International Journal of Heat and Mass Transfer 139 (2019) 613–631

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

journal homepage: www.elsevier.com/locate/ijhmt

Uncertainty quantification in three dimensional natural convection using polynomial chaos expansion and deep neural networks

Shantanu Shahane *, Narayana R. Aluru, Surya Pratap Vanka

Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States

ARTICLE INFO

Article history:
Received 26 December 2018
Received in revised form 4 May 2019
Accepted 6 May 2019

Keywords:
Deep neural networks
Polynomial chaos expansion
Natural convection
Uncertainty quantification

ABSTRACT

This paper analyzes the effects of input uncertainties on the outputs of a three dimensional natural convection problem in a differentially heated cubical enclosure. Two different cases are considered for parameter uncertainty propagation and global sensitivity analysis. In case A, stochastic variation is introduced in the two non-dimensional parameters (Rayleigh and Prandtl numbers) with an assumption that the boundary temperature is uniform. Being a two dimensional stochastic problem, the polynomial chaos expansion (PCE) method is used as a surrogate model. Case B deals with non-uniform stochasticity in the boundary temperature. Instead of the traditional Gaussian process model with the Karhunen-Loève expansion, a novel approach is successfully implemented to model uncertainty in the boundary condition. The boundary is divided into multiple domains and the temperature imposed on each domain is assumed to be an independent and identically distributed (i.i.d) random variable. Deep neural networks are trained with the boundary temperatures as inputs and Nusselt number, internal temperature or velocities as outputs. The number of domains which is essentially the stochastic dimension is 4, 8, 16 or 32. Rigorous training and testing process shows that the neural network is able to approximate the outputs to a reasonable accuracy. For a high stochastic dimension such as 32, it is computationally expensive to fit the PCE. This paper demonstrates a novel way of using the deep neural network as a surrogate modeling method for uncertainty quantification with the number of simulations much fewer than that required for fitting the PCE, thus, saving the computational cost.

1. Introduction

Flow due to natural convection has been studied extensively in the literature [1–7] since it is practically useful in cooling or heating systems for applications like electronics, nuclear reactors, computing servers, etc. [8–10]. De Vahl Davis [1], Shu et al. [6] and Le Quéré [3] simulated natural convection in a two dimensional differentially heated square cavity with gravity in a direction orthogonal to the applied temperature difference. De Vahl Davis [1] used the stream function-vorticity formulation for laminar flow. Accurate benchmark solution was obtained using mesh refinement and extrapolation. Le Quéré [3] used a pseudo-spectral algorithm combining spatial expressions of Chebyshev polynomial series with a finite difference time marching scheme. The results for Rayleigh number up to 10^8 are presented. Shu et al. [6] solved the same problem with local radial basis function based differential quadrature (RDF-DQ) method. This method is a mesh-free approach with the RBFs as test functions to estimate the derivatives at any node as a weighted sum of values at the neighboring nodes. They discussed the effects of the RBF shape parameter and its fine tuning to get accurate solutions. Fusegi et al. [2] presented the results for three dimensional differentially heated cubical enclosure. They used the finite difference discretization with SIMPLE algorithm [11] for laminar flow at Rayleigh numbers in the range of 10^3 to 10^6. Rayleigh-Bénard is another class of natural convection problems in which the temperature difference is applied parallel to the direction of gravity with the lower wall heated and upper wall cooled. Hu et al. [4], Li et al. [5] and Yigit et al. [7] studied the Rayleigh-Bénard convection numerically in two and three dimensional cavities with cubical and cylindrical shapes having various aspect ratios.

The numerical and experimental results of natural convection are utilized for verification and validation of the numerical software packages. Due to inaccuracies in the measurement and control, the experimental results are prone to errors. These errors can be estimated by introducing stochastic variations in the inputs and propagating these to the outputs. It is possible that the stochastic mean of an output parameter is different from its value at the input mean. Deterministic simulations alone cannot

* Corresponding author.
E-mail address: shahaneshantanu@gmail.com (S. Shahane).

https://doi.org/10.1016/j.ijheatmasstransfer.2019.05.014
0017-9310/

estimate the shift of mean. Thus in the recent years, there has been a growing interest in analysis of the effects of stochastic variations in the inputs on the outputs. There are multiple examples in the literature in which the uncertainty propagation techniques are combined with the deterministic numerical simulations [12–23]. It is popular to use the polynomial chaos expansion (PCE) for uncertainty propagation in which the output is approximated as a summation of polynomial basis which are functions of the stochastic inputs. The two main classes of methods to estimate the coefficients of the PCE are stochastic Galerkin projection [12–15] and collocation [18,19,17]. Stochastic Galerkin method requires modification of the underlying deterministic code since it requires solution of a new set of equations and thus, is called as intrusive method. This becomes a significant additional effort of software development and it is difficult to couple with the legacy codes. Hence, recently non-intrusive stochastic collocation methods have gained popularity. The basic idea is to have multiple evaluations of the deterministic simulation at predefined collocation points which are samples from the underlying probability distribution function of the input parameters. The PCE coefficients are then estimated from the output values obtained from the deterministic solution at these input samples. The coefficients can be used for post-processing operations like output statistics estimation, response surface plotting and sensitivity analysis.

Uncertainty quantification (UQ) for various types of natural convection problems has been studied in the literature [15–17,20]. Maitre et al. [15] used the zero-Mach-number model to simulate natural convection in a two dimensional differentially heated square cavity with uncertainty in the cold wall temperature. The random component of the cold wall temperature is modeled using the Gaussian process with an auto-correlation function which is approximated by the truncated Karhunen-Loève (KL) expansion. PCE coefficients are estimated by the stochastic Galerkin projection. Output statistics for various values of the non-Boussinesq parameter ϵ are presented. Ganap-athysubramanian and Zabaras [17] presented an adaptive refinement based approach for reducing the number of deterministic simulations in high dimensional stochastic simulations. The adaptive sampling method is applied to the two dimensional natural convection problem with random boundary condition which is modeled by the KL expansion method. Venturi et al. [16] studied the stability of the two dimensional Rayleigh-Bénard convection subject to stochastic boundary temperatures. The random boundary condition is assumed to be a non-uniform Gaussian random processes approximated by the KL expansion. It is found that the stochastic wall temperatures can extend the stability range of quasi-conduction states beyond the classical bifurcation point. Fajraoui et al. [20] analyzed the natural convection of porous media in a two dimensional differentially heated square cavity with uncertainty in the Rayleigh number, permeability anisotropy ratio, dispersion coefficients and heterogeneity variation. PCE method is used to estimate the statistics and sensitivity of the output parameters such as temperature and Nusselt number distributions.

The PCE method is extremely useful for low dimensional uncertainty quantification. But at higher dimensions, it faces the problem known as 'curse of dimensionality' i.e., for a linear increase in the stochastic dimensions, the number of samples grows exponentially. The Smolyak algorithm [24] addresses this problem to some extent by reducing the number of samples in high dimensions without compromising the interpolation accuracy. Even with the use of the Smolyak algorithm, number of samples of the order of 10^3-10^4 are required for five or more input dimensions. For instance, an eight and sixteen dimensional problem needs 3905 and 51073 samples respectively, for the accuracy level of five [25]. Practically, it is computationally expensive to simulate the deterministic solution thousands of times. Thus, an alternate method is required for uncertainty propagation. The Monte Carlo method is a simple approach which approximates the statistics of the output by running the deterministic simulations at pseudo random samples of the inputs [26]. Since the error using the Monte Carlo method is $\mathcal{O}(\frac{1}{\sqrt{n}})$, the number of samples is practically too high which makes using the Monte Carlo method directly with the deterministic simulation difficult. Thus, it is popular to use a surrogate model which is trained and tested using deterministic simulations. A good surrogate model can be trained with a small number of deterministic simulations and its evaluation is cheap. A well tested surrogate model is further used to estimate the outputs at multiple sample inputs. Since the surrogate model evaluation is cheap, there is practically no limit on the number of input samples for the Monte Carlo method. Note that the PCE is also a surrogate model which is ideal for low stochastic dimensions with a possibility of direct estimation of the output statistics without the use of the Monte Carlo method.

In order to simulate a high dimensional stochastic problem, a neural network (NN) is used as a surrogate model. Hornik et al. [27] showed that multilayer feed forward networks are universal approximators i.e., with mild assumptions on the underlying function to be approximated, the network can achieve any desired degree of accuracy by choosing suitable number of neurons. The NNs can handle the 'curse of dimensionality' by multiple nonlinear activation functions. In recent years, the NNs have been extremely popular in many fields of work as discussed by the review paper by Schmidhuber [28]. Here, only the applications related to surrogate modeling for numerical simulations are discussed [29–41]. Sablani et al. [30] used a NN as a surrogate model for inverse heat conduction problem of estimation of heat transfer coefficient from the temperature-time history at different locations. Since the NN is trained for the inverse problem directly using forward deterministic simulations, the estimation can be done non-iteratively. Czél et al. [32] similarly used NN for non-iterative estimation of heat capacity and temperature dependent thermal conductivity using the experimental transient temperature histories. A radial basis function type NN is trained using the numerical solution of the direct heat conduction problem. Both the above publications show that computational time is saved by non-iterative estimation due to the NN surrogate model coupled with the forward numerical simulations. Gholami et al. [29] trained a NN for a three dimensional two fluid flow in a 90° curved channel and compared both the numerical simulations and NN predictions with experimental data. It is reported that the NN model is reasonably accurate and significantly faster compared to the full numerical simulation. Tripathy and Bilionis [34] trained a NN to solve a steady state two dimensional diffusion process with spatially varying uncertainty in the diffusion coefficient. This uncertainty is modeled as a log normal random field with mean and covariance functions of the Gaussian random field which is approximated by the Karhunen-Loève (KL) expansion. Using the trained NN as a surrogate with diffusion coefficient as the input, statistics of the output parameter are estimated. Zhang et al. [36] have shown the utility of the physics informed neural networks (PINNs) for uncertainty quantification in direct and inverse stochastic problems. The basic idea of a PINN is to minimize the residual when the NN is substituted in the model differential equation together with the standard loss function of the NN. Automatic differentiation is used to estimate the residual. The paper claims that minimizing the residual along with loss function enhances the accuracy of the prediction. Naphon et al. [37] studied the heat transfer and pressure drop in a spiral tube with magnetic field using neural network. They utilized four training algorithms to adjust the weights of the network so that the prediction error is minimized. The neural network is trained

with experimental data. Naphon et al. [38] analyzed the pressure drop and heat transfer for jet impingement of nanofluids using numerical simulations and neural network. The numerical analysis is done using Eulerian two phase model and the predictions are validated with experimental data. The neural network estimates of Nusselt number and pressure drop are found to be accurate within ±1.5%. Nabian and Meidani [39] have developed a meshfree framework for solving a high dimensional random partial differential equations using residual neural networks. They have demonstrated the framework on a stochastic Poisson equation on a square domain with central hole. Karumuri et al. [40] have also used a similar approach. Chan and Elsheikh [41] have coupled neural networks with a multiscale finite volume method to model a Poisson equation having stochasticity in the diffusion coefficient. They have found this coupled approach to be hundred times faster than the multiscale finite volume method.

This paper presents results of input uncertainty propagation for a three dimensional natural convection problem in a differentially heated cubical enclosure. Two different cases of input uncertainties are considered. Case A assumes that the boundary conditions are uniform. Thus, uncertainty is introduced in the two non-dimensional parameters (Rayleigh and Prandtl numbers). For this case, the polynomial chaos expansion (PCE) method is used as a surrogate model with stochastic collocation to estimate the PCE coefficients. Since this is a two dimensional stochastic problem, the number of samples required is small enough and thus, the estimation of the PCE coefficients is feasible. Case B deals with non-uniform stochastic boundary condition with deterministic material properties of the fluid. Since the temperature difference between the opposite walls drives the natural convection flow, the cold wall is held at a constant temperature and uncertainty is introduced in the hot wall temperature. Although the conventional method to deal with boundary condition uncertainties is to use the Gaussian process model with the Karhunen-Loève expansion [17,16,15,34], in this work, a novel approach is successfully implemented. The basic idea is to divide the hot wall into multiple domains and impose a temperature boundary condition on each domain. It is assumed that each domain temperature is an independent and identically distributed (i.i.d) random variable. Sets of forward deterministic simulations are used to train deep neural networks (DNNs) with the boundary temperatures as inputs and Nusselt number, internal temperature or velocities as outputs. The number of domains which is equal to the stochastic dimension ranges from 4 to 32. The DNN is successfully trained and tested with less number of samples compared to those required for PCE coefficient estimation for the high dimensional problem. This new approach to deal with stochastic boundary conditions with DNN as a surrogate model is found to be much better than those methods presented in the literature so far.

2. Deterministic problem description

2.1. Governing equations

In this paper, we consider the three dimensional natural convection in a differentially heated cube (Fig. 1). A temperature gradient is applied on two opposite faces ($X = 0$ and $X = 1$) of a cube with sides of length L. The remaining four faces are thermally insulated. Gravity is imposed in the Y direction which is orthogonal to the direction of the temperature gradient. Because of the thermal expansion of the fluid inside the cube due to the temperature variation, a buoyant force causes the lighter fluid to move upwards thus creating currents. The flow field can be described by three dimensional incompressible Navier-Stokes and energy equations. For moderate density variations, natural convection can be modeled using the Boussinesq approximation. The system of equations is written in terms of non-dimensional variables as follows [3]:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{Pr}{Ra^{0.5}} \nabla^2 \mathbf{u} - \nabla P - \hat{\mathbf{g}} Pr \Theta \tag{2}$$

$$\frac{\partial \Theta}{\partial t} + \nabla \cdot \mathbf{u}\Theta = \frac{1}{Ra^{0.5}} \nabla^2 \Theta \tag{3}$$

where $\mathbf{u}$ is the velocity vector, Θ is the temperature, t is time, $\hat{\mathbf{g}}$ is the unit vector in the direction of gravity, P is the pressure, $Pr = \frac{\nu}{\alpha}$ is the Prandtl number and $Ra = \frac{g\beta\Delta T L^3}{\nu\alpha}$ is the Rayleigh number. Characteristic values for non-dimensionalization are as follows: velocity $u_c = \left(\frac{\alpha}{L}\right) Ra^{0.5}$, time $t_c = \left(\frac{L^2}{\alpha}\right) Ra^{-0.5}$, pressure $P_c = \rho u_c^2$ and L is the cavity length. Non-dimensional temperature is defined as $\Theta = \frac{(T - T_m)}{(T_h - T_c)}$ where, T_h and T_c are hot and cold wall temperatures respectively and $T_m = \frac{(T_h + T_c)}{2}$ is the mean temperature.

