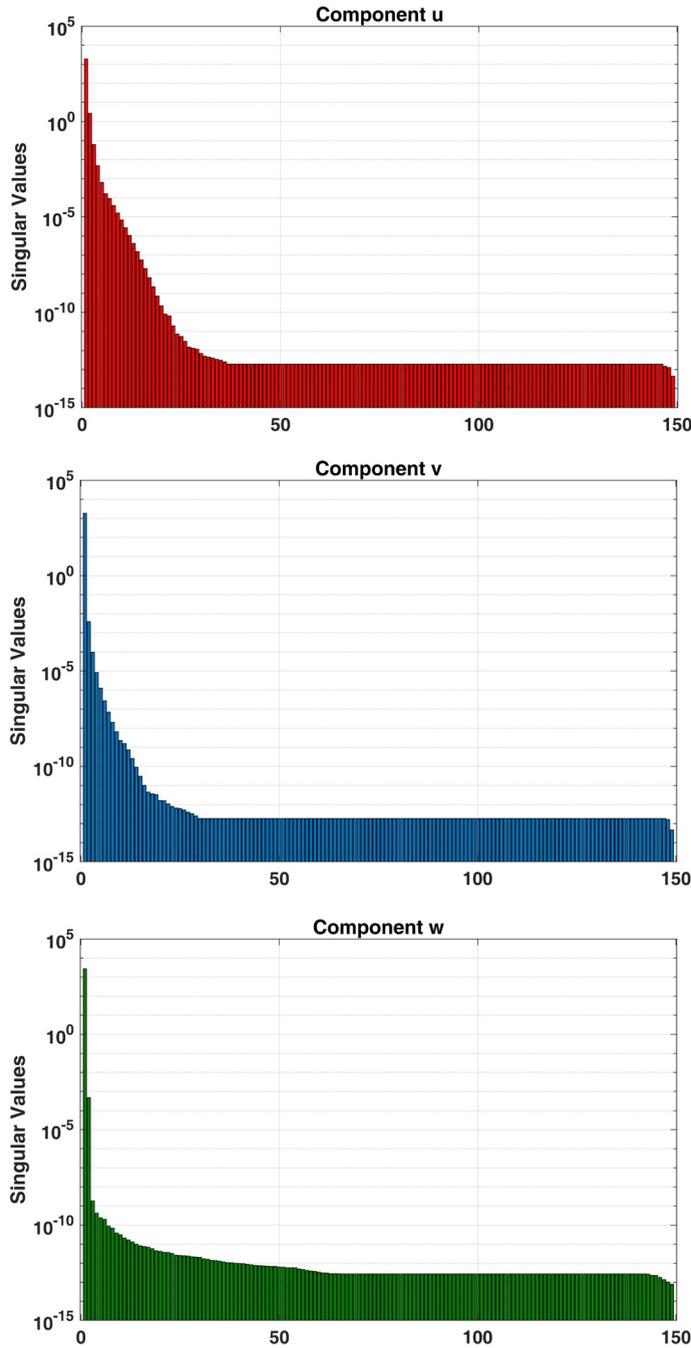


Fig. 5. Singular values based on different snapshots for Example 2.

In the current paper, we engage the singular values of snapshot matrix. According to this idea, the best selection is where the singular value goes to zero. Fig. 4 demonstrates the singular values of snapshots u , v and w components. By computing the singular values in Fig. 4, we can conclude that $\lambda_{20} \leq \times 10^{-15}$ for 800 collocation points. Thus, we can choose 20 POD basis.

6.2. Example 2

In the current example, we solve the Rayleigh-Benard convection problem. The distributed nodes in the computational region are 4000 points. According to Fig. 1 the bottom of domain is heated and the top of domain is cooled furthermore the rest walls are insulated. Also the velocities in x - and y -directions are zero. The computational domain of Fig. 1 is

