ESTIMATION OF HEAT TRANSFER PARAMETERS BY USING TRAINED POD-RBF AND GREY WOLF OPTIMIZER

Minh Ngoc Nguyen[1,2,*]**, Nha Thanh Nguyen**[1,2]**, Thien Tich Truong**[1,2]
[1]*Ho Chi Minh City University of Technology, Vietnam*
[2]*Vietnam National University Ho Chi Minh City, Vietnam*
*E-mail: nguyenngocminh@hcmut.edu.vn

Abstract. The article presents a numerical model for estimation of heat transfer parameters, e.g. thermal conductivity and convective coefficient, in two-dimensional solid bodies under steady-state conduction. This inverse problem is stated as an optimization problem, in which input is reference temperature data and the output is the design variables, i.e. the thermal properties to be identified. The search for optimum design variables is conducted by using a recent heuristic method, namely Grey Wolf Optimizer. During the heuristic search, direct heat conduction problem has to be solved several times. The set of heat transfer parameters that lead to smallest error rate between computed temperature field and reference one is the optimum output of the inverse problem. In order to accelerate the process, the model order reduction technique Proper-Orthogonal-Decomposition (POD) is used. The idea is to express the direct solution (temperature field) as a linear combination of orthogonal basis vectors. Practically, a majority of the basis vectors can be truncated, without losing much accuracy. The amplitude of this reduced-order approximation is then further interpolated by Radial Basis Functions (RBF). The whole scheme, named as trained POD-RBF, is then used as a surrogate model to retrieve the heat transfer parameters.

Keywords: inverse analysis, Grey Wolf Optimizer, heat transfer parameters identification, Proper Orthogonal Decomposition (POD), Radial Basis Function (RBF).

1. INTRODUCTION

In direct heat transfer analysis, distribution of temperature within a conducting domain is determined given known boundary conditions and thermal properties. In contrast, based on the knowledge of temperature history within a conducting body, inverse heat transfer analysis is used to determine the thermal properties and/or boundary conditions. The estimated quantities of inverse heat transfer analysis are very sensitive to the inaccuracy of input data. Mathematically, the problem is ill-posed [1]. Unfortunately,

noise in measurement of temperature is not avoidable. Therefore, development of computational schemes which can overcome the issue of ill-posedness has attracted much attention from researchers.

Inverse analysis has been widely used in heat transfer to identify heat flux [2–4], boundary conditions [5–7] and unknown thermal properties such as conductivity and convective coefficient [8–11]. Basically, the problem is described as minimization of the error rate between computed temperature and measured data. The design variables are the unknown quantities to be determined. For solution of optimization problem, either gradient-based or non-gradient-based methods can be used. The gradient-based approaches [3,4,8] usually involve sensitivity analysis, i.e. the computation of derivative of objective function with respect to the sought variables. However, derivation of objective function as an explicit function of design variables is usually not a trivial task. Another drawback is that the gradient-based approach may fall into local optimum. On the other hand, the non-gradient-based methods do not require sensitivity analysis. Instead, various heuristic algorithms are used such as Genetic Algorithm [11], Particle Swarm Optimization [2], Differential Evolution [12], Firefly Algorithm [7], Cuckoo Search [13] and so on. Although each algorithm has a different strategy, they commonly employ a group of M agents which search N rounds in the admissible solution space to find the optimum one, i.e. the unknown quantities to be estimated. Indeed, it is common knowledge that there exists no algorithm which is superior to the others in all types of problems. Nevertheless, the attractiveness of GWO algorithm comes from the fact that it has small number of user-defined parameters to control the balance of *exploitation* (local search) and *exploration* (global search). In this work, the recently proposed Grey Wolf Optimizer (GWO) [14] is used to solve the optimization problem to identify the thermal parameters, e.g. heat conductivity and convective coefficient. The algorithm has been widely applied in many fields such as machine learning [15,16], electric engineering [17], earthquake engineering [18], image processing [19], path planning [20]. However, to the best knowledge of the authors, GWO has not been investigated in inverse heat transfer analysis.