2.2. Solution algorithm

The governing Eqs. (1)–(3) are solved using the software OpenCast [42] with finite volume method on a collocated grid. The fractional step method [43] is used to integrate the equations. An intermediate velocity field ($\mathbf{u}^*$) is first estimated by solving the modified momentum Eq. (4) without the pressure gradient. The diffusion term is discretized implicitly using second order Crank-Nicolson method whereas an explicit second order Adams-Bashforth is used for the convection term.

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -Conv(\mathbf{u}^n, \mathbf{u}^{n-1}) + Diff(\mathbf{u}^*, \mathbf{u}^n) + Buoy(T^n) \tag{4}$$

The velocity correction Eq. (5) is obtained by subtracting the modified momentum equation from the original equation.

$$\mathbf{u}^{n+1} = \mathbf{u}^* - (\nabla \Phi)^{n+1} \frac{\Delta t}{\rho} \tag{5}$$

Imposing divergence free condition on the $(n+1)$ velocity field gives the Poisson equation for Φ(6).

$$\nabla \cdot \left(\frac{\nabla \Phi}{\rho}\right)^{n+1} = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \tag{6}$$

The overall algorithm for marching from time step n to $n+1$ can be summarized as follows:

1. Solve for $\mathbf{u}^*$ using Eq. (5)
2. Solve the Poisson equation for Φ (Eq. (6)) iteratively to estimate Φ^{n+1}
3. Correct the velocities ($\mathbf{u}^{n+1}$) using Eq. (6))
4. Solve the energy Eq. (6)).

Note that the pressure P can be estimated from Φ if required:

$$P = \Phi - \mu \frac{\Delta t}{\rho} \nabla^2 \Phi \tag{7}$$

3. Grid independence study and verification

The numerical results from the software are verified using the three dimensional natural convection simulations of Fusegi et al. [2]. For Rayleigh number up to 10^6, the flow remains laminar [2].

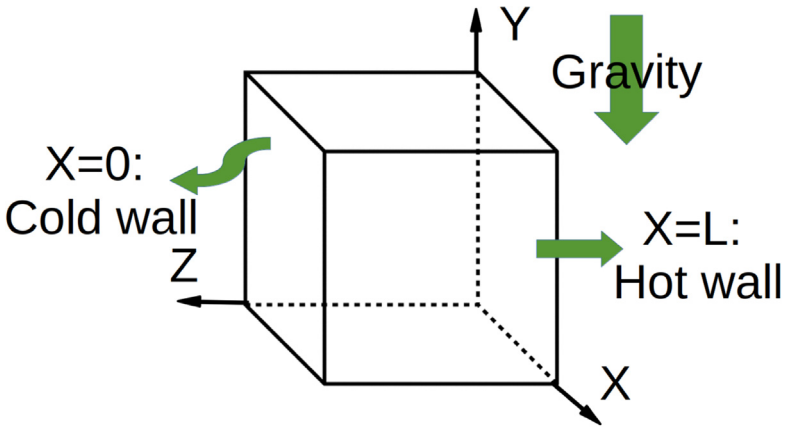


Fig. 1. Differentially heated cube schematic.

Hence Rayleigh numbers of 10^5 and 10^6 are used for verification. The cube is meshed with three different grid sizes with $32^3, 64^3$ and 128^3 structured hexahedrons. Steady state solution is estimated by time marching. For all the scalar fields ϕ, when the non-dimensional steady state error computed over the entire domain defined as $\frac{max(|\phi^{new}-\phi^{old}|)}{max(|\phi^{new}|)}$ is less than 10^{-4}, it is assumed that the steady state is reached. The temperatures at boundary faces $X = 0$ and $X = 1$ are set to 0.95 and 1.05 respectively. Note that the actual value of temperature difference is immaterial as the non-dimensional equation involves the Rayleigh number. In this work, we have fixed the temperature difference to 0.1 and the value of the coefficient of thermal expansion (β) is tuned in order to set the required Rayleigh number. The temperature and velocities are plotted along a centerline for both the Rayleigh numbers (Fig. 2). Note that the characteristic velocity used by Fusegi et al. [2] is different compared to the discussion in Section 2.1. Hence only for verification, the velocities are scaled by $u_c = \sqrt{g\beta L(T_h - T_c)}$. In each figure, estimates from the three grid levels computed by OpenCast [42] are superimposed with the results from Fusegi et al. [2] whenever available. The numerical estimates from OpenCast match well with the published results thus, verifying the code. For the $Ra = 10^5$ case (Fig. 2(a), (c) and (e)), it can be seen that all the three grid results from OpenCast overlap with each other. For $Ra = 10^6$ (Fig. 2(b), (d) and(f)), the coarsest grid (32^3) plot is slightly off but the remaining two finer grid plots overlap. This shows that grid independence is achieved and for all further computations, a grid of size 101^3 is used.

4. Parameter uncertainty quantification

Engineering problems have uncertainties in input parameters due to difficulty in precise measurement and control. It is necessary to propagate the input stochasticity to the output. Let $w(\boldsymbol{x}, \xi)$ be a function which maps inputs to an output. Here, the vector $\boldsymbol{x}$ denotes all the deterministic inputs whereas, the vector ξ denotes all the stochastic parameters. It is assumed that the stochastic variable follows a known probability distribution: $\xi \sim f(\xi)$. The aim is to estimate the stochastic mean of the output $w(\boldsymbol{x}, \xi)$ defined as:

$$w_f(\boldsymbol{x}) = \int w(\boldsymbol{x}, \xi) f(\xi) d\xi \tag{8}$$

Since the function w cannot be expressed in a closed form, above integral has to be approximated numerically. Importance sampling based Monte Carlo approximation of the integral is given by [26]:

$$w_f(\boldsymbol{x}) \approx \frac{1}{n}\sum_{i=1}^{i=n} w(\boldsymbol{x}, \xi_i)\frac{f(\xi_i)}{p(\xi_i)} \tag{9}$$

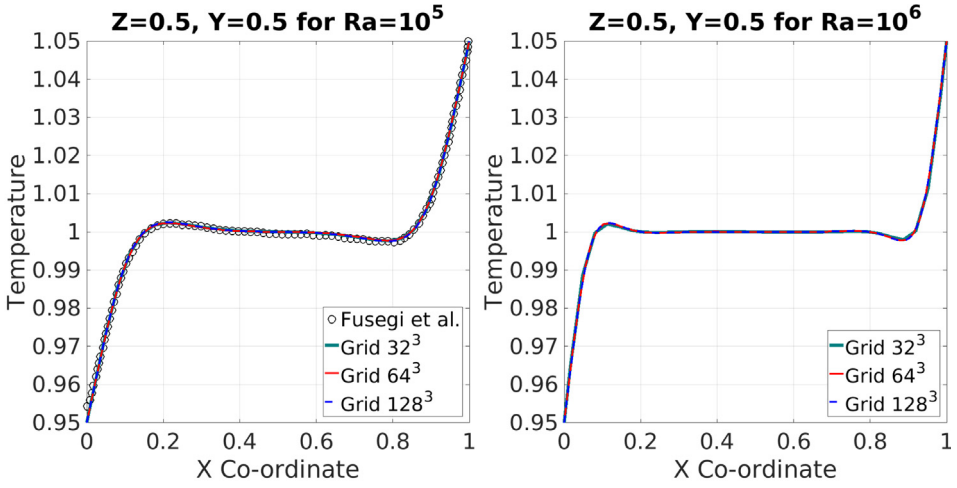
where ξ_i are n samples drawn from the probability distribution $p(\xi)$. It is effective to set $p(\xi)$ as $f(\xi)$ to reduce variance i.e., $\xi_i \sim f(\xi)$[26]. Thus, Eq. (9) is simplified to:

$$w_f(\boldsymbol{x}) \approx \frac{1}{n}\sum_{i=1}^{i=n} w(\boldsymbol{x}, \xi_i) \tag{10}$$

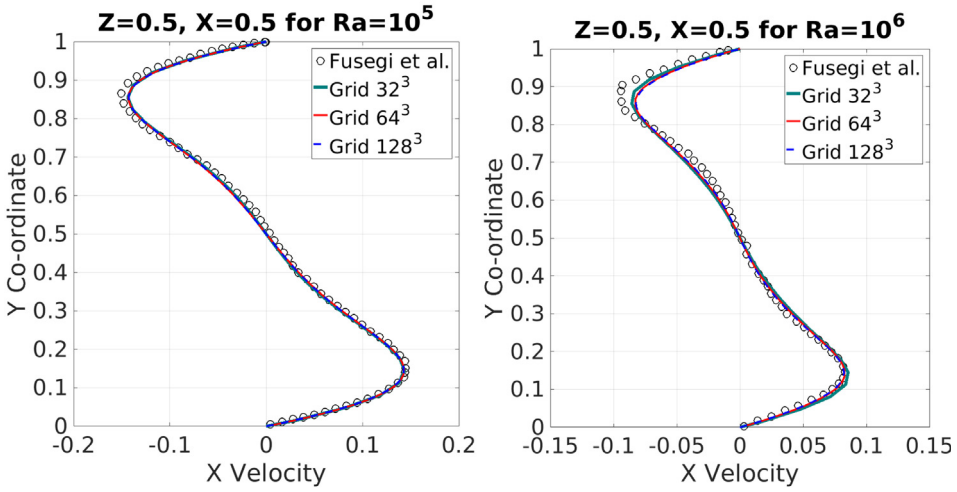
The error in the integral estimated using the Monte Carlo method is $\mathcal{O}(\frac{1}{\sqrt{n}})$. Thus, the number of samples (n in Eq. (10) can be of the order of thousands or more depending on the integrand. It is practically not possible to have so many samples as each sample corresponds to one deterministic simulation which is typically computationally costly. Thus, a surrogate model which is trained and tested using deterministic simulations is further used to estimate the outputs at multiple sample inputs. Since the surrogate model evaluation is cheap, there is practically no limit on the number of input samples. The input parameters affecting the simulation of natural convection are boundary temperature, domain length and material properties like viscosity, thermal diffusivity and coefficient of thermal expansion. Two separate cases of input uncertainties are analyzed in this paper. Case A assumes that the boundary temperature is uniform. Thus, the governing Eqs. (1)–(3) show that the physics can be parametrized just using Rayleigh and Prandtl numbers. The uniform boundary temperature, domain length and the material properties all are included in the Rayleigh and Prandtl numbers. Hence, from the perspective of parameter uncertainty propagation, this is a two dimensional problem. Case B considers the possibility of uncertainty in the non-uniform boundary temperature with deterministic material properties. Case A is analyzed using the polynomial chaos expansion (Section 4.1) whereas, deep neural networks (Section 4.2) are used as surrogate models for the case B. Such an analysis is practically important as there are stochastic variations in the boundary conditions due to inaccuracy in measurement and control. The fluid material properties also vary stochastically due to the presence of impurities. Hence, the effect of these uncertainties on the temperature and velocity distribution and Nusselt number is studied in this work. The following sections summarize the surrogate modeling strategies.

4.1. Case A: analysis of uncertainty in Rayleigh and Prandtl numbers

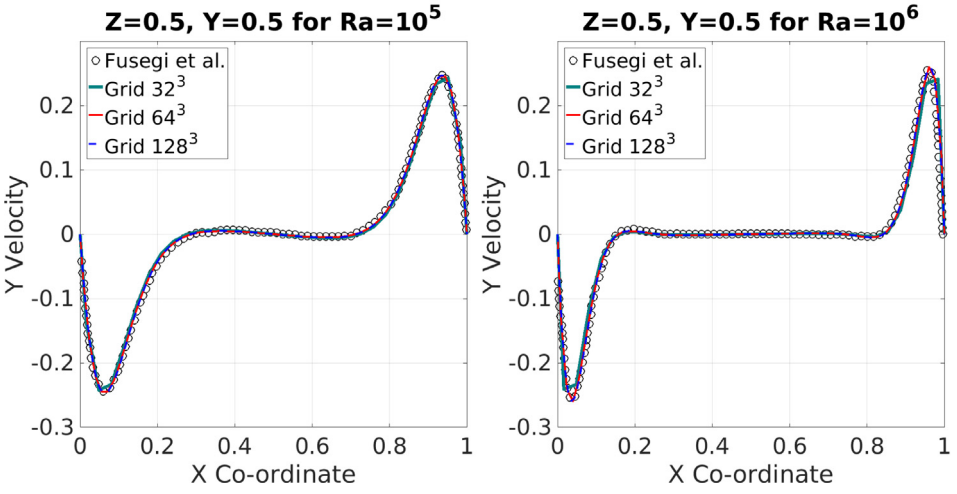
Literature on uncertainty quantification describes various methods to estimate the relationship between stochastic inputs and outputs. Most of these methods rely on the idea of expanding



(a) Temperature vs X: $Ra = 10^5$ (b) Temperature vs X: $Ra = 10^6$



(c) Y vs X Velocity: $Ra = 10^5$ (d) Y vs X Velocity: $Ra = 10^6$



(e) Y Velocity vs X: $Ra = 10^5$ (f) Y Velocity vs X: $Ra = 10^6$

Fig. 2. Temperature and velocity contours.

the outputs as a linear combination of polynomial basis functions in the stochastic dimension. Orthogonal polynomial is a popular choice as basis since orthogonality helps in convergence. Xiu and Karniadakis [44] showed that the Wiener's polynomial chaos [45] with orthogonal polynomials of Askey family lead to optimal convergence of the interpolation error. They identified which orthogo-