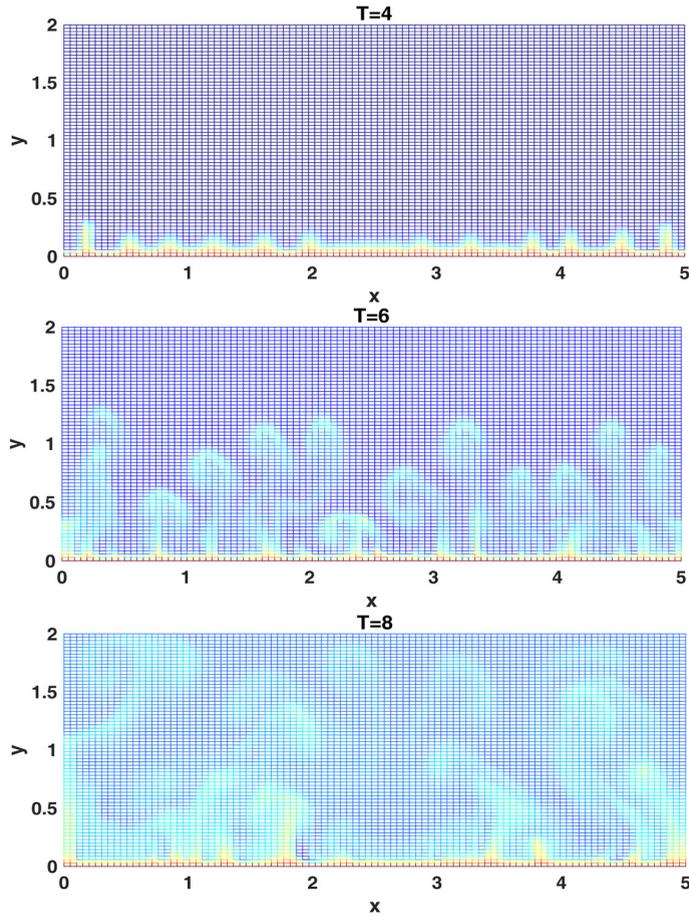


Fig. 6. Temperature profile for Example 2.

Table 1

Results computed for component u with $T_f = 10$ and $dt = 10^{-5}$ for Example 1.

N	L_∞	L_2	CPU time(s)
100	5.3511×10^{-5}	8.7013×10^{-5}	150
200	2.0135×10^{-5}	4.5318×10^{-5}	420
400	8.6501×10^{-6}	1.5908×10^{-5}	1340
800	3.0511×10^{-6}	7.3801×10^{-6}	2007
Number of POD basis	L_∞	L_2	CPU time(s)
20	2.3411×10^{-1}	7.8891×10^{-1}	8
40	1.5312×10^{-2}	6.4011×10^{-2}	15
50	6.3015×10^{-3}	1.2077×10^{-2}	29
60	2.3015×10^{-3}	8.7739×10^{-3}	49

$\Omega = (0, 5) \times (0, 2)$. Fig. 5 demonstrates the singular values based on the component u , v and w and different snapshots for Example 2. Results of Fig. 5 are based on $Re = 100$, $Pr = 7$, $Pe = Re \times Pr$, $T_N = 55$ and $T_S = 35$. From Fig. 5, we can conclude that number of 20 POD basis is suitable for simulation of this problem according to the mentioned parameters. Thus, Fig. 6 illustrates the approximation of temperature in final times $T = 4$, $T = 6$ and $T = 8$ for Example 2. From Fig. 6 it is clear that the hot fluid goes to the cold place. The computational region of Figs. 8 and 9 is $\Omega = [0, 100] \times [0, 50]$. Figs. 8 and 9 display the simulations of fluid velocity and temperature profile with 30 POD basis for Example 2. In this status, Fig. 8 depicts the fluid flow as it is called the Benard convection multi-cellular patterns. In Table 4, we compare the used CPU time of EFG, VMEFG and POD-VMEFG techniques based on final time $T_f = 10$ and $dt = 10^{-5}$ and the number of collocation points $N = 800$ and $N = 1600$ for Test problem 2.