During the search for optimum solution, the direct heat transfer problems have to be solved many times to evaluate temperature field. The difference between the computed temperature and reference one, i.e. the objective function, is then determined. The process is time-consuming and needs to be accelerated. The model order reduction technique Proper Orthogonal Decomposition (POD) has been successfully employed in direct heat transfer problems [21–24]. The core idea is to find a set of orthogonal vectors (POD bases) using singular value decomposition, which is then utilized to approximate the temperature field. Temperature is expressed as a linear combination of POD basis and associated amplitudes. Usually, this linear combination can be truncated, thus the problem size is reduced, while high accuracy is still attained. Ostrowski et al. [9,25] pointed out that POD also acts as a filter to lessen the influence of noise in measured temperature data, improving the stability of inverse heat transfer analysis. Consequently, the benefit of the employment of POD in inverse heat transfer problems is two-fold: acceleration of computational process and regularization method to treat the ill-posedness. The amplitude vectors in POD approximation is then further interpolated using Radial Basis

Functions (RBF), which are defined as functions of thermal parameters, resulting in the *trained POD-RBF* surrogate model [9, 26–28].

In this paper, the trained POD-RBF is coupled with GWO to develop a numerical model to identify thermal conductivity and convective coefficient, in two-dimensional solid bodies under steady-state conduction.

The paper is organized as follows. Immediately after the Introduction, a brief review of GWO is presented in Section 2. Section 3 is reserved for trained POD-RBF in identification of thermal properties. In Section 4, a numerical example is presented and discussed in details, demonstrating the numerical scheme. Finally, conclusions and remarks are given in the last Section.

Nomenclature (units are given in square bracket)			
Symbol	Definition	Symbol	Definition
α	The *alpha* wolf (i.e. the search agent that has the best fitness in the whole search)	δ	The *delta* wolf (i.e. the search agent that has the third best fitness in the whole search)
β	The *beta* wolf (i.e. the search agent that has the second best fitness in the whole search)		
T [K]	Temperature	h [W/(m^{2}K)]	Convective heat transfer co-efficient
q [W/m^2]	Heat flux	k [W/(mK)]	Thermal conductivity
T_a [K]	Ambient temperature		
$\mathbf{T}_{snap}$	Snapshot matrix	$\mathbf{p}$	Vector of thermal properties (i.e. h and k in the current work)
$\mathbf{\Phi}$	Orthogonal basis vectors		

2. GREY WOLF OPTIMIZER (GWO)

GWO is a bio-inspired optimization technique recently proposed by Mirjalili et al. [14]. In an attempt to mimic the social hierarchy of grey wolf, the fitness of wolves after each iteration is sorted in ascending order (in the context of minimization problem, the wolf with lowest value of objective function is the fittest). The three fittest solutions are named the *alpha* (α), the *beta* (β), and the *delta* (δ), respectively. The rest of the population is called *omegas*. With the hypothesis that the leadership hierarchy of grey wolf also applies in hunting process, the algorithm updates the position of an ordinary grey wolf

(i.e the *omegas*) at the current iteration $t + 1$, $\vec{X}(t+1)$, by the last known positions of the best candidates, i.e. the *alpha*, *beta* and *delta* wolf

$$\vec{X}(t+1) = \frac{1}{3}\left(\vec{X}_1 + \vec{X}_2 + \vec{X}_3\right), \tag{1}$$

where $\vec{X}_1, \vec{X}_2, \vec{X}_3$ are some points surrounding the positions of three dominant wolves (denoted by $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$)

$$\vec{X}_1 = \vec{X}_\alpha - a_\alpha \cdot \vec{D}_\alpha, \quad \vec{D}_\alpha = \left(c_\alpha \cdot \vec{X}_\alpha - \vec{X}(t)\right), \tag{2}$$

$$\vec{X}_2 = \vec{X}_\beta - a_\beta \cdot \vec{D}_\beta, \quad \vec{D}_\beta = \left(c_\beta \cdot \vec{X}_\beta - \vec{X}(t)\right), \tag{3}$$

$$\vec{X}_3 = \vec{X}_\delta - a_\delta \cdot \vec{D}_\delta, \quad \vec{D}_\delta = \left(c_\delta \cdot \vec{X}_\delta - \vec{X}(t)\right), \tag{4}$$