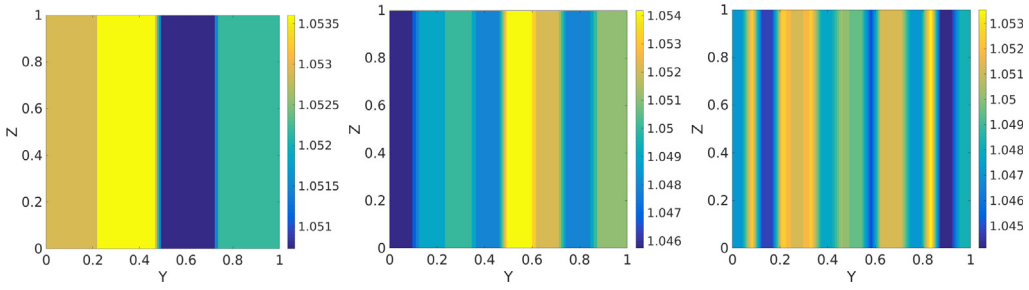
nal polynomial basis is suitable depending on the probability distribution function followed by the stochastic variable. For example the Hermite polynomials are orthogonal to each other when the standard normal distribution is used as weighting function. Thus, Hermite polynomial basis function is recommended when the stochastic variable follows normal distribution. A polynomial chaos series (Eq. (11)) is used to expand a second order random field [44]. The series is truncated to order n for all practical purposes.

$$w(\boldsymbol{x},\xi(\theta)) = \sum_{i=0}^{\infty} w_i(\boldsymbol{x})\boldsymbol{\Psi}_i(\xi(\theta)) \approx \sum_{i=0}^{n} w_i(\boldsymbol{x})\boldsymbol{\Psi}_i(\xi(\theta)) \tag{11}$$

where $\boldsymbol{x}$ is the spatial variable, $\xi = (\xi_1, \xi_2, \ldots \xi_r)$ is the random variable vector, θ is an elementary event, $\boldsymbol{\Psi}_i$ is a multi-dimensional orthogonal polynomial of order i, w is the output to be estimated and w_i are the coefficients of the series. In this case, ξ is a two dimensional vector following normal distribution and $\boldsymbol{\Psi}_i$ is a two dimensional Hermite polynomial. In this paper, stochastic collocation method is used to estimate the deterministic coefficients (w_i) of the polynomial chaos expansion. Collocation acts as a wrapper over the existing deterministic software since it is a non-intrusive method. Thus, no modification of the deterministic software is required. After deterministic simulations at M sample points (ξ^m), a constraint $w(\boldsymbol{x},\xi^m) = w_{sim}(\boldsymbol{x},\xi^m)$ is imposed. The left hand side is estimated from polynomial chaos expansion (Eq. (11)) and right hand side from each deterministic simulation. These M constraints can be written in the matrix vector form [46]. For accuracy, it is recommended to have more samples than the number of basis functions ($M > n+1$) and thus the Vandermonde system (Eq. (12)) is overdetermined.

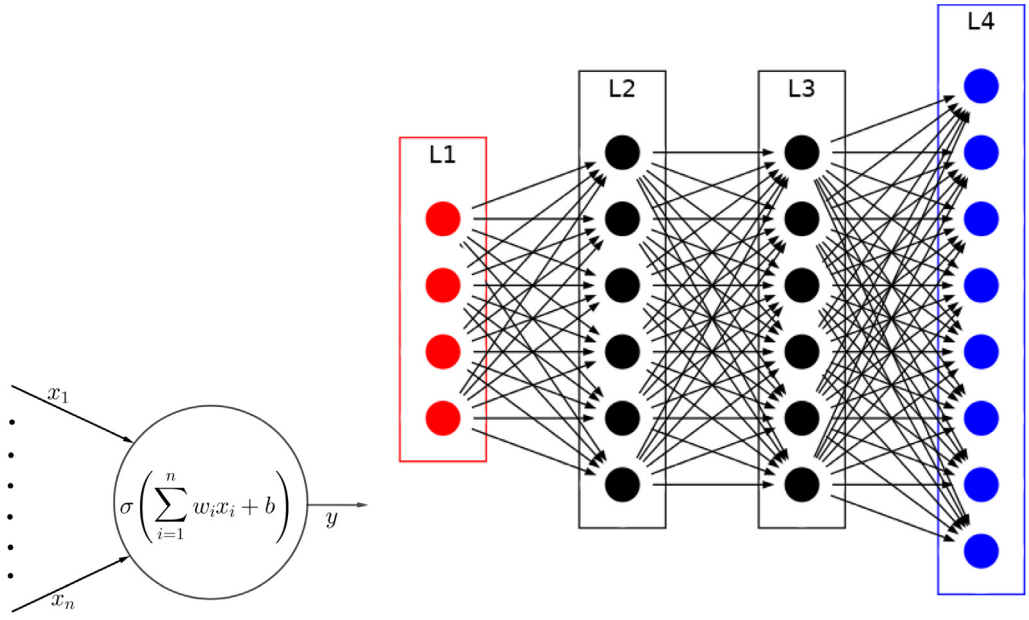
$$\begin{bmatrix} \boldsymbol{\Psi}_0(\xi^1) & \cdots & \boldsymbol{\Psi}_n(\xi^1) \\ \vdots & & \vdots \\ \boldsymbol{\Psi}_0(\xi^M) & \cdots & \boldsymbol{\Psi}_n(\xi^M) \end{bmatrix} \begin{bmatrix} w_0(\boldsymbol{x}) \\ \vdots \\ w_n(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} w_{sim}(\boldsymbol{x},\xi^1) \\ \vdots \\ w_{sim}(\boldsymbol{x},\xi^M) \end{bmatrix} \tag{12}$$

Sampling strategy plays a vital role in the accuracy and stability of stochastic collocation. Uniformly distributed samples lead to highly oscillatory interpolation and hence, poor convergence. Thus, for one dimensional stochastic problems, the roots of the basis orthogonal polynomials is a popular choice of sample points (ξ^m) [46]. For higher stochastic dimensions, tensor product of the single dimensional samples can be used. The sample size grows exponentially with dimensions if tensor product is used. This is a problem as each sample corresponds to a deterministic simulation and thus,



(a) Number of Strips: 4 (b) Number of Strips: 8 (c) Number of Strips: 16

Fig. 3. Samples of temperature boundary condition.



(a) Single Neuron (b) Deep Neural Network

Fig. 4. Neural network schematics.

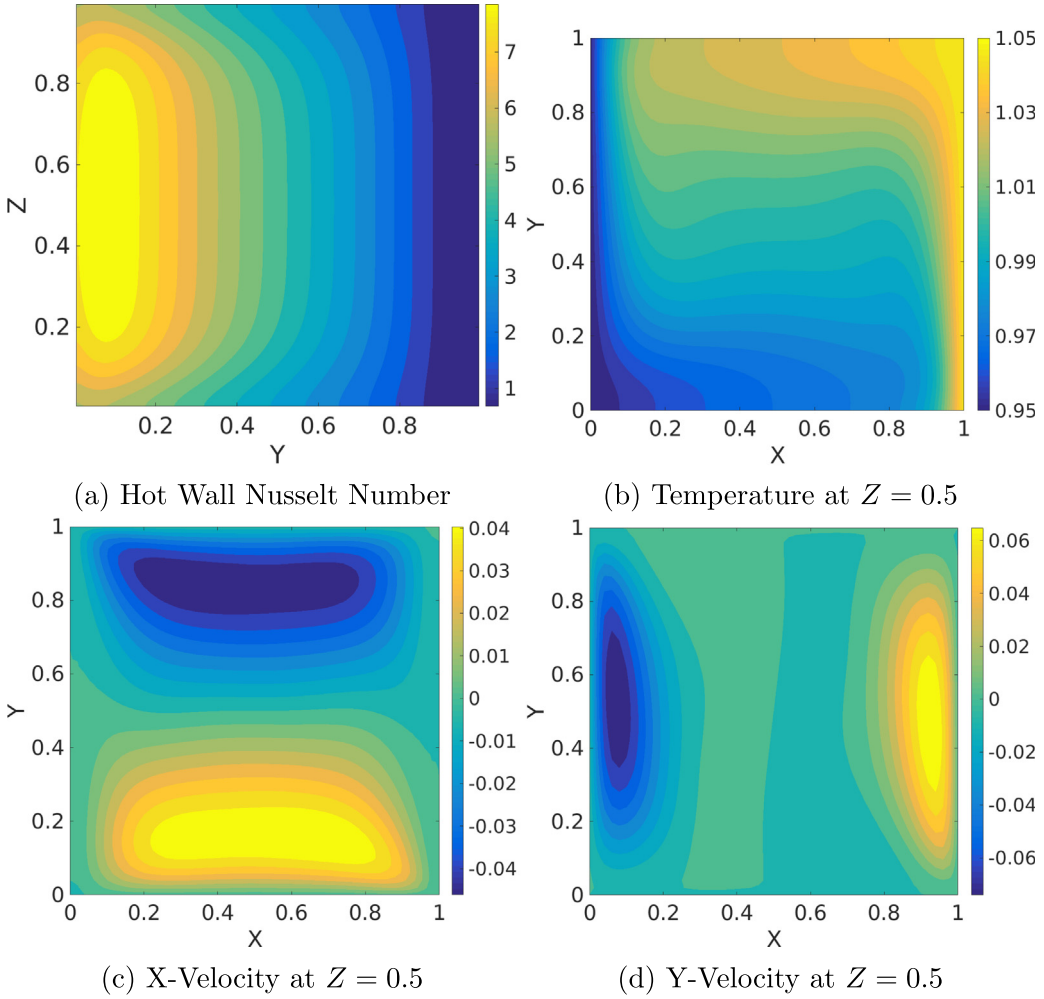
the cost of computations grows exponentially. An algorithm to reduce the number of samples in high dimensions without spoiling the interpolation accuracy was published by Smolyak [24]. It is found that for stochastic dimensions less than three, the Smolyak algorithm is not effective in reducing the sample size [25]. Thus, in this work, a tensor product of single dimensional samples (roots of the Hermite polynomials) are used [25]. UQLab, a MATLAB based tool developed by Marelli and Sudret [47] is used for estimation of polynomial chaos coefficients and response surfaces. The Polynomial Chaos-Kriging module of UQLab is used as it is found to be more effective than the basic polynomial chaos method.

4.2. Case B: analysis of uncertainty in boundary conditions

Practically, a heat exchanger setup with a closed loop feedback system is used to maintain the cold and hot wall temperatures. Due to the errors in measurement and control, there are stochastic variations in the set temperature. For a large wall, there would be multiple heat exchangers in contact with the wall. Since the objective is to maintain a uniform temperature, the design and operation of all the heat exchangers would be similar. Thus, it is safe to assume that the temperature achieved by each of them is a random variable following the normal distribution with mean as the expected temperature and error modeled as the standard deviation. It is also assumed that the heat exchangers are independently controlled. Thus, the set temperatures are independent and identically distributed (i.i.d) random variables. In the literature, there are examples of using Gaussian process model with the Karhunen-Loève expansion to model uncertainty in the boundary condition [17,16,15]. In the present work, the wall is subdivided into multiple domains with different values of temperatures imposed as boundary conditions. From Fig. 10, it can be seen that the variation of Nusselt number is stronger in the direction of gravity (Y) compared to the orthogonal direction (Z). Thus, the wall is divided into strips along gravity (Y). Fig. 3 shows samples of the temperature boundary condition with 4, 8 and 16 number of strips. Each strip temperature is assumed to be an i.i.d. random variable following a normal distribution ($\mu = 1.05, 3\sigma = 0.01$). A neural network is a set of interconnected nodes such that the information flows from inputs to outputs. Each node is known as a neuron. Fig. 4a shows a single neuron which has n scalar inputs ($x_1, x_2, \ldots, x_n$) and single output (y). Each neuron performs the following two operations in sequence:

1. Linear transformation: $a = \sum_{i=1}^{n} w_i x_i + b$; where, w_i are the weights and b is a bias term
2. Element-wise nonlinear transformation: $y = \sigma(a)$; where, σ is the activation function.

A neural network is formed by stacking single neurons in a layer and connecting multiple layers as shown in Fig. 4b. It depicts an input (layer L1), an output (layer L4) and two hidden layers (layers



(a) Hot Wall Nusselt Number (b) Temperature at $Z = 0.5$

(c) X-Velocity at $Z = 0.5$ (d) Y-Velocity at $Z = 0.5$

Fig. 5. Deterministic results for $Ra = 10^5$ and $Pr = 7.5$.

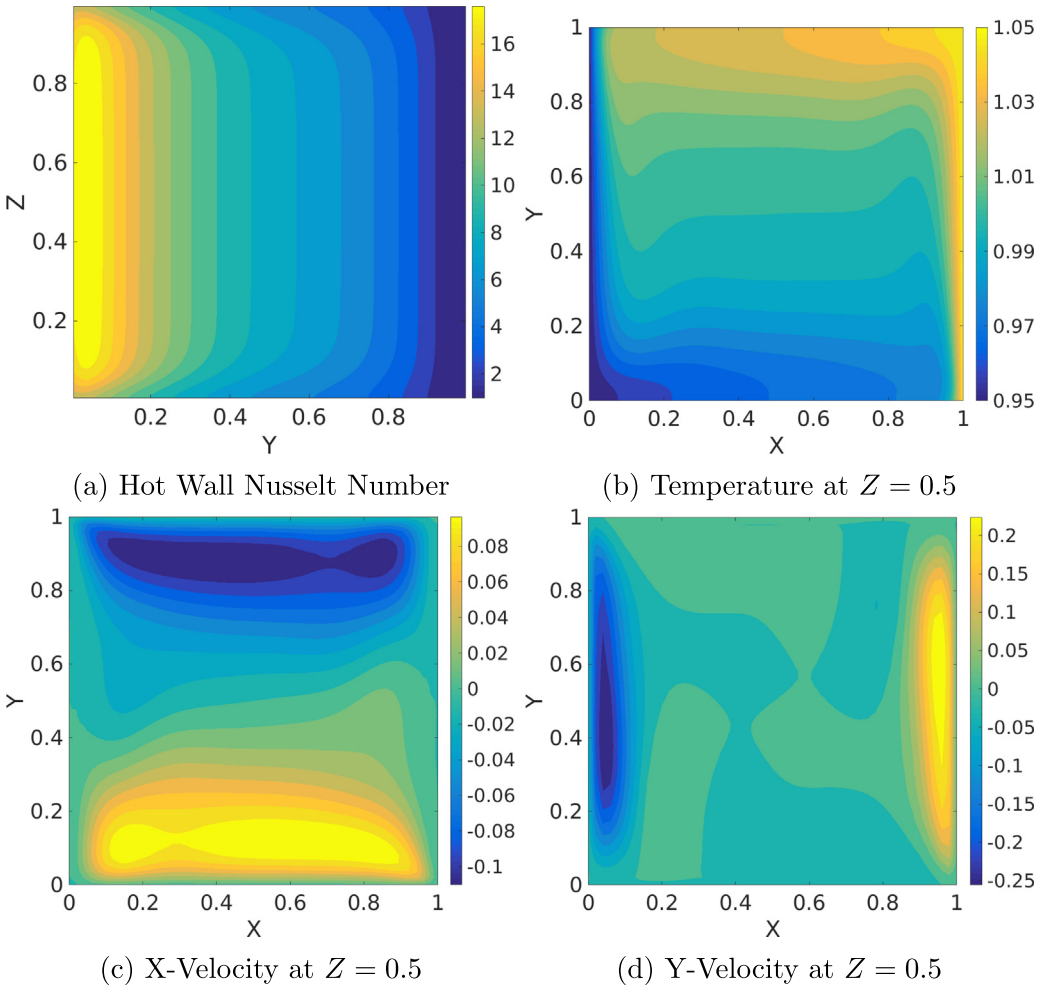


Fig. 6. Deterministic results for $Ra = 10^6$ and $Pr = 7.5$.