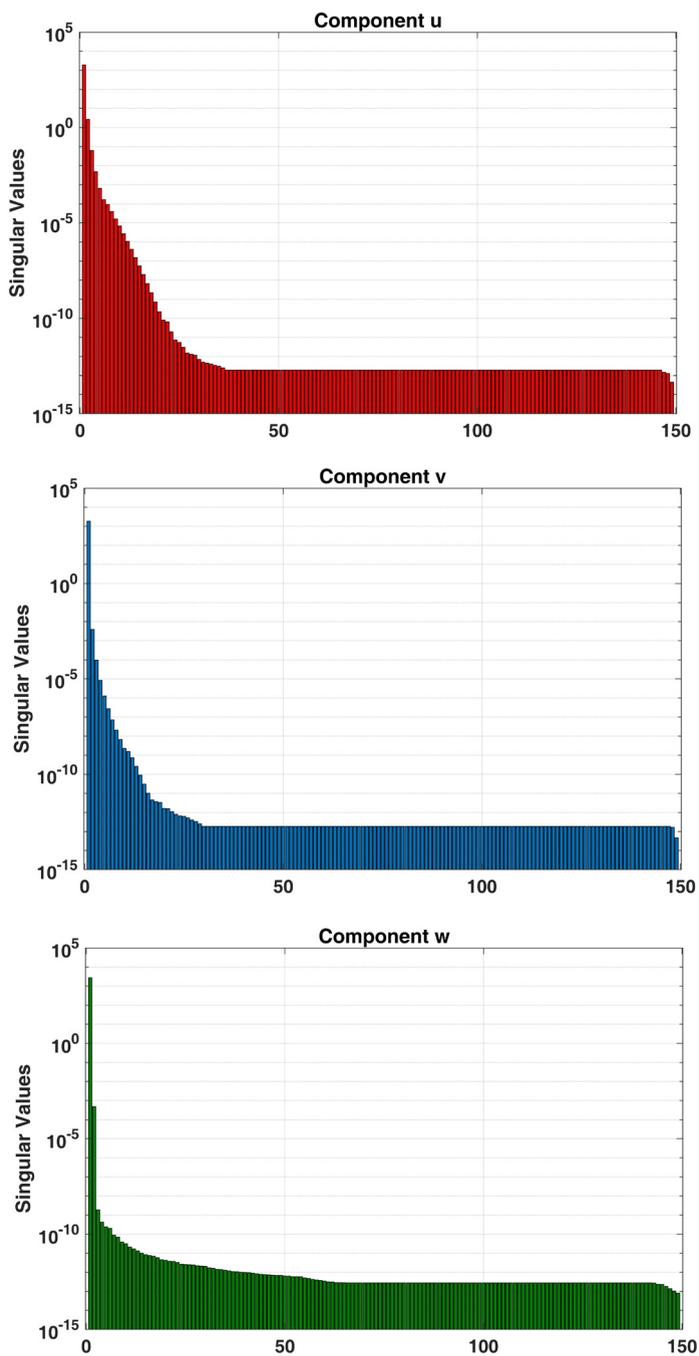


Fig. 7. Singular values based on different snapshots for Example 3.

Table 2

Results computed for component v with $T_f = 10$ and $dt = 10^{-5}$ for Example 1.

N	VMEFG			EFG		
	L_∞	L_2	CPU time	L_∞	L_2	CPU time
100	8.33×10^{-4}	1.02×10^{-3}	150	2.11×10^{-1}	8.69×10^{-1}	25
200	6.01×10^{-5}	7.65×10^{-4}	420	3.80×10^{-2}	7.99×10^{-2}	87
400	2.03×10^{-5}	8.77×10^{-5}	1340	7.52×10^{-3}	1.22×10^{-2}	167
800	9.10×10^{-6}	7.33×10^{-6}	2007	9.08×10^{-4}	6.55×10^{-3}	387

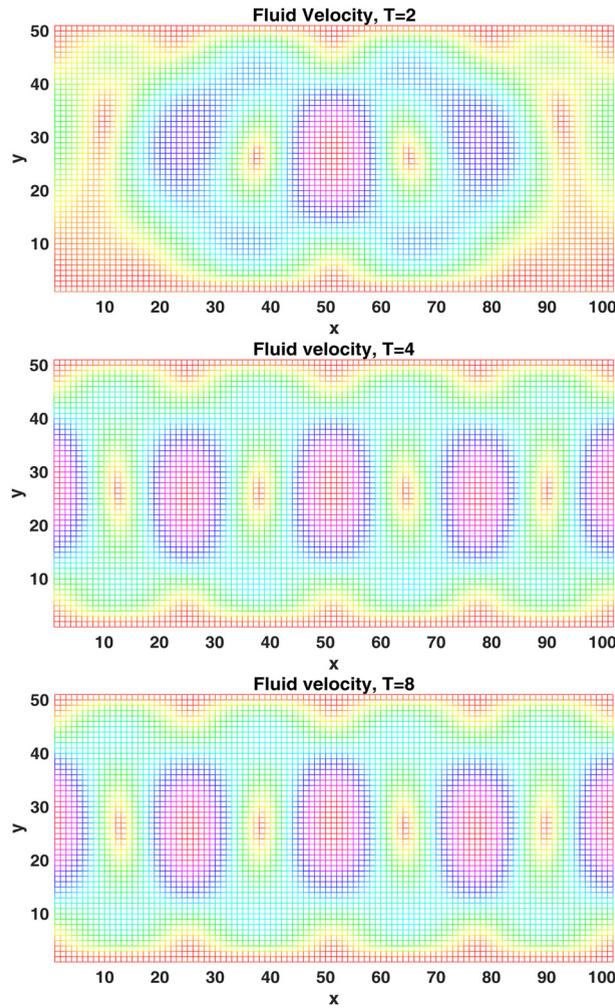


Fig. 8. Fluid velocity for Example 2.

Table 3
Results computed for component w with $T_f = 10$ and $dt = 10^{-5}$ for Example 1.

N	VMEFG			EFG		
	L_∞	L_2	CPU time	L_∞	L_2	CPU time
100	7.22×10^{-5}	6.339×10^{-4}	150	3.52×10^{-1}	9.97×10^{-1}	25
200	3.56×10^{-5}	7.99×10^{-5}	420	5.30×10^{-2}	8.99×10^{-2}	87
400	8.79×10^{-6}	2.00×10^{-5}	1340	8.97×10^{-3}	2.03×10^{-2}	167
800	4.55×10^{-6}	8.70×10^{-6}	2007	1.0×10^{-3}	9.88×10^{-3}	387