The numbers a_i and c_i ($i = \alpha, \beta, \delta$) are calculated by

$$a_i = 2s \cdot r_1 - s, \tag{5}$$

$$c_i = 2r_2, \tag{6}$$

where r_1 and r_2 are random real values ranging from 0 to 1. Parameter s gradually decreases from some pre-defined value $s_{\max}$ (in [14], $s_{\max}$ is set to 2) to zero with respect to the number of iterations

$$s = s_{\max}\left(1 - \frac{t}{t_{\max}}\right), \tag{7}$$

with $t_{\max}$ being the pre-set maximum number of iterations. The value of controlling parameter s has influence on a_i in Eq. (5), which is key for a wolf to decide whether it approaches or run away from the three leading wolves (the alpha, beta and delta). Particularly, if $|a_i| < 1$, the wolf will join with the three dominant ones to encircle and attack the prey. This is *exploitation*, i.e. the local search in optimization. On the other hand, if $|a_i| > 1$, the wolf runs away to explore the space far from the leaders, with a hope to discover a more attractive prey. This option allows *exploration*, i.e. the global search, in order to avoid being trapped in local optimum.

Gao and Zhao [29] argue that the equal weights in Eq. (1) do not reflect the rank of the three dominant wolves. The individual roles of the *alpha*, *beta* and *delta* are the same, despite the fact that *alpha* is closest to the prey (in the context of optimization problem). Instead, more weights should be assigned to the *alpha* in order to enhance local search. Furthermore, the weights should also follow a descending order: $\omega_1 \geq \omega_2 \geq \omega_3 \geq 0$. Based on the above reasoning, they propose the following calculation of the weights

$$\omega_1 = \cos\theta, \quad \omega_2 = \frac{1}{2}\sin\theta \cdot \cos\phi, \quad \omega_3 = 1 - \omega_1 - \omega_2, \tag{8}$$

where

$$\theta = \frac{2}{\pi} \cdot \arccos\left(\frac{1}{3}\right) \cdot \arctan t \text{ and } \phi = \frac{1}{2} \cdot \arctan t. \tag{9}$$

The second argument of Gao and Zhao [29] is that in the beginning of the search, the wolves should be encouraged to go for a global search, while in long term, local

search should be more emphasized. Therefore the controlling parameter s is suggested to decline exponentially, instead of linearly as in Eq. (7)

$$s = s_{\max} \cdot \exp\left(-\frac{10t}{t_{\max}}\right).$$ (10)

3. TRAINED POD-RBF FOR IDENTIFICATION OF THERMAL PROPERTIES

3.1. Governing equations of direct heat transfer problems in two-dimensional domains

Let us consider a two-dimensional solid body Ω being bounded by Γ. When there is no heat sink/source, the governing equation of steady-state heat transfer in the body Ω is written by

$$\nabla \cdot (k\nabla T) = 0,$$ (11)

where T is the temperature and k is the thermal conductivity. Without consideration of heat radiation, the boundary conditions are given as follows

$$T = \bar{T}, \text{on } \Gamma_1: \text{Dirichlet boundary},$$ (12)

$$(k\nabla T) \cdot \mathbf{n} = \bar{q}, \text{on } \Gamma_2: \text{Neumann boundary},$$ (13)

$$(k\nabla T) \cdot \mathbf{n} = h(T_a - T), \text{on } \Gamma_3: \text{convection boundary}.$$ (14)

In Eqs. (12)–(14), $\bar{T}$ is the prescribed temperature; $\bar{q}$ is the prescribed heat flux; $\mathbf{n}$ is the outward normal unit vector of the boundary; T_a is the ambient temperature and h is the convective heat transfer coefficient.