Table 1
Stochastic collocation error analysis.

Accuracy Level	# Samples	Ra 1E5	Ra 1E6
5	25	3.45E−05	1.32E−03
6	36	6.76E−05	5.24E−04
7	49	1.79E−05	2.50E−04

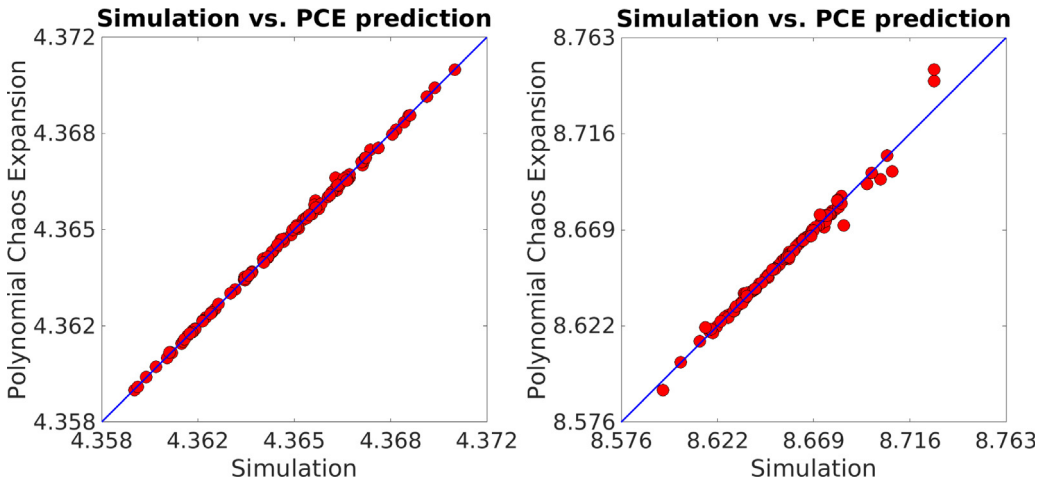
L2 and L3). The arrows indicate the direction of information flow from input to output layer through the hidden layers. A deep neural network (DNN) is essentially a neural network with multiple hidden layers. Adding multiple hidden layers increases the nonlinearity of the network and thus, the network can approximate more complex functions successfully. The number of neurons in the input and output layers is specified by the problem definition whereas, number of hidden layers and neurons has to be fine tuned. The linear transformation followed by the nonlinear activation function of each neuron can be written in a single matrix vector equation:

$$\boldsymbol{y}^{(j)} = \begin{cases} \boldsymbol{x} & \text{if } j = 1 \\ \sigma(\boldsymbol{W}^{(j)}\boldsymbol{y}^{(j-1)} + \boldsymbol{b}^{(j)}) & \forall\, j \in \{2, 3, \ldots, L\} \end{cases} \tag{13}$$

where $\boldsymbol{x} \in \mathbb{R}^{l_1}$ is the input vector, $\boldsymbol{W}^{(j)} \in \mathbb{R}^{l_j \times l_{j-1}}$ is the matrix of the weights, $\boldsymbol{y}^{(j)} \in \mathbb{R}^{l_j}$ is the activation produced by the j^{th} layer and $\boldsymbol{b}^{(j)} \in \mathbb{R}^{l_j}$ is the bias. L is the total number of layers including input, output and hidden layers. Number of neurons in the j^{th} layer is denoted by l_j. For instance, in Fig. 4b, $L = 4, l_1 = 4, l_4 = 8$ and $l_2 = l_3 = 6$. Applying Eq. (13) sequentially starting from the input layer is known as forward propagation. This operation estimates the output vector ($\boldsymbol{y}^{(L)}$) from the input vector ($\boldsymbol{y}^{(1)} = \boldsymbol{x}$) if the weights and bias are known. Logistic sigmoid, hyperbolic tangent and rectified linear unit (ReLU) are some of the popular activation functions [48]. In this work, the ReLU function defined by $\sigma(y) = max\{0, y\}$ is used for all the hidden layers. For output layer, in order to allow negative values, the identity function $\sigma(y) = y$ is used. The process of estimation of weights and bias using a given set of inputs $(\boldsymbol{x}_i, 1 \leqslant i \leqslant m)$ and the corresponding outputs $(\boldsymbol{z}_i, 1 \leqslant i \leqslant m)$ is known as training. For the given set of m training samples, mean squared error between the neural network estimate $(\hat{\boldsymbol{z}}_i)$ and the true value $(\boldsymbol{z}_i)$ of the output is defined as the loss function:

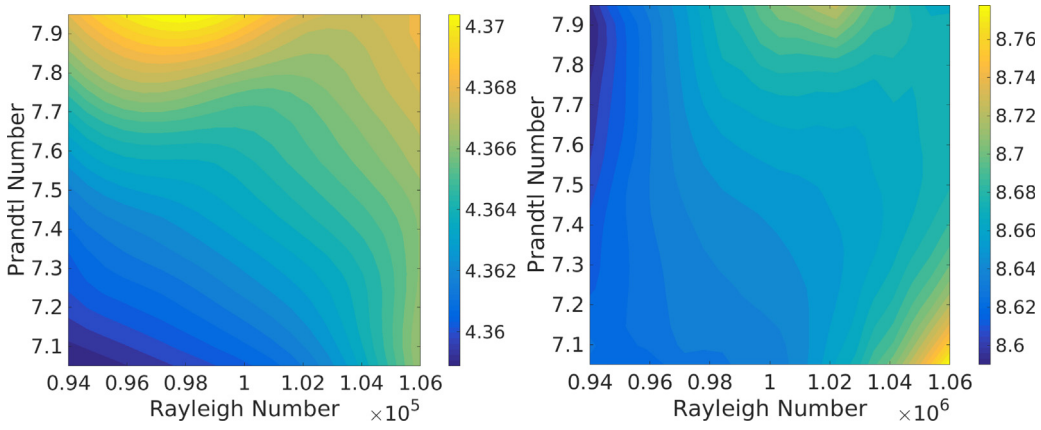
$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{b}; \boldsymbol{x}_i, \boldsymbol{z}_i) = \frac{1}{m}\sum_{i=1}^{i=m} ||\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i||_2^2 \tag{14}$$

It is commonly seen that the neural network performs well on the training data but performs poorly on the unseen test data. This phenomenon is known as overfitting and is controlled with regularization. Goodfellow et al. [48] discuss various regularization methods in detail. Here, the L2 weight regularization with parameter λ is used in which, the loss function (Eq. (14)) is modified:



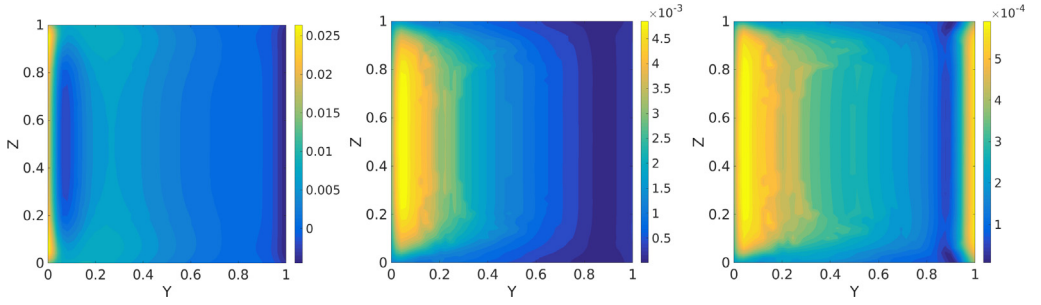
(a) $Ra = 10^5$ (b) $Ra = 10^6$

Fig. 7. Mean Nusselt number estimate from numerical simulation and polynomial chaos expansion.



(a) $Ra = 10^5$ (b) $Ra = 10^6$

Fig. 8. Spatial mean Nusselt number response surface.



(a) Diff. of Stochastic Mean and Deterministic Value (b) Stochastic Std. Dev. (c) Stochastic Std. Dev. by Stochastic Mean

Fig. 9. Local Nusselt number at hot wall for $Ra = 10^5$.

$$\mathcal{L}(\boldsymbol{W},\boldsymbol{b};\boldsymbol{x}_i,\boldsymbol{z}_i) = \frac{1}{m}\sum_{i=1}^{i=m}||\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i||_2^2 + \lambda\sum_{l=1}^{L}||\boldsymbol{W}^{(l)}||_2^2 \tag{15}$$

The weights and bias which minimize the loss function are estimated using a numerical optimization algorithm. The gradient of the loss function with respect to the weights and bias is required in the optimization algorithms like gradient descent. The gradient is estimated by the backpropagation algorithm [49]. The Adam optimizer is used in this work with the parameters β_1 and β_2 as suggested by Kingma and Ba [50]. Other hyper-parameters like learning rate, regularization constant, number of hidden layers and number of hidden units are tuned using a validation set. All the implementation details are given in Section 5.3.1.

5. Uncertainty propagation results

5.1. Deterministic results: output values at input mean

Uncertainty quantification analyzes the effects of small stochasticity in the input on the output. Since the stochasticity in the input is of the order of a small percentage of its mean, a similar variation is expected in the output. Thus, in this paper, a comparison of the stochastic mean of each output variable is done with the value of that variable at the input mean. The temperature and velocities are non-dimensionalized as discussed in Section 2.1. Figs. 5 and 6 plot the Nusselt number at hot wall and temperature, X and Y velocities along the Z midplane for Rayleigh number of 10^5 and 10^6, respectively. The stochastic means of the Nusselt number, temperature and velocities are expected to follow trends similar to the Figs. 5 and 6. Thus, in the following sections, contour plots of the difference between the output stochastic mean and the deterministic values are plotted for comparison. The difference gives an estimate of the effect of uncertainty in the input on the outputs compared to the case with deterministic inputs.

5.2. Case A

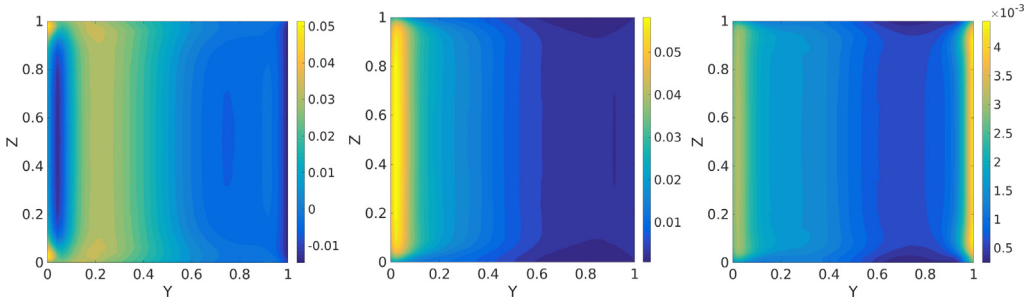
The non-dimensionalized governing Eqs. (1)–(3) show that the natural convection problem is parametrized by two parameters viz. the Rayleigh number and the Prandtl number. It is assumed that both of them follow a normal distribution with a 2% standard deviation with respect to mean:

- $Ra \sim \mathcal{N}(\mu = 10^5, \sigma = 0.02\mu)$ or $Ra \sim \mathcal{N}(\mu = 10^6, \sigma = 0.02\mu)$
- $Pr \sim \mathcal{N}(\mu = 7.5, \sigma = 0.02\mu)$.

Results from two different Rayleigh numbers (10^5 and 10^6) for which the flow is known to remain laminar are presented here [2]. High Rayleigh number implies higher buoyancy compared to viscous forces and thus, higher velocities and Nusselt number are observed. The fluid inside the cube is assumed to be water and hence, the Prandtl number is set to 7.5. As mentioned in Section 4.1, a tensor product of roots of Hermite polynomial scaled with mean and standard deviation is chosen as samples.

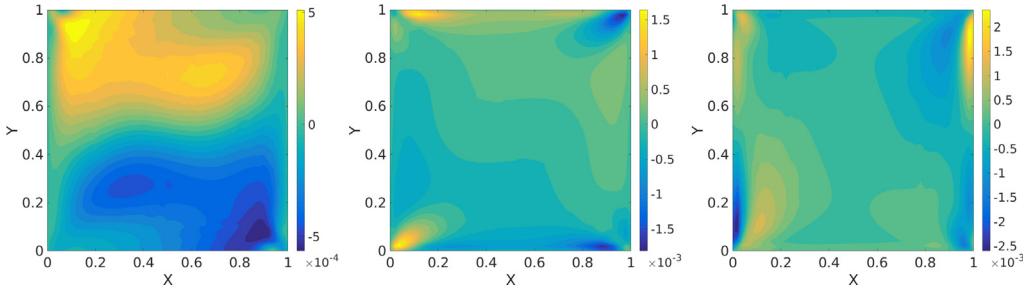
5.2.1. Convergence of the stochastic method

Hundred uniform Latin hypercube samples are used as test points to verify the convergence of the stochastic collocation



(a) Diff. of Stochastic Mean and Deterministic Value (b) Stochastic Std. Dev. (c) Stochastic Std. Dev. by Stochastic Mean

Fig. 10. Local Nusselt number at hot wall for $Ra = 10^6$.