6.3. Example 3 (Channel flow with two rectangular protrusions)

In the current example, we investigate two computational domains that they are a channel with two rectangular protrusions. The total width of one channel is 2.4 and with two identical rectangular protrusions at the bottom and at the top of the channel with width 0.7 and length 0.05 as are depicted in Fig. 2. The total width of another channel is 2.85 and also with two rectangular protrusions at the bottom of the channel that length and width of small rectangular are 0.4 and 0.1, respectively, and the length and width of the larger one are 0.6 and 0.1, respectively, as these are shown in Fig. 3. In Figs. 2 and 3 the black walls are insulated thus the related boundary conditions for them are homogeneous Neumann boundary conditions. Also, In Figs. 2 and 3 values $T_S = 10$ and $T_N = 1$ are the hottest and coldest places of these domains, respectively. For two computational regions, we consider $Re = 100$, $Pr = 7$, $Pe = Re \times Pr$, $T_N = 1$ and $T_S = 10$. Also the velocities in x - and y -directions are zero. Similarly, Fig. 7 illustrates the singular values based on the components u , v and w and different snapshots with $Re = 2000$, $Pr = 1$, $Pe = Re \times Pr$, $T_N = 0$ and $T_S = 1$ for Example 3. By computing the singular values, we can see $\lambda_{20} \leq 2.3310 \times 10^{-10}$ for 1450 collocation points. In Eq. (3.23), the value of $I(m)$ converges to one. On the other

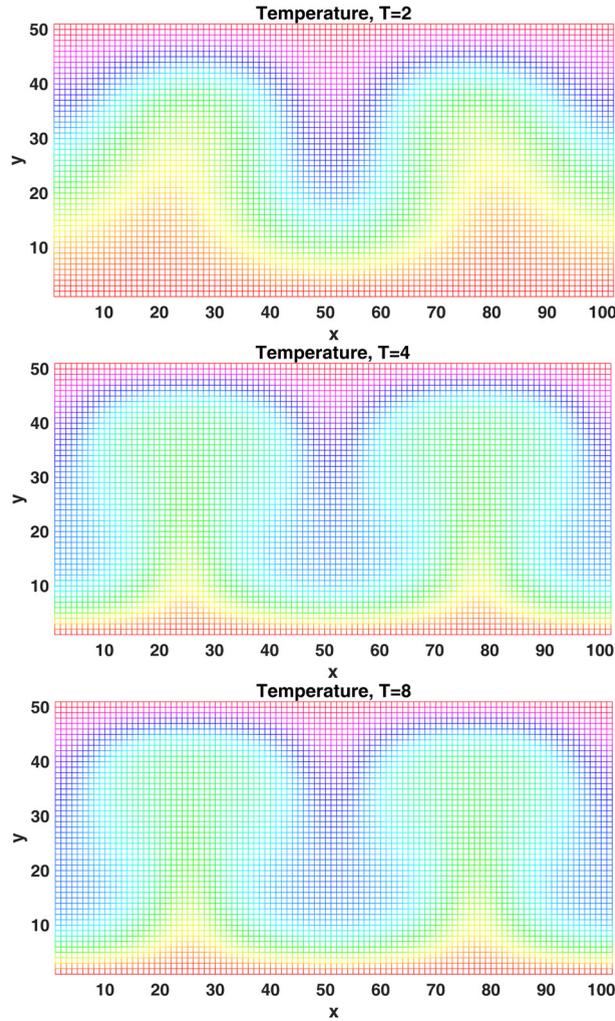


Fig. 9. Temperature profile for Example 2.

Table 4
The used CPU time calculated with $T_f = 10$ and $dt = 10^{-5}$ for Example 2.

N	EFG Method	VMEFG Method	POD-MVEFG Method	
			20 POD basis	40 POD basis
800	37 min	1 hr & 22 min	8 s	21 s
1600	1 hr & 4 min	2 hr & 52 min		

hand, from Eq. (3.23), we have $I(1) = 0.999032$ and $I(20) = 0.999999$. Thus, we can declare if $\lambda_{i+1} \leq 2.3310 \times 10^{-10}$ then the number of the optimal POD basis will be equal to i . In other words, we compute $I(m)$ based upon the different values of m . Therefore, the smallest number m , where the value of $I(m)$ is enough closed to one, can be selected as the number of POD bases. For Example 3, we take 20 POD basis for 1450 nodes. Figs. 10 and 11 illustrate the numerical temperature profile based on the two computational domains and $dt = 10^{-5}$ for Example 3. In Table 5, we compare the used CPU time of EFG, VMEFG and POD-VMEFG techniques based on final time $T_f = 10$ and $dt = 10^{-5}$ and the number of collocation points $N = 1000$ and $N = 12000$ for Test problem 3.