After some mathematical manipulation, the partial differential equation (11) is transformed into weak formulation as follows

$$\int_\Omega (\delta\nabla T) k\nabla T d\Omega - \int_\Gamma \bar{q}\delta T d\Gamma - \int_\Gamma h(T_a - T)\delta T d\Gamma = 0.$$ (15)

3.2. Training data and reference data

Given the same domain geometry and boundary conditions, the training data are temperature values obtained from solution of direct problem, corresponding to *known* thermal properties, i.e. thermal conductivity k and convective coefficient h. One set of (k, h) is connected to one set of training temperature data. In fact, the training data can be obtained by measurement, given that the number of experiments and the number of sampling points are large enough. Another option is that a finite element model can be developed for generation of training data.

Reference data are temperature values collected at some certain points in problem domain (usually on the boundaries). Thermal properties that lead to reference data are not known *a priori* and have to be identified by inverse analysis. In this paper, the reference data are also taken from finite element solution of the direct steady-state heat transfer. At each point, a noise of 5% is added to finite element solution to mimic that of measurement.

3.3. Model order reduction by Proper Orthogonal Decomposition

The training data can be arranged as an *m-by-n* matrix $\mathbf{T}_{snap}$, in which n is the number of data sets (one data set corresponds to one set of parameters (k, h)), and m is the number of points where the data are collected. In this work, nodal values of temperature at all nodes obtained by direct solution of finite element analysis are taken as the training data. Following the terminology used in literatures [21–24, 30], each column of training data is called a *snapshot*, and the matrix of training data itself is called the *snapshot matrix*

$$\mathbf{T}_{snap} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{T}_2 & \dots & \mathbf{T}_i & \dots & \mathbf{T}_n \end{bmatrix}. \tag{16}$$

A singular value decomposition applied on $\mathbf{T}_{snap}$ reads

$$\mathbf{T}_{snap} = \mathbf{\Phi}\mathbf{D}\mathbf{V}^T, \tag{17}$$

where $\mathbf{\Phi}$ (size *m-by-m*) and $\mathbf{V}$ (size *n-by-n*) are orthogonal matrices, and $\mathbf{D}$ is a rectangular matrix of size *m-by-n*. In matrix $\mathbf{D}$, only the values along the diagonal are non-negative, which are called *singular values*, while the rest are all zeroes. In practice, the singular values are sorted in descending order, i.e. $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r \geq 0$, $r = \min(m, n)$. Denote $\mathbf{A} = \mathbf{D}\mathbf{V}^T$, Eq. (17) can be rewritten as

$$\mathbf{T}_{snap} = \mathbf{\Phi}\mathbf{A}. \tag{18}$$

By Eq. (18), the snapshot matrix is expressed as a linear combination, in which $\mathbf{\Phi}$ is the set of orthogonal basis vectors and $\mathbf{A}$ stores the associated amplitudes. Taking the advantage that the singular values in $\mathbf{D}$ drop quickly to zero, the snapshot can be approximated with up to l terms, with $l \leq r$, without losing much accuracy

$$\mathbf{T}_{snap} \approx \bar{\mathbf{\Phi}}\bar{\mathbf{A}}, \tag{19}$$

in which the set of truncated orthogonal basis vectors $\bar{\mathbf{\Phi}}$ is the first l columns of $\mathbf{\Phi}$. The set of truncated amplitudes is calculated by

$$\bar{\mathbf{A}} = \bar{\mathbf{\Phi}}^T\mathbf{T}_{snap}. \tag{20}$$

Similarly to [23], the "cumulative energy coefficient" is defined as

$$e\left(l\right) = \frac{\sum_{i=1}^{l} \lambda_i}{\sum_{j=1}^{r} \lambda_j}. \tag{21}$$

The "truncated energy" is then calculated by

$$\varepsilon = 1 - e\left(l\right). \tag{22}$$

Simply by setting the expected value of ε, e.g. $\varepsilon = 10^{-8}$, the l number of POD basis vectors can be selected.