(a) Temperature (b) X Velocity (c) Y Velocity

Fig. 11. Difference between stochastic mean and deterministic value at $Z = 0.5$ mid-plane for $Ra = 10^5$.

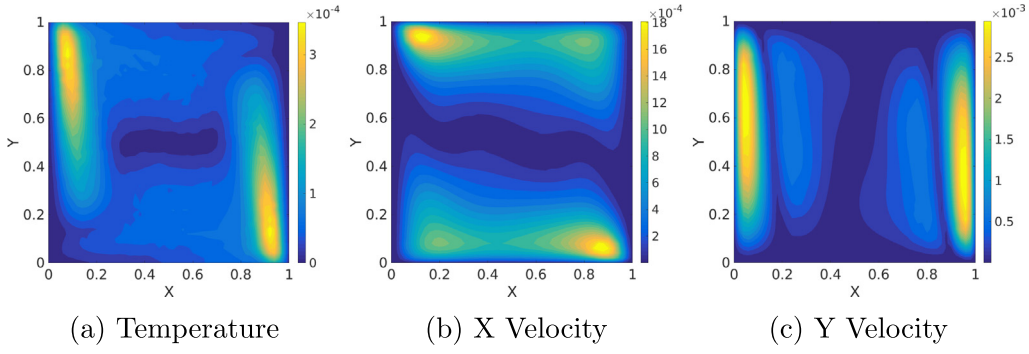


Fig. 12. Stochastic standard deviation at $Z = 0.5$ mid-plane for $Ra = 10^5$.

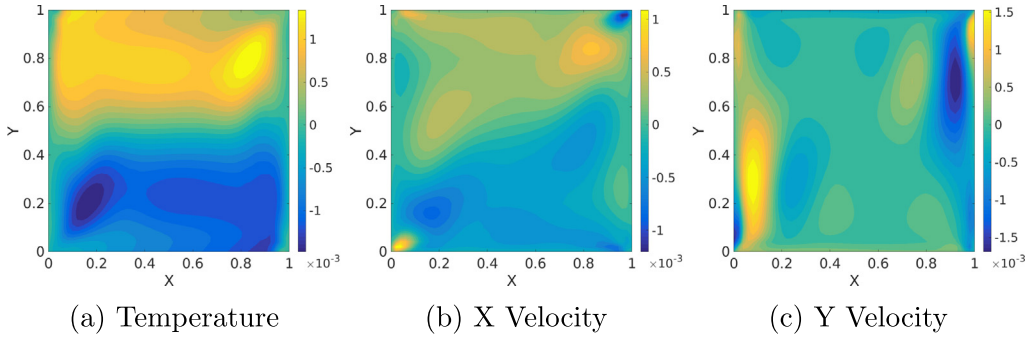


Fig. 13. Difference between stochastic mean and deterministic value at $Z = 0.5$ mid-plane for $Ra = 10^6$.

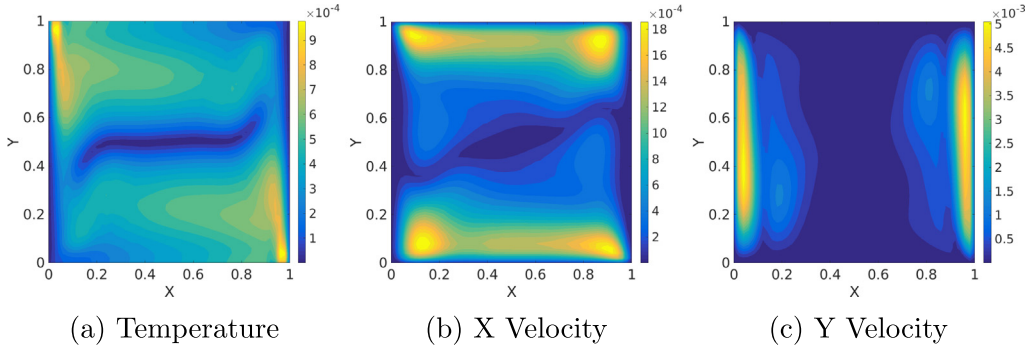


Fig. 14. Stochastic standard deviation at $Z = 0.5$ mid-plane for $Ra = 10^6$.

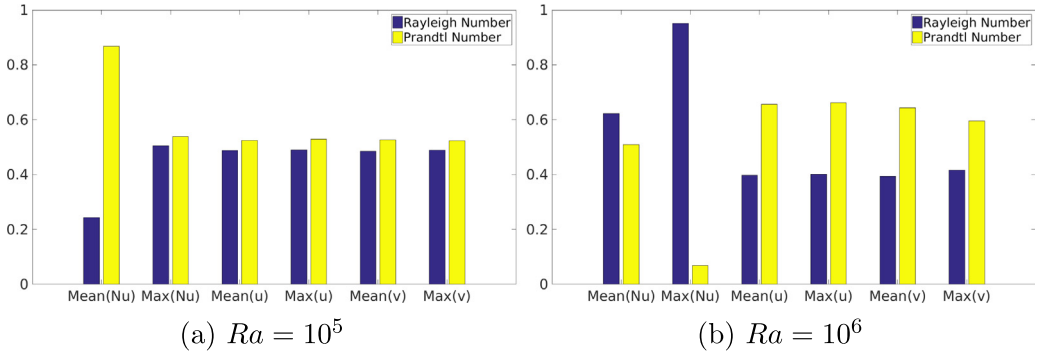


Fig. 15. Sensitivity: 6 outputs, 2 inputs.

method. Two independent estimates of the same output parameter are obtained using polynomial chaos expansion and deterministic simulation. The root mean square of the difference between these two estimates normalized by the maximum value of the parameter is defined as the non-dimensional error estimate. Spatial mean Nusselt number (Eq. (16)) over the hot face is used to estimate the collocation error.

$$Nu_{mean} = \int Nu(y,z)|_{x=1} dydz = \int \frac{\partial T(y,z)}{\partial x}|_{x=1} dydz \tag{16}$$

First column of Table 1 is the accuracy level of the sample points used for interpolation. Accuracy level l integrates polynomials up to degree $2l-1$ exactly [25]. Second column is the number of sample points i.e., the number of deterministic simulations required (M in Eq. (12)). The last two columns list the non-dimensional RMS error in computation of the spatial mean Nusselt number for both the Rayleigh numbers. Although the error increases slightly at the first two levels, eventually the error drops with higher accuracy level. This proves the convergence of the stochastic method. At the highest accuracy level, the error is of order 10^{-4} or 10^{-5} which shows that the polynomial chaos is reasonably accurate and can be used for further analysis. For visual inspection, spatial mean Nusselt number estimates from the numerical simulation and the polynomial chaos expansion are plotted together in Fig. 7 for the hundred test points. Ideally, all the points should lie on the $Y = X$ line but due to the stochastic interpolation error, some points are off the line. Since most of the points follow the expected trend of the $Y = X$ line, it can be concluded that the polynomial chaos is accurate.

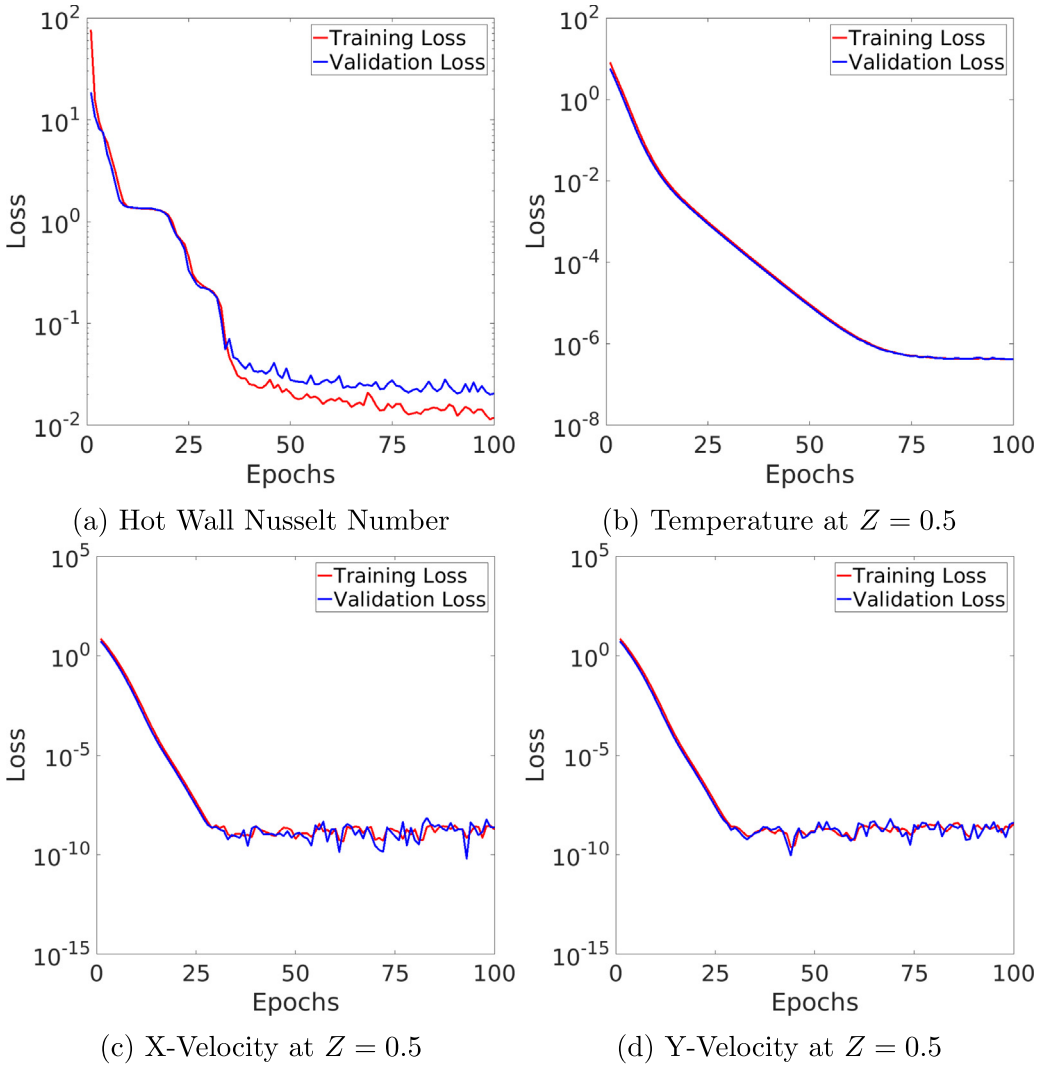


Fig. 16. DNN training and validation loss for 4 strips.

Table 2
Relative average percent error: DNN training and testing.

	4 Strips		8 Strips		16 Strips		32 Strips	
	Train	Test	Train	Test	Train	Test	Train	Test
Wall Nusselt No.	0.302	0.390	0.205	0.512	0.602	2.260	1.393	2.373
Temperature	0.029	0.030	0.024	0.025	0.022	0.023	0.022	0.024
X Velocity	1.114	1.144	0.864	0.903	0.704	0.731	0.644	0.722
Y Velocity	0.372	0.387	0.297	0.311	0.248	0.260	0.233	0.230

5.2.2. Nusselt number

Response surface gives a visual representation of the variation of an output parameter with input stochastic parameters. For a two dimensional stochastic problem, the response surface can be plotted as a contour. Fig. 8 plots the response surfaces of Nusselt number averaged over the hot wall (Eq. (16)) for both the Rayleigh numbers. In each plot, X and Y axes denote Rayleigh and Prandtl numbers, respectively which are the stochastic input parameters. Since both the input variables are assumed to follow normal distribution, they are plotted in the range $(\mu - 3\sigma, \mu + 3\sigma)$. For example, Rayleigh and Prandtl numbers are plotted in the range $(10^5 - 3 \times 2000, 10^5 + 3 \times 2000)$ and $(7.5 - 3 \times 0.15, 7.5 + 3 \times 0.15)$, respectively in Fig. 8a. The contour lines represent the value of mean Nusselt number. The slope of the contour line can be used to estimate the local sensitivity of the output with respect to a particular input. For example, the contour lines are nearly vertical in the left region of Fig. 8b which implies that the local sensitivity of the mean Nusselt number in the left region is high towards the input plotted on the X axis i.e., the Rayleigh number. The local Nusselt number on the wall ($Nu(y,z)$) varies due to the stochasticity in the Rayleigh and Prandtl numbers. Figs. 9 and 10 plot its statistics for both the Rayleigh numbers. The stochastic mean plots look visually similar to the plots at mean Rayleigh and Prandtl numbers (Figs. 5a and 6a). Thus, Figs. 9a and 10a plot contours of difference between the stochastic mean and deterministic value of the Nusselt number at hot wall (Figs. 5a and 6a). Figs. 9b and 10b plot the standard deviation due to the stochastic input variation. Higher standard deviation is observed in the region of higher mean. Thus, Figs. 9c and 10c are plotted to annihilate the effect of mean. The local ratio of standard deviation to mean shows higher values on the left and right sides near $Y = 0$ and $Y = 1$ as gravity is acting in Y direction. It can be seen that the difference between the stochastic mean and the deterministic values (Fig. 9a) is one order of magnitude higher than the stochastic standard deviation (Fig. 9b) for Rayleigh number of 10^5. This implies that the input stochasticity shifts the deterministic mean of the output more than its standard deviation. On the other hand, for Rayleigh number of 10^6, both the difference (Fig. 10)a and stochastic standard