7. Conclusion

In this paper, we developed a new reduced order model based on the meshless variational multiscale interpolating element free Galerkin (IEFG) method for solving the two-dimensional nonstationary Boussinesq equations. The interpolating moving least squares approximation is employed in the IEFG technique to derive an improved meshless weak form formu-

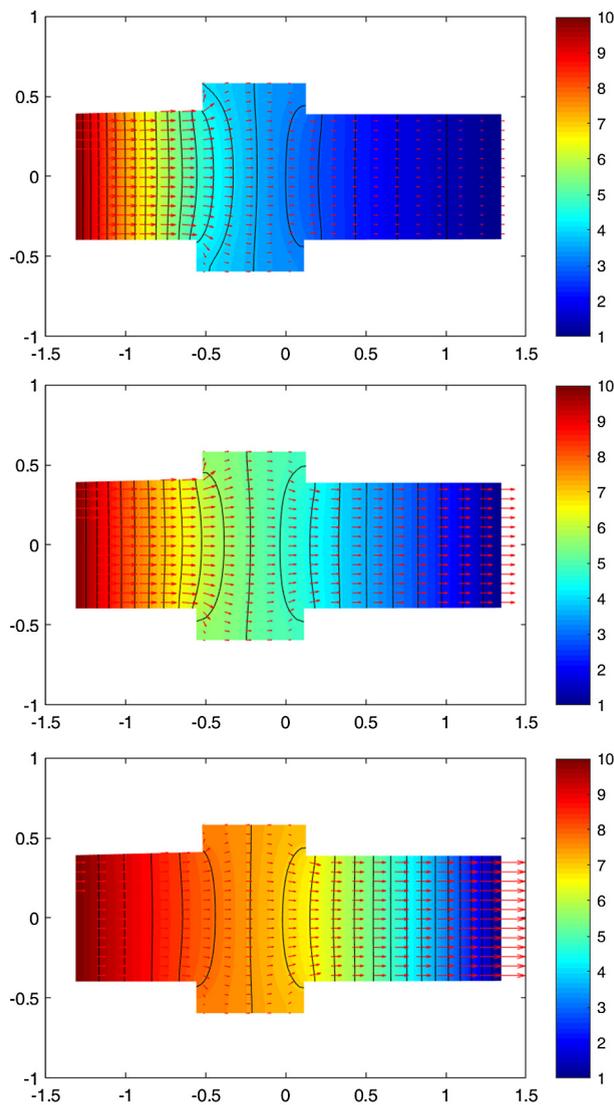


Fig. 10. Temperature profile for Example 3.

Table 5

The used CPU time computed with $T_f = 2$ and $dt = 10^{-5}$ for Example 3.

N	EFG Method	MVEFG Method	POD-MVEFG Method	
			50 POD basis	80 POD basis
1000	56 min	2 hr & 15 min	35 s	177 s
12000	2 hr & 43 min	4 hr & 2 min		

lation. First, the time variable is discretized by a finite difference scheme. The time-discrete plane is based on a two-step formulation such that in the first step we calculated the pressure component by solving a Poisson equation. Then, in the second step we updated the velocity vector. By applying the variational multiscale approach, we increase the accuracy of the IEFG method. Furthermore, to increase the efficiency of the proposed method, we proposed a new reduced order model based on the proper orthogonal decomposition method. The numerical results show the accuracy and efficiency of the new scheme.

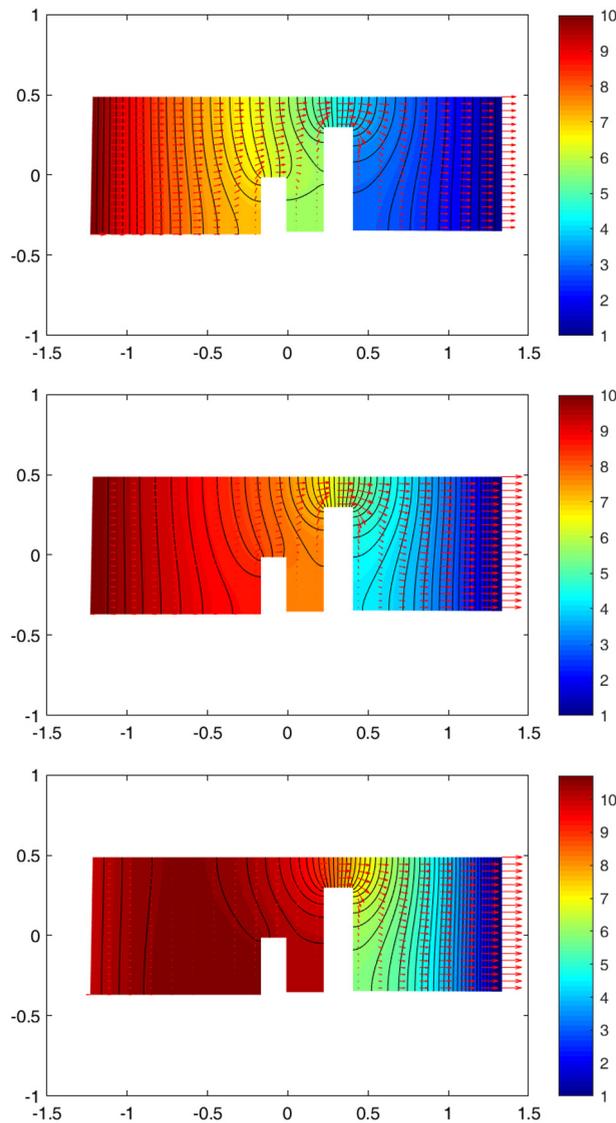


Fig. 11. Temperature profile for Example 3.