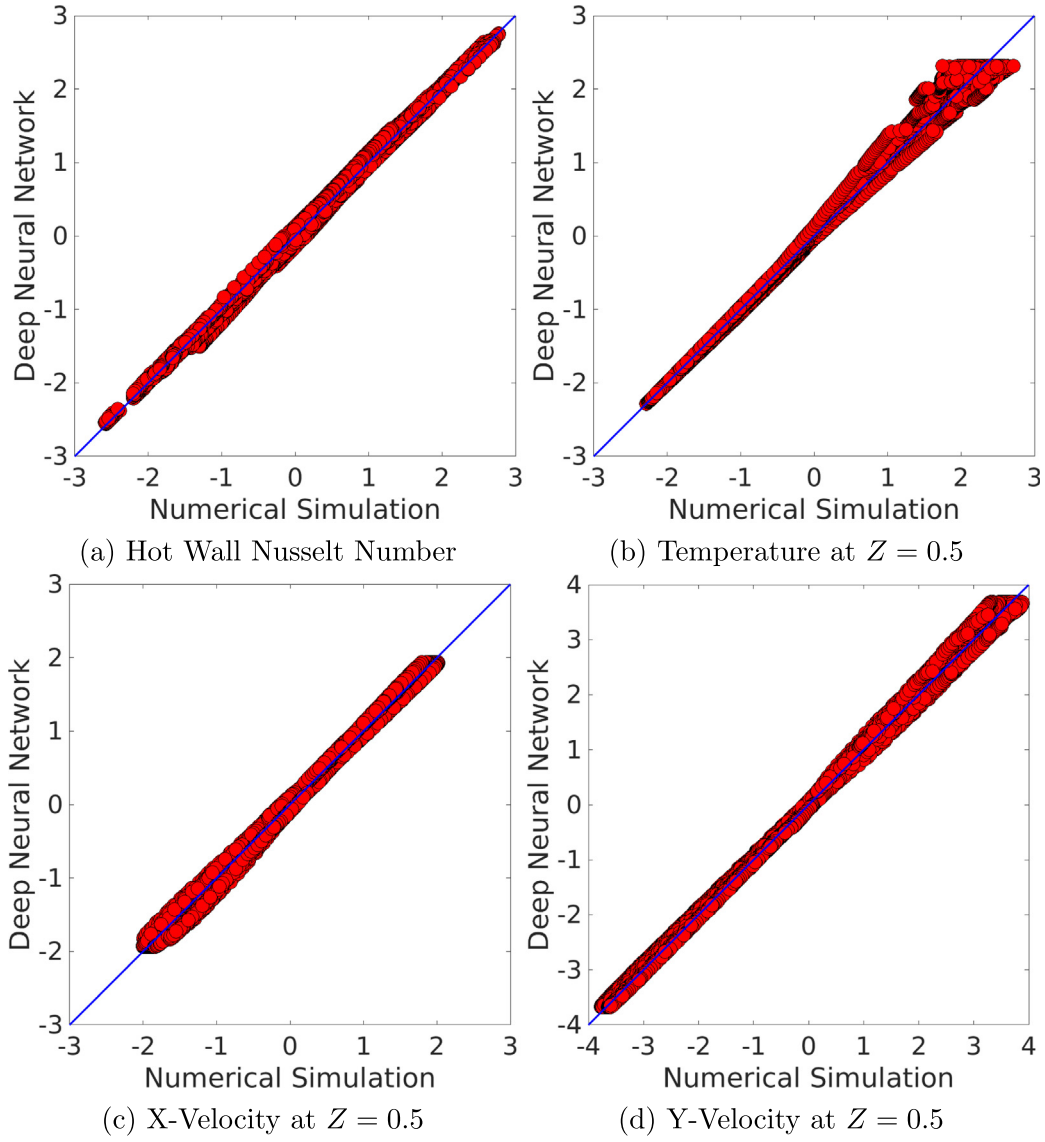


Fig. 17. Estimate from numerical simulation and DNN for 4 strips.

deviation (Fig. 10b) are of similar orders of magnitude. Thus, the shifting of mean and the standard deviation on the shift are of similar orders of magnitude for higher Rayleigh number. The difference between the stochastic mean and the deterministic value is of the order of 0.3% of the deterministic value for both the Rayleigh numbers.

5.2.3. Velocity and temperature

For the case of temperature gradient in the X direction and gravity in the Y direction, temperature and velocity contours at the Z mid-plane are quite informative. All the stochastic mean contour plots look visually similar to those of the deterministic natural convection problem (Figs. 5 and 6) and thus, are not plotted. Instead, the difference between the stochastic mean and deterministic values is plotted here. Figs. 11–14 plot the difference and standard deviation of temperature and velocities for the Rayleigh numbers of 10^5 and 10^6, respectively. It is observed that both the difference and standard deviations are of similar orders of magnitude. The difference and standard deviation in the temperature are three orders of magnitude smaller than the mean. On the other hand, the difference and standard deviation in velocities are two orders of magnitude smaller than the mean. Thus, it can be concluded that the effect of uncertainty is significant on the velocities than on the temperature. The uncertainty has higher impact on the temperature at $Ra = 10^6$ simulation than at $Ra = 10^5$. Similar to the Nusselt number, the standard deviation is higher when the mean value is higher. The difference between the stochastic mean and the deterministic value for temperature is of the order of 0.1—0.3% of the deterministic value for low and high Rayleigh numbers respectively. The difference is of the order of 1—3% for both the velocities of low and high Rayleigh numbers respectively. Thus, the effect of input stochasticity is higher on the velocities compared to temperature.

5.2.4. Sensitivity analysis

Sensitivity analysis quantifies the variation in an output due to the variation in a particular input. In this work, the global sensitivity is estimated using the Sobol indices based on the Sobol decomposition [51]. Partial Sobol index measures the contribution of a subset of inputs to the total variance. This includes the variance due to a coupling between the inputs. Total Sobol index for each input is defined as the sum of all the partial indices involving that input parameter. Here, the total Sobol indices are estimated from the polynomial chaos coefficients using the sensitivity analysis tool of the software UQLab [47]. Wall Nusselt number and X and Y velocities (u,v) have been chosen as representative outputs of the natural convection problem to study the sensitivity. The mean and maximum are taken over the hot wall and entire cube for the Nusselt number and velocities, respectively. Fig. 15 plots the sensitivity (total Sobol index) of the mean and maximum of Nusselt number and velocities with respect to each stochastic input parameter viz. Rayleigh and Prandtl numbers. It can be seen that the Nusselt number is more sensitive to Prandtl number for the case of $Ra = 10^5$ whereas, it is more sensitive to Rayleigh number for $Ra = 10^6$. On the other hand, the velocities are more sensitive to the Prandtl number in both the cases. But for the higher Rayleigh number case, their sensitivity towards the Prandtl number increases.

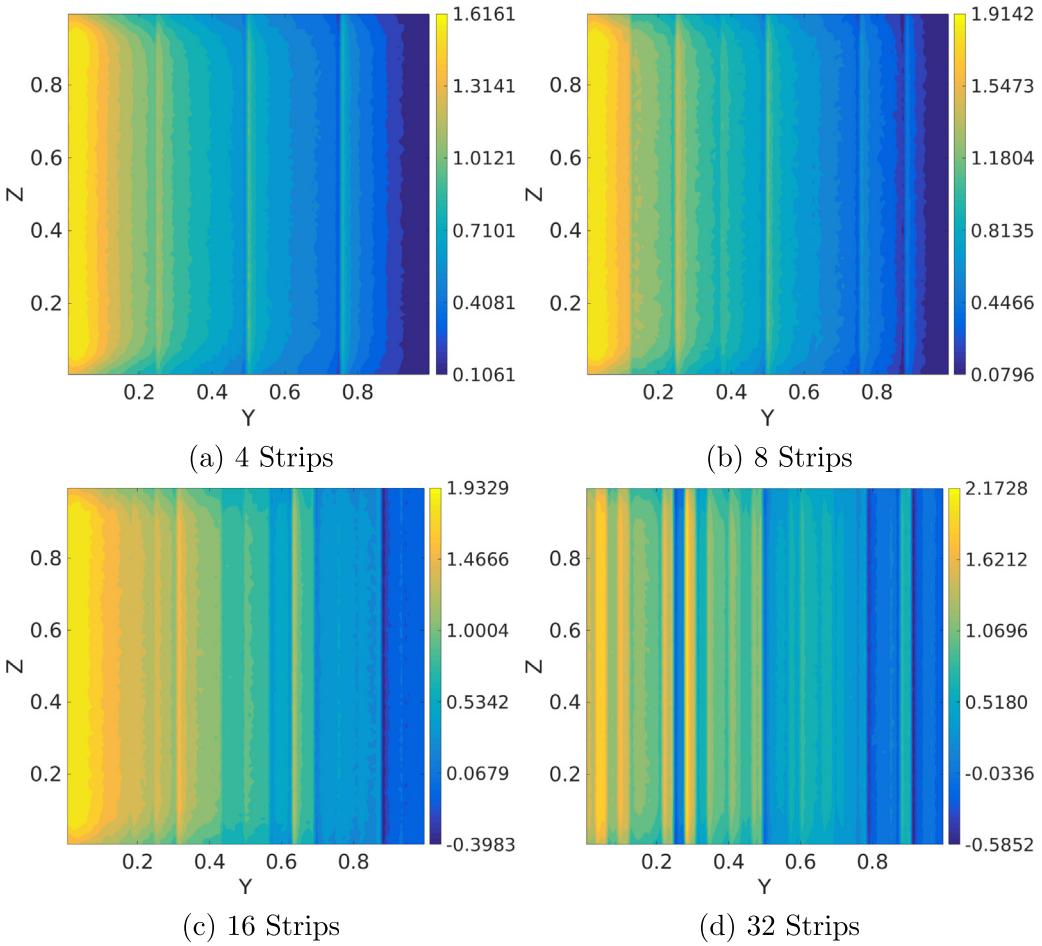


Fig. 18. $Ra = 10^6$ hot wall Nusselt number: difference between stochastic mean and deterministic value.

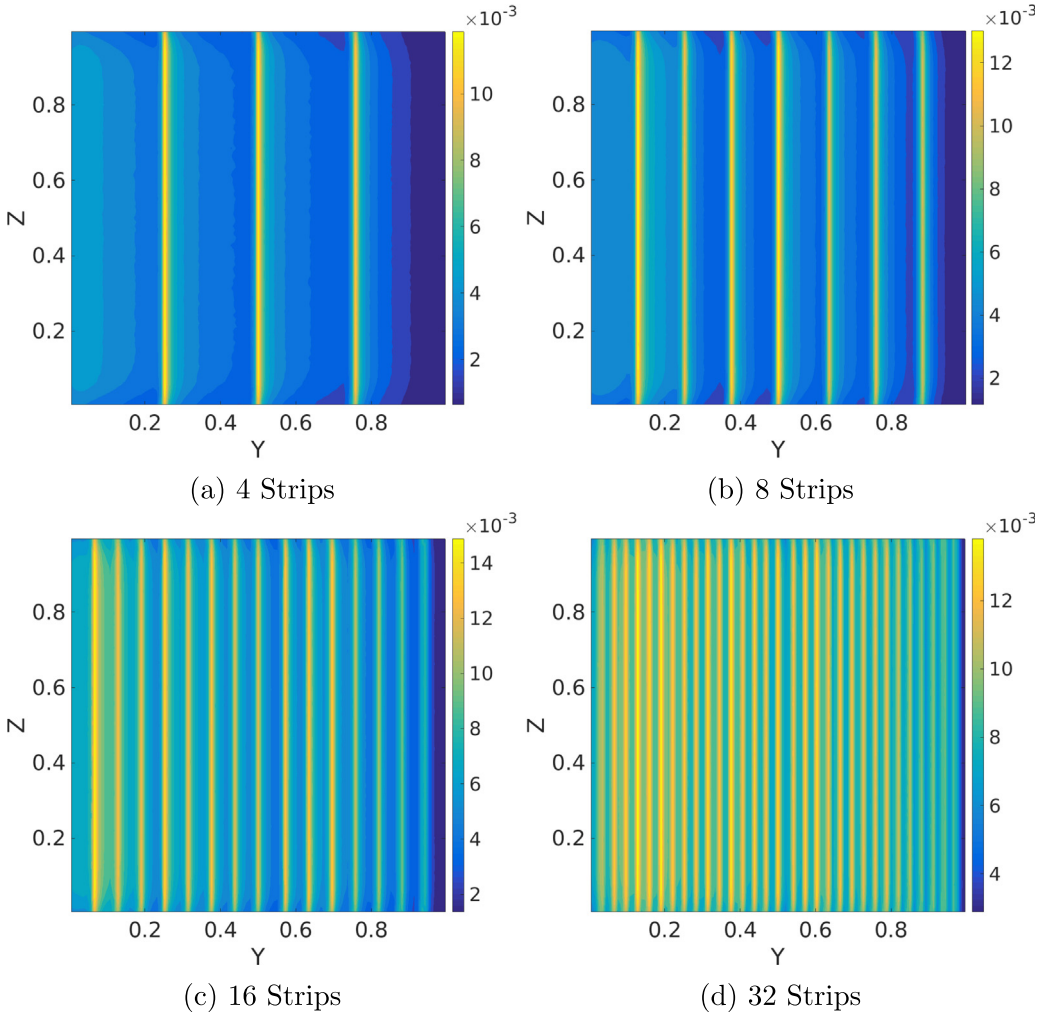


Fig. 19. $Ra = 10^6$ hot wall Nusselt number: stochastic standard deviation.

5.3. Case B

5.3.1. Deep neural network training and testing

As described briefly in Section 4.2, the hot wall is divided into strips along the direction of gravity Y (Fig. 3). Different boundary temperatures are prescribed on each strip. It is assumed that each strip temperature is an independent and identically distributed (i.i.d) random variable and follows a normal distribution with $\mu = 1.05$ and $3\sigma = 0.01$. Since the cold wall is held at a constant temperature of 0.95, the mean temperature difference driving the natural convection flow is still 0.1. This implies that a 3σ error of 10% is specified in the input stochasticity. The number of strips in this study is varied from 4 to 32 in multiples of 2. The material properties of the fluid are kept constant. In order to estimate the statistics of the outputs, a deep neural network (DNN) surrogate model is used. The boundary temperatures on each strip are inputs to the DNN. A 99^3 finite volume mesh is used for the numerical simulation. Four separate DNNs are trained with the following outputs:

1. Nusselt number along the hot wall ($99^2 = 9801$ outputs)
2. Temperature along the $Z = 0.5$ midplane ($99^2 = 9801$ outputs)
3. X velocity along the $Z = 0.5$ midplane ($99^2 = 9801$ outputs)
4. Y velocity along the $Z = 0.5$ midplane ($99^2 = 9801$ outputs).

Latin hypercube samples (LHS) are generated using the python package pyDOE [52]. The uniformly distributed LHS are transformed into normal distribution using the inverse cumulative distribution function (ppf) of the statistical functions module of scipy [53]. Separate sets of LHS are generated for training, validation and testing. Cases with 4, 8 and 16 strips are trained with 500 samples whereas, for 32 strips, 1000 samples are required. For each case, two different sets of 100 samples are used for validation and testing. The number of neurons in the input and output layers is specified by the number of inputs and outputs. The learning rate, regularization constant, optimizer and the number of hidden layers and neurons are highly problem specific and are chosen so that both the training and validation error are simultaneously minimized. The prediction accuracy is then checked on an unseen testing set. This overall procedure helps in fitting a DNN with low bias and low variance [48]. The DNNs are implemented in the Python library Tensorflow [54] with a high level API Keras [55]. Among the various optimizers available in Keras, the Adam optimizer [50] is found most suitable in this work. Settings of Adam optimizer are as follows: learning rate of $10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999$ and 'amsgrad' option switched on. ReLU and identity are the activation functions for all the hidden and output layers respectively. Other hyperparameters specific to each of the four DNNs are as follows:

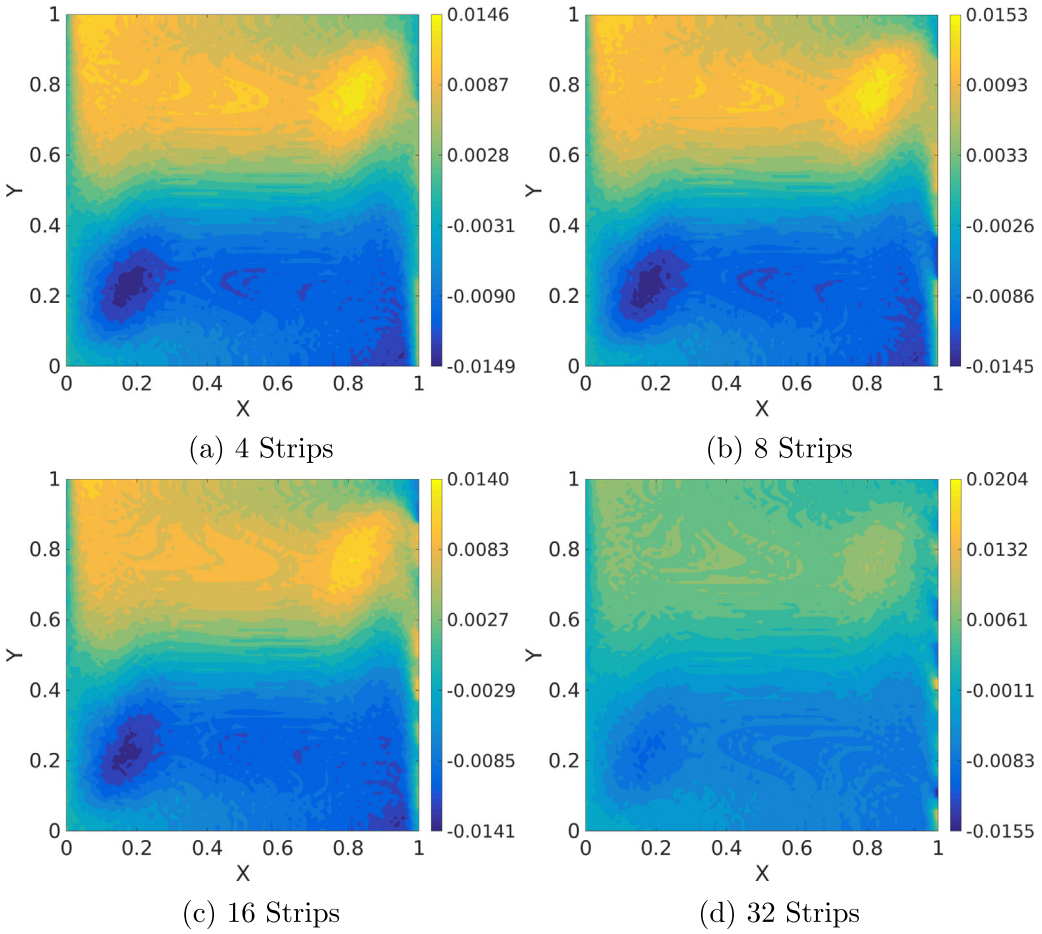


Fig. 20. $Ra = 10^6 Z$ midplane temperature: difference between stochastic mean and deterministic value.

1. Wall Nusselt number: $\lambda = 0.001, L_h = 5, n_h = 300$
2. Temperature: $\lambda = 0.001, L_h = 4, n_h = 300$
3. X velocity: $\lambda = 0.01, L_h = 4, n_h = 300$
4. Y velocity: $\lambda = 0.01, L_h = 4, n_h = 300$.

where λ is L_2 regularization constant, L_h is the number of hidden layers and n_h is the number of neurons in each hidden layer. All the DNNs are trained for 100 iterations on the entire dataset known as epochs. All the hyperparameters given above are tuned using the validation set with an objective to minimize bias and variance. Fig. 16 plots the loss versus epochs during training for each of the four DNNs for the case with 4 strips. Losses for DNNs of 8, 12 and 16 strips are similar and hence are not plotted here. Since both the training and validation losses are close to each other, it shows that the variance is low. Table 2 documents the training and testing errors for all the 16 DNNs: 4 DNNs each for 4, 8, 16, and 32 strips. The relative average percent error is defined as hundred times the L_1 norm of the difference between true values (numerical simulation) and DNN estimates divided by the maximum absolute value of the output. The training and testing errors are small and close enough thus, implying low bias and variance. For visual inspection, estimates from the numerical simulation and the DNN are plotted together in Fig. 17 for the testing samples for the case of 4 strips. Each output is normalized by subtracting its mean and dividing by its standard deviation and thus, is non-dimensional. Ideally, all the points should lie on the $Y = X$ line but due to the interpolation error, some points are off the line. Since most of the points follow the expected trend of the $Y = X$ line, it can be concluded that the neural network surrogate is accurate. The statistics of output parameters are estimated using the Monte Carlo method by evaluating the neural network on 100,000 random samples.

5.3.2. Nusselt number

Fig. 18 plots the difference between the stochastic mean and the deterministic value of the Nusselt number on the hot wall. Since this wall is subjected to the stochastic boundary condition, the demarcations of the strips can be seen. For example, Figs. 18a and b have four and eight strips respectively. Similar strips are also observed on the stochastic standard deviation contours (Fig. 19). The maximum deterministic value of the Nusselt number over the hot wall is 18.71 (Fig. 6a). The maximum values of the difference are 1.62, 1.91, 1.93 and 2.17 for 4, 8, 16 and 32 strips respectively. Hence, this shift in the stochastic mean from the deterministic value is comparable to the input uncertainty of 10%. On the other hand, the stochastic standard deviation is of the order of 10^{-2}. Thus, it is seen that the shift in mean is more pronounced than its deviation similar to case A.

5.3.3. Velocity and temperature

Figs. 20–22 plot the difference between the stochastic mean and the deterministic value of the temperature, X and Y velocities respectively along the $Z = 0.5$ midplane. The effect of the number of strips along the boundary condition is clearly visible in the Nusselt number plots (Figs. 18 and 19). On the other hand, the number of strips does not affect the temperature and velocity contours. The temperature and velocity contour plots for all the 4 cases look similar. The difference between the stochastic mean and the determin-

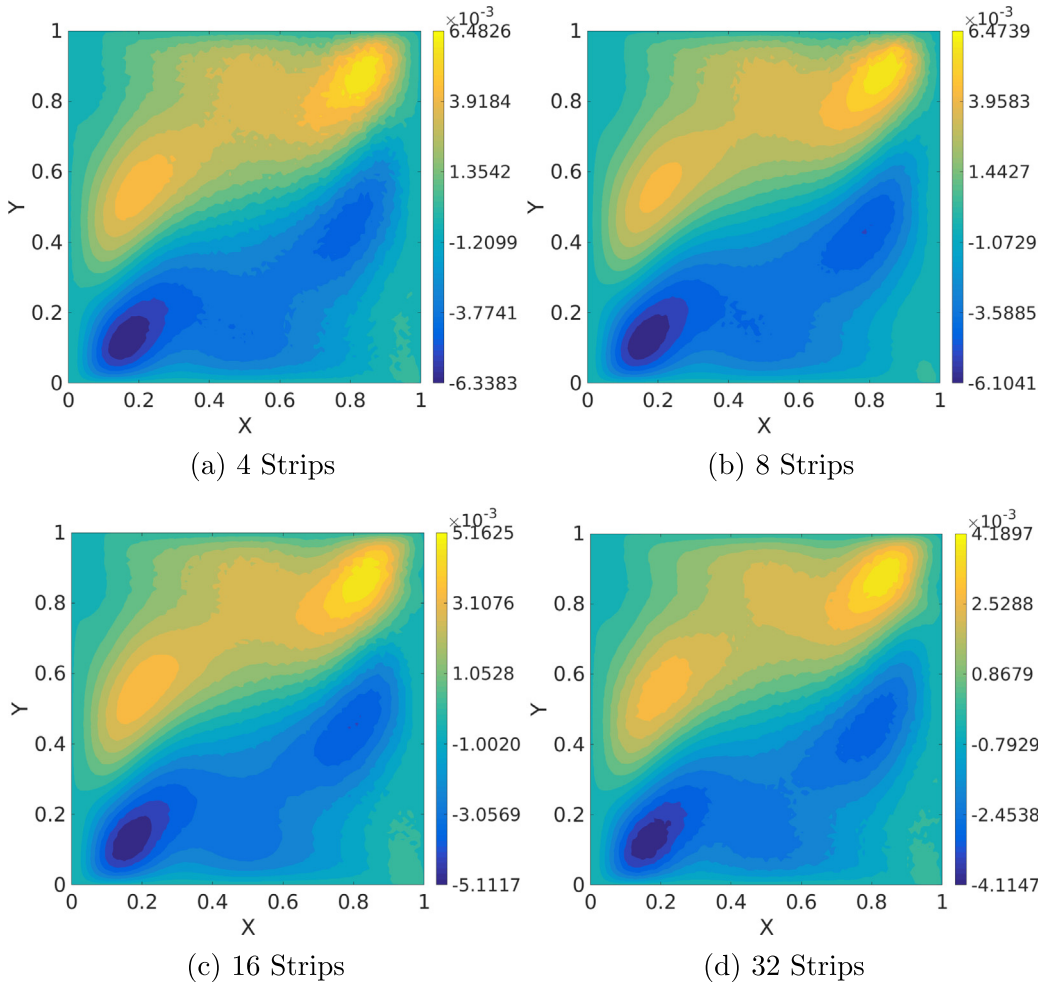


Fig. 21. $Ra = 10^6 Z$ midplane X-velocity: difference between stochastic mean and deterministic value.

istic value of the temperature is of the order of 3–4% of the deterministic value. For velocities the difference is around 4–5% of the deterministic value. This is smaller than that of the Nusselt number which is 8–12%. Thus, the effect of boundary uncertainty is more pronounced in the Nusselt number compared to the temperature and velocities. The stochastic standard deviation is quite similar for all the four cases and thus, only the case with 32 strips is plotted here (Fig. 23). It can be seen it is orders of magnitude smaller than the shift in the mean.

6. Conclusions

This paper presents the input uncertainty propagation results for a three dimensional natural convection problem in a differentially heated cubical enclosure with two different cases. Case A assumes that the boundary conditions are uniform. Thus, uncertainty is introduced in the two non-dimensional parameters (Rayleigh and Prandtl numbers). For this case, the polynomial chaos expansion (PCE) method is used as a surrogate model with stochastic collocation to estimate the PCE coefficients. Case B deals with non-uniform stochastic boundary condition with deterministic material properties of the fluid. Since the temperature difference between the opposite walls drives the natural convection flow, the cold wall is held at a constant temperature and uncertainty is introduced in the hot wall temperature. A deep neural network based surrogate model is used here for estimating the output statistics with Monte Carlo method. It is observed that the mean value of an output parameter averaged over the stochastic variation of the input can be different from the deterministic output value. The difference normalized by the deterministic value is defined as the relative shift of mean. Table 3 summarizes the shift of mean relative to deterministic values of each of the output. Mean Rayleigh numbers of 10^5 and 10^6 are considered in case A whereas, case B deals only with a mean of 10^6. The relative values of input standard deviation with respect to mean is 2% and 3.33% for case A and case B respectively. It can be seen that for case A, the stochastic effect is negligible in the Nusselt number and temperature whereas, it is of similar order as input in the velocities. On the other hand, for case B, the stochastic effect is much higher in Nusselt number. For both the cases, the standard deviation is much lower than the shift of mean. In general, it is concluded that the effect of uncertainties in the boundary conditions are much higher than the uncertainties in the Rayleigh and Prandtl numbers with uniform boundary conditions. This paper also demonstrates the use of deep neural network for uncertainty quantification in natural convection problem for a high dimensional stochastic problem. The novel approach of dividing the boundary surface into domains and treating each domain value as a stochastic input is demonstrated here for the first time (to the best of our knowledge). This approach tries to mimic the experiments, when multiple feedback control systems are used to impose the boundary conditions. Note that although in this work, it is assumed that the domain values are independent and identically distributed variables, this assumption can be relaxed during sampling if additional data of