CRediT authorship contribution statement

Author’s contributions: Six authors contributed equally and significantly in writing this article. Authors wrote, read and approved the final manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors are very grateful to the reviewers for carefully reading this paper and for their comments and suggestions which have improved the paper. A. Khodadadian and C. Heitzinger acknowledge financial support by FWF Austrian Science Fund START Project no. Y660 PDE Models for Nanotechnology.

References

- [1] Z.D. Luo, A stabilized Crank–Nicolson mixed finite volume element formulation for the non-stationary incompressible Boussinesq equations, *J. Sci. Comput.* 66 (2) (2016) 555–576.
- [2] S. Ravindran, Error analysis for Galerkin POD approximation of the nonstationary Boussinesq equations, *Numer. Methods Partial Differ. Equ.* 27 (6) (2011) 1639–1665.
- [3] J. Wu, *The 2D Incompressible Boussinesq Equations*, Peking University Summer School Lecture Notes, Beijing, 2012.
- [4] Z. Luo, A stabilized mixed finite element formulation for the non-stationary incompressible Boussinesq equations, *Acta Math. Sci.* 36 (2) (2016) 385–393.
- [5] Z. Luo, Proper orthogonal decomposition-based reduced-order stabilized mixed finite volume element extrapolating model for the nonstationary incompressible Boussinesq equations, *J. Math. Anal. Appl.* 425 (1) (2015) 259–280.
- [6] D.A. Bistrián, R.F. Susan-Resiga, Weighted proper orthogonal decomposition of the swirling flow exiting the hydraulic turbine runner, *Appl. Math. Model.* 40 (2016) 4057–4078.
- [7] S. Walton, O. Hassan, K. Morgan, Reduced order modelling for unsteady fluid flow using proper orthogonal decomposition and radial basis functions, *Appl. Math. Model.* 37 (2013) 8930–8945.
- [8] F. Fang, C. Pain, I. Navon, G. Gorman, M. Piggott, P. Allison, P. Farrell, A. Goddard, A POD reduced order unstructured mesh ocean modelling method for moderate Reynolds number flows, *Ocean Model.* 28 (1–3) (2009) 127–136.
- [9] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annu. Rev. Fluid Mech.* 25 (1) (1993) 539–575.
- [10] G. Kerschen, J.C. Golinval, A.F. Vakakis, L.A. Bergman, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, *Nonlinear Dyn.* 41 (1–3) (2005) 147–169.
- [11] S. Chaturantabut, Dimension Reduction for Unsteady Nonlinear Partial Differential Equations via Empirical Interpolation Methods, *Tech. Rep.*, 2009.
- [12] S. Chaturantabut, D.C. Sorensen, A state space error estimate for POD-DEIM nonlinear model reduction, *SIAM J. Numer. Anal.* 50 (1) (2012) 46–63.
- [13] D. Fang, W. Le, T. Zhang, The 2D regularized incompressible Boussinesq equations with general critical dissipations, *J. Math. Anal. Appl.* 461 (1) (2018) 868–915.
- [14] D. Xiao, Z. Lin, F. Fang, C.C. Pain, I.M. Navon, P. Salinas, A. Muggeridge, Non-intrusive reduced-order modeling for multiphase porous media flows using Smolyak sparse grids, *Int. J. Numer. Methods Fluids* 83 (2) (2017) 205–219.
- [15] S. Ravindran, Reduced-order adaptive controllers for fluid flows using POD, *J. Sci. Comput.* 15 (4) (2000) 457–478.
- [16] S.S. Ravindran, A reduced-order approach for optimal control of fluids using proper orthogonal decomposition, *Int. J. Numer. Methods Fluids* 34 (5) (2000) 425–448.
- [17] D. Xiao, F. Fang, C. Pain, P. Salinas, I. Navon, Non-intrusive model reduction for a 3D unstructured mesh control volume finite element reservoir model and its application to fluvial channels, *Int. J. Oil Gas Coal Technol.* 19 (2018) 316–339.
- [18] D. Xiao, F. Fang, J. Zheng, C. Pain, I. Navon, Machine learning-based rapid response tools for regional air pollution modelling, *Atmos. Environ.* 199 (2019) 463–473.
- [19] D. Xiao, P. Yang, F. Fang, J. Xiang, C.C. Pain, I. Navon, M. Chen, A non-intrusive reduced-order model for compressible fluid and fractured solid coupling and its application to blasting, *J. Comput. Phys.* 330 (2017) 221–244.