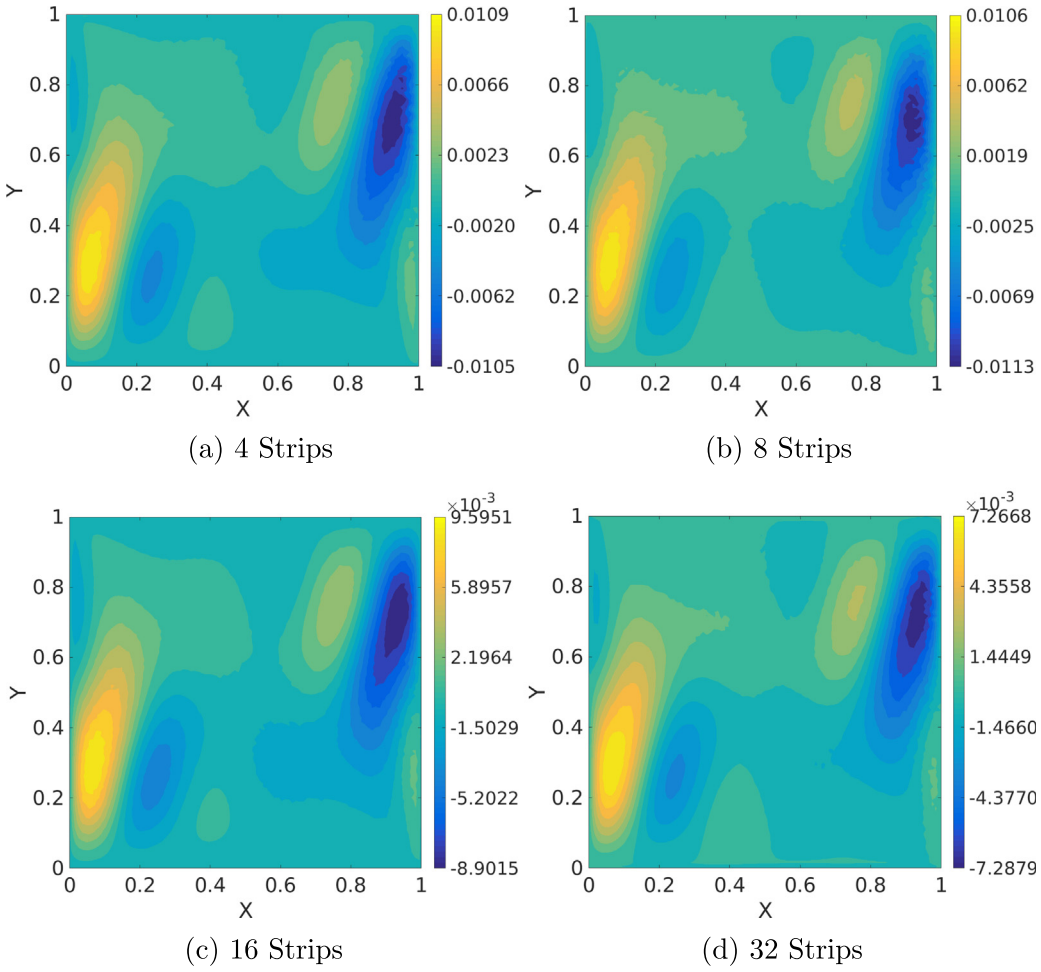


Fig. 22. $Ra = 10^6 Z$ midplane y-velocity: difference between stochastic mean and deterministic value.

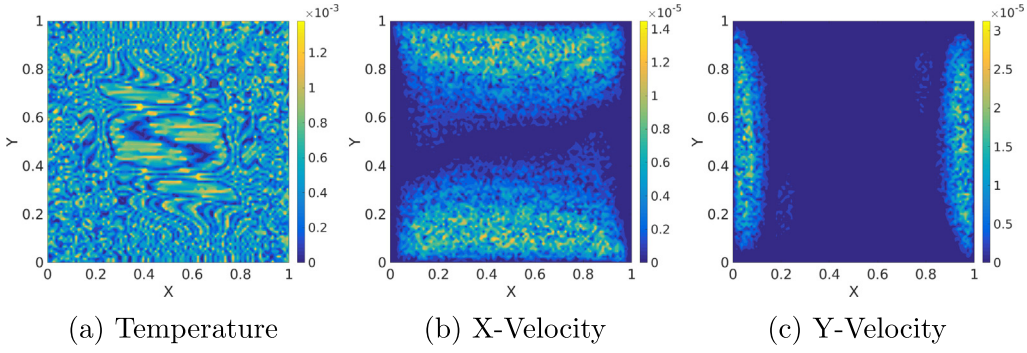


Fig. 23. Stochastic standard deviation at Z mid-plane $Ra = 10^6$ (32 strips).

Table 3
Shift of mean relative to deterministic values.

	Input	Nusselt No.	Temperature	Velocities
Case A $Ra = 10^5$	$\sigma = 2\%\mu$	0.3%	0.1%	3%
Case A $Ra = 10^6$	$\sigma = 2\%\mu$	0.3%	0.3%	1.5%
Case B $Ra = 10^6$	$\sigma = 3.33\%\mu$	12%	4%	5%

the feedback control system is available. The approach can also be used for other class of problems like optimization and inverse problems.

Declaration of Competing Interest

None.

Acknowledgments

This work was funded by the Digital Manufacturing and Design Innovation Institute with support in part by the U.S. Department of the Army. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Army.

References

[1] G. De Vahl Davis, Natural convection of air in a square cavity: a bench mark numerical solution, Int. J. Numer. Methods Fluids 3 (3) (1983) 249–264.
[2] T. Fusegi, J. Hyun, K. Kuwahara, B. Farouk, A numerical study of three-dimensional natural convection in a differentially heated cubical enclosure, Int. J. Heat Mass Transf. 34 (6) (1991) 1543–1557.
[3] P. Le Quéré, Accurate solutions to the square thermally driven cavity at high Rayleigh number, Comput. Fluids 20 (1) (1991) 29–41.
[4] Y. Hu, Y. Li, C. Wu, S. Li, M. Li, Flow pattern and heat transfer in Rayleigh-Bénard convection of cold water near its density maximum in a rectangular cavity, Int. J. Heat Mass Transf. 107 (2017) 1065–1075.
[5] Y. Li, H. Zhang, L. Zhang, C. Wu, Three-dimensional numerical simulation of double-diffusive Rayleigh-Bénard convection in a cylindrical enclosure of aspect ratio 2, Int. J. Heat Mass Transf. 98 (2016) 472–483.
[6] C. Shu, H. Ding, K. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 192 (7–8) (2003) 941–954.
[7] S. Yigit, R. Poole, N. Chakraborty, Effects of aspect ratio on laminar Rayleigh-Bénard convection of power-law fluids in rectangular enclosures: a numerical investigation, Int. J. Heat Mass Transf. 91 (2015) 1292–1307.
[8] T. Icoz, Y. Jaluria, Design of cooling systems for electronic equipment using both experimental and numerical inputs, J. Electron. Packag. 126 (4) (2004) 465–471.
[9] A. Sharma, K. Velusamy, C. Balaji, S. Venkateshan, Conjugate turbulent natural convection with surface radiation in air filled rectangular enclosures, Int. J. Heat Mass Transf. 50 (3–4) (2007) 625–639.
[10] M. Herrlin, C. Belady, Gravity-assisted air mixing in data centers and how it affects the rack cooling effectiveness, in: Thermal and Thermomechanical Phenomena in Electronics Systems, 2006. ITHERM'06. The Tenth Intersociety Conference on, IEEE, 2006, p. 5.
[11] S. Patankar, Numerical Heat Transfer and Fluid Flow, CRC Press, 1980.
[12] D. Xiu, G. Karniadakis, Modeling uncertainty in steady state diffusion problems via generalized polynomial chaos, Comput. Methods Appl. Mech. Eng. 191 (43) (2002) 4927–4948.
[13] D. Xiu, G. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos, J. Comput. Phys. 187 (1) (2003) 137–167.
[14] O. Maitre, O. Knio, H. Najm, R. Ghanem, A stochastic projection method for fluid flow: I. Basic formulation, J. Comput. Phys. 173 (2) (2001) 481–511.
[15] O. Maitre, M. Reagan, B. Debusschere, H. Najm, R. Ghanem, O. Knio, Natural convection in a closed cavity under stochastic non-Boussinesq conditions, SIAM J. Scient. Comput. 26 (2) (2004) 375–394.
[16] D. Venturi, M. Choi, G. Karniadakis, Supercritical quasi-conduction states in stochastic Rayleigh-Bénard convection, Int. J. Heat Mass Transf. 55 (13–14) (2012) 3732–3743.
[17] B. Ganapathysubramanian, N. Zabaras, Sparse grid collocation schemes for stochastic natural convection problems, J. Comput. Phys. 225 (1) (2007) 652–685.
[18] M. Carnevale, F. Montomoli, A. DAmmaro, S. Salvadori, F. Martelli, Uncertainty quantification: a stochastic method for heat transfer prediction using LES, J. Turbomach. 135 (5) (2013) 051021.
[19] P. Marepalli, J. Murthy, B. Qiu, X. Ruan, Quantifying uncertainty in multiscale heat conduction calculations, J. Heat Transf. 136 (11) (2014), 111301.
[20] N. Fajraoui, M. Fahs, A. Younes, B. Sudret, Analyzing natural convection in porous enclosure with polynomial chaos expansions: effect of thermal dispersion, anisotropic permeability and heterogeneity, Int. J. Heat Mass Transf. 115 (2017) 205–224.
[21] K. Fezi, M. Krane, Uncertainty quantification in modeling metal alloy solidification, J. Heat Transf. 139 (8) (2017), 082301.
[22] S. Hosder, R. Walters, R. Perez, A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, 891, 2006.
[23] D. Kumar, M. Raisee, C. Lacor, An efficient non-intrusive reduced basis model for high dimensional stochastic problems in CFD, Comput. Fluids 138 (2016) 67–82.
[24] S. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Soviet Math. Dokl. 4 (1963) 240–243.
[25] F. Heiss, V. Winschel, Likelihood approximation by numerical integration on sparse grids, J. Economet. 144 (1) (2008) 62–80.
[26] R. Caflisch, Monte carlo and quasi-Monte carlo methods, Acta Numer. 7 (1998) 1–49.
[27] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (5) (1989) 359–366.
[28] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Networks 61 (2015) 85–117.
[29] A. Gholami, H. Bonakdari, A. Zaji, A. Akhtari, Simulation of open channel bend characteristics using computational fluid dynamics and artificial neural networks, Eng. Appl. Comput. Fluid Mech. 9 (1) (2015) 355–369.
[30] S. Sablani, A. Kacimov, J. Perret, A. Mujumdar, A. Campo, Non-iterative estimation of heat transfer coefficients using artificial neural network models, Int. J. Heat Mass Transf. 48 (3–4) (2005) 665–679.
[31] A. Santra, N. Chakraborty, S. Sen, Prediction of heat transfer due to presence of copper-water nanofluid using resilient-propagation neural network, Int. J. Therm. Sci. 48 (7) (2009) 1311–1318.
[32] B. Czél, K. Woodbury, G. Gróf, Simultaneous estimation of temperature-dependent volumetric heat capacity and thermal conductivity functions via neural networks, Int. J. Heat Mass Transf. 68 (2014) 1–13.
[33] Z. Zhang, K. Duraisamy, Machine learning methods for data-driven turbulence modeling, in: 22nd AIAA Computational Fluid Dynamics Conference, 2015, p. 2460.
[34] R. Tripathy, I. Bilionis, Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification, arXiv preprint arXiv:1802.00850.
[35] Y. Khoo, J. Lu, L. Ying, Solving parametric PDE problems with artificial neural networks, arXiv preprint arXiv:1707.03351.
[36] D. Zhang, L. Lu, L. Guo, G. Karniadakis, Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems, arXiv preprint arXiv:1809.08327.
[37] P. Naphon, S. Wiriyasart, T. Arisariyawong, Artificial neural network analysis the pulsating Nusselt number and friction factor of TiO2/water nanofluids in the spirally coiled tube with magnetic field, Int. J. Heat Mass Transf. 118 (2018) 1152–1159.
[38] P. Naphon, S. Wiriyasart, T. Arisariyawong, L. Nakharintr, ANN, numerical and experimental analysis on the jet impingement nanofluids flow and heat transfer characteristics in the micro-channel heat sink, Int. J. Heat Mass Transf. 131 (2019) 329–340.
[39] M. Nabian, H. Meidani, A deep neural network surrogate for high-dimensional random partial differential equations, arXiv preprint arXiv:1806.02957.
[40] S. Karumuri, R. Tripathy, I. Bilionis, J. Panchal, Simulator-free Solution of High-Dimensional Stochastic Elliptic Partial Differential Equations using Deep Neural Networks, arXiv preprint arXiv:1902.05200.
[41] S. Chan, A. Elsheikh, A machine learning approach for efficient uncertainty quantification using multiscale methods, J. Comput. Phys. 354 (2018) 493–511.
[42] S. Shahane, N. Aluru, P. Ferreira, S.G. Kapoor, S.P. Vanka, Finite volume simulation framework for die casting with uncertainty quantification, Appl. Math. Model. 2019.
[43] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (12) (1965) 2182–2189.
[44] D. Xiu, G. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, SIAM J. Scient. Comput. 24 (2) (2002) 619–644.
[45] N. Wiener, The homogeneous chaos, Am. J. Math., vol. 60 (4).
[46] R. Smith, Uncertainty quantification: theory, implementation, and applications, 2013.
[47] S. Marelli, B. Sudret, UQLab: A framework for uncertainty quantification in Matlab, in: Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management, 2014, pp. 2554–2563.
[48] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, <http://www.deeplearningbook.org>, 2016.
[49] Y. Chauvin, D. Rumelhart, Backpropagation: Theory, Architectures, and Applications, Psychology Press, 2013.
[50] D. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980.
[51] I. Sobol, Sensitivity estimates for nonlinear mathematical models, Math. Model. Comput. Exp. 1 (4) (1993) 407–414.
[52] pyDOE: The experimental design package for python, URL <https://pythonhosted.org/pyDOE/>, 2018.
[53] Statistical functions (scipy.stats), URL <https://docs.scipy.org/doc/scipy/reference/stats.html>, 2018.
[54] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, URL <https://www.tensorflow.org/>, software available from tensorflow.org, 2015.
[55] F. Chollet, et al., Keras, <https://keras.io>, 2015.