- [20] D. Xiao, F. Fang, A.G. Buchan, C.C. Pain, I.M. Navon, J. Du, G. Hu, Non-linear model reduction for the Navier–Stokes equations using residual DEIM method, *J. Comput. Phys.* 263 (2014) 1–18.
- [21] D. Xiao, F. Fang, J. Du, C. Pain, I. Navon, A. Buchan, A.H. Elsheikh, G. Hu, Non-linear Petrov–Galerkin methods for reduced order modelling of the Navier–Stokes equations using a mixed finite element pair, *Comput. Methods Appl. Mech. Eng.* 255 (2013) 147–157.
- [22] D. Xiao, F. Fang, C. Pain, G. Hu, Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation, *Int. J. Numer. Methods Fluids* 79 (11) (2015) 580–595.
- [23] P. Zhang, X. Zhang, H. Xiang, L. Song, A fast and stabilized meshless method for the convection-dominated convection-diffusion problems, *Numer. Heat Transf. Appl.* 70 (4) (2016) 420–431.
- [24] X. Zhang, H. Xiang, A fast meshless method based on proper orthogonal decomposition for the transient heat conduction problems, *Int. J. Heat Mass Transf.* 84 (2015) 729–739.
- [25] J. Du, I. Navon, J. Steward, A. Alekseev, Z. Luo, Reduced-order modeling based on POD of a parabolized Navier–Stokes equation model I: forward model, *Int. J. Numer. Methods Fluids* 69 (3) (2012) 710–730.
- [26] J. Du, I. Navon, J. Zhu, F. Fang, A. Alekseev, Reduced order modeling based on POD of a parabolized Navier–Stokes equations model II: trust region POD 4D VAR data assimilation, *Comput. Math. Appl.* 65 (3) (2013) 380–394.
- [27] F. Liu, Y. Cheng, The improved element-free Galerkin method based on the nonsingular weight functions for inhomogeneous swelling of polymer gels, *Int. J. Appl. Mech.* 10 (04) (2018) 1850047.
- [28] F. Liu, Q. Wu, Y. Cheng, A meshless method based on the nonsingular weight functions for elastoplastic large deformation problems, *Int. J. Appl. Mech.* 11 (01) (2019) 1950006.
- [29] H. Cheng, M. Peng, Y. Cheng, Analyzing wave propagation problems with the improved complex variable element-free Galerkin method, *Eng. Anal. Bound. Elem.* 100 (2019) 80–87.
- [30] S. Yu, M. Peng, H. Cheng, Y. Cheng, The improved element-free Galerkin method for three-dimensional elastoplasticity problems, *Eng. Anal. Bound. Elem.* 104 (2019) 215–224.
- [31] D. Liu, Y. Cheng, The interpolating element-free Galerkin (IEFG) method for three-dimensional potential problems, *Eng. Anal. Bound. Elem.* 108 (2019) 115–123.
- [32] H. Cheng, M. Peng, Y. Cheng, The dimension splitting and improved complex variable element-free Galerkin method for 3-dimensional transient heat conduction problems, *Int. J. Numer. Methods Eng.* 114 (3) (2018) 321–345.
- [33] H. Cheng, M. Peng, Y. Cheng, A hybrid improved complex variable element-free Galerkin method for three-dimensional advection-diffusion problems, *Eng. Anal. Bound. Elem.* 97 (2018) 39–54.
- [34] M. Dehghan, M. Abbaszadeh, The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations, *J. Comput. Phys.* 351 (2017) 478–510.
- [35] M. Abbaszadeh, M. Dehghan, The reproducing kernel particle Petrov–Galerkin method for solving two-dimensional nonstationary incompressible Boussinesq equations, *Eng. Anal. Bound. Elem.* 106 (2019) 300–308.
- [36] M. Abbaszadeh, M. Dehghan, Reduced order modeling of time-dependent incompressible Navier–Stokes equation with variable density based on a local radial basis functions-finite difference (LRBF-FD) technique and the POD/DEIM method, *Comput. Methods Appl. Mech. Eng.* 364 (2020) 112914.
- [37] M. Abbaszadeh, M. Dehghan, Investigation of the Oldroyd model as a generalized incompressible Navier–Stokes equation via the interpolating stabilized element free Galerkin technique, *Appl. Numer. Math.* 150 (2020) 274–294.
- [38] M. Abbaszadeh, M. Dehghan, An upwind local radial basis functions-differential quadrature (RBFs-DQ) technique to simulate some models arising in water sciences, *Ocean Eng.* 197 (2020) 106844.

- [39] M. Abbaszadeh, M. Dehghan, A. Khodadadian, C. Heitzinger, Analysis and application of the interpolating element free Galerkin (IEFG) method to simulate the prevention of groundwater contamination with application in fluid flow, *J. Comput. Appl. Math.* 368 (2020) 112453.
- [40] M. Dehghan, M. Abbaszadeh, A. Khodadadian, C. Heitzinger, Galerkin proper orthogonal decomposition-reduced order method (POD-ROM) for solving generalized Swift-Hohenberg equation, *Int. J. Numer. Methods Heat Fluid Flow* 29 (2019) 2642–2665.
- [41] X. Li, Q. Wang, Analysis of the inherent instability of the interpolating moving least squares method when using improper polynomial bases, *Eng. Anal. Bound. Elem.* 73 (2016) 21–34.
- [42] X. Li, Error estimates for the moving least-square approximation and the element-free Galerkin method in n -dimensional spaces, *Appl. Numer. Math.* 99 (2016) 77–97.
- [43] Y. Wang, I.M. Navon, X. Wang, Y. Cheng, 2D Burgers equation with large Reynolds number using POD/DEIM and calibration, *Int. J. Numer. Methods Fluids* 82 (12) (2016) 909–931.
- [44] L. Chen, X. Liu, X. Li, The boundary element-free method for 2D interior and exterior Helmholtz problems, *Comput. Math. Appl.* 77 (3) (2019) 846–864.
- [45] L. Chen, X. Li, Boundary element-free methods for exterior acoustic problems with arbitrary and high wavenumbers, *Appl. Math. Model.* 72 (2019) 85–103.
- [46] X. Li, H. Dong, Analysis of the element-free Galerkin method for Signorini problems, *Appl. Math. Comput.* 346 (2019) 41–56.
- [47] X. Li, A meshless interpolating Galerkin boundary node method for Stokes flows, *Eng. Anal. Bound. Elem.* 51 (2015) 112–122.
- [48] T.J. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, *Comput. Methods Appl. Mech. Eng.* 127 (1–4) (1995) 387–401.
- [49] T.J. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quinicy, The variational multiscale method—a paradigm for computational mechanics, *Comput. Methods Appl. Mech. Eng.* 166 (1–2) (1998) 3–24.
- [50] A. Masud, On a stabilized finite element formulation for incompressible Navier–Stokes equations, in: *Proceedings of the Fourth US–Japan Conference on Computational Fluid Dynamics, Tokyo, Japan, 2002*, pp. 28–30.
- [51] A. Masud, R. Khurram, A multiscale/stabilized finite element method for the advection–diffusion equation, *Comput. Methods Appl. Mech. Eng.* 193 (21–22) (2004) 1997–2018.
- [52] M. Ayub, A. Masud, A new stabilized formulation for convective-diffusive heat transfer, *Numer. Heat Transf., Part B, Fundam.* 44 (1) (2003) 1–23.
- [53] A. Masud, T.J. Hughes, A stabilized mixed finite element method for Darcy flow, *Comput. Methods Appl. Mech. Eng.* 191 (39–40) (2002) 4341–4370.
- [54] A. Masud, L.A. Bergman, Application of multi-scale finite element methods to the solution of the Fokker–Planck equation, *Comput. Methods Appl. Mech. Eng.* 194 (12–16) (2005) 1513–1526.
- [55] L.P. Franca, A. Nesliturk, M. Stynes, On the stability of residual-free bubbles for convection-diffusion problems and their approximation by a two-level finite element method, *Comput. Methods Appl. Mech. Eng.* 166 (1–2) (1998) 35–49.
- [56] L.P. Franca, A. Nesliturk, On a two-level finite element method for the incompressible Navier–Stokes equations, *Int. J. Numer. Methods Eng.* 52 (4) (2001) 433–453.
- [57] L. Zhang, J. Ouyang, X. Zhang, The two-level element free Galerkin method for MHD flow at high Hartmann numbers, *Phys. Lett. A* 372 (35) (2008) 5625–5638.
- [58] L. Zhang, J. Ouyang, X.-H. Zhang, On a two-level element-free Galerkin method for incompressible fluid flow, *Appl. Numer. Math.* 59 (8) (2009) 1894–1904.
- [59] L. Zhang, J. Ouyang, X. Zhang, W. Zhang, On a multi-scale element-free Galerkin method for the Stokes problem, *Appl. Math. Comput.* 203 (2) (2008) 745–753.
- [60] T. Gerasimov, N. Noii, O. Allix, L. De Lorenzis, A non-intrusive global/local approach applied to phase-field modeling of brittle fracture, *Adv. Model. Simul. Eng. Sci.* 5 (14) (2018), <https://doi.org/10.1186/s40323-018-0105-8>.
- [61] N. Noii, F. Aldakheel, T. Wick, P. Wriggers, An adaptive global–local approach for phase-field modeling of anisotropic brittle fracture, *Comput. Methods Appl. Mech. Eng.* 361 (2020) 112744.
- [62] F. Aldakheel, N. Noii, T. Wick, Thomas P. Wriggers, A global–local approach for hydraulic phase-field fracture in poroelastic media, *Comput. Math. Appl.* (2020), <https://doi.org/10.1016/j.camwa.2020.07.013>.
- [63] J.-H. Yeon, S.-K. Youn, Meshfree analysis of softening elastoplastic solids using variational multiscale method, *Int. J. Solids Struct.* 42 (14) (2005) 4030–4057.
- [64] J.-H. Yeon, S.-K. Youn, Variational multiscale analysis of elastoplastic deformation using meshfree approximation, *Int. J. Solids Struct.* 45 (17) (2008) 4709–4724.
- [65] L. Zhang, J. Ouyang, X. Wang, X. Zhang, Variational multiscale element-free Galerkin method for 2D Burgers' equation, *J. Comput. Phys.* 229 (19) (2010) 7147–7161.
- [66] J.-G. Liu, C. Wang, H. Johnston, A fourth order scheme for incompressible Boussinesq equations, *J. Sci. Comput.* 18 (2) (2003) 253–285.