3.4. Approximation of the amplitudes by Radial Basis Function (RBF)

Let the amplitudes in Eq. (20) be function of thermal properties, the following linear combination can be written for each column of $\bar{\mathbf{A}}$

$$\bar{\mathbf{a}} = \bar{\mathbf{a}}(\mathbf{p}) = \mathbf{B} \cdot \mathbf{f}(\mathbf{p}), \tag{23}$$

in which $\mathbf{B}$ stores the unknown coefficients; $\mathbf{p}$ is the vector of thermal properties; and $\mathbf{f}$ is the vector of n Radial Basis Functions (corresponding to n sets of parameters mentioned in Section 3.2)

$$\mathbf{f}(\mathbf{p}) = \begin{bmatrix} f_1(\mathbf{p}) & f_2(\mathbf{p}) & \cdots & f_i(\mathbf{p}) & \cdots & f_n(\mathbf{p}) \end{bmatrix}^T. \tag{24}$$

Various types of RBF have been introduced in literatures. Curious readers are referred to [31] for details. Here, the recently proposed quartic polynomial radial basis is employed [32]

$$f_i(\mathbf{p}) = 1 - 6r_i^2 + 8r_i^3 - 3r_i^4, \text{ where } r_i = \left\| \mathbf{p} - \mathbf{p}^i \right\|. \tag{25}$$

Requiring that Eq. (23) holds for all the snapshots in the training data, the following matrix equation is obtained

$$\bar{\mathbf{A}} = \bar{\mathbf{A}}(\mathbf{p}) = \mathbf{B} \cdot \mathbf{F}(\mathbf{p}), \tag{26}$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & f_1\left(\mathbf{p}^2, \mathbf{p}^1\right) & \cdots & f_1\left(\mathbf{p}^n, \mathbf{p}^1\right) \\ f_2\left(\mathbf{p}^1, \mathbf{p}^2\right) & 1 & \cdots & f_2\left(\mathbf{p}^n, \mathbf{p}^2\right) \\ \vdots & \vdots & \ddots & \vdots \\ f_n\left(\mathbf{p}^1, \mathbf{p}^n\right) & f_n\left(\mathbf{p}^2, \mathbf{p}^n\right) & \cdots & 1 \end{bmatrix}. \tag{27}$$

The n sets of thermal properties in Eq. (24), i.e. $\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^n$, are the sets used to get training data and thus are all known. Therefore, matrices $\mathbf{F}$ and $\mathbf{B}$ can be easily computed.

When POD basis $\bar{\mathbf{\Phi}}$ and the matrix of coefficients $\mathbf{B}$ are known, the POD-RBF system has been trained. For an arbitrary set $\mathbf{p}$, e.g. the one generated by the optimization algorithm, the temperature values can be quickly retrieved by

$$\mathbf{T}_{retrieved} = \bar{\mathbf{\Phi}} \cdot \mathbf{B} \cdot \mathbf{f}(\mathbf{p}). \tag{28}$$

4. NUMERICAL EXAMPLES

Let us consider a steady-state heat transfer problem in a complicated domain as presented in Fig. 1. The width of the three fins are the same. Temperature on the right surface is prescribed by $T = 300$ K. Heat convection takes place on the left surface with ambient temperature $T_a = 200$ K and convective coefficient is h W/(m^{2}K). The other boundaries are all insulated. Thermal conductivity within the domain is k W/(mK). Inward heat flux is applied on the curved surface of the middle fin is $q = 20000$ W/m^2. Parameters h and k will be identified by the proposed trained POD-RBF system.

The finite element model, which is used to generate the training data, is verified by a convergence study. Three levels of quadrilateral mesh are considered: 219 elements (272

nodes), 546 elements (622 nodes), 1984 elements (2125 nodes). The "equivalent thermal energy" is defined as

$$U = \int_{\Omega} (\nabla T)^T k (\nabla T) \, \mathrm{d}\Omega \tag{29}$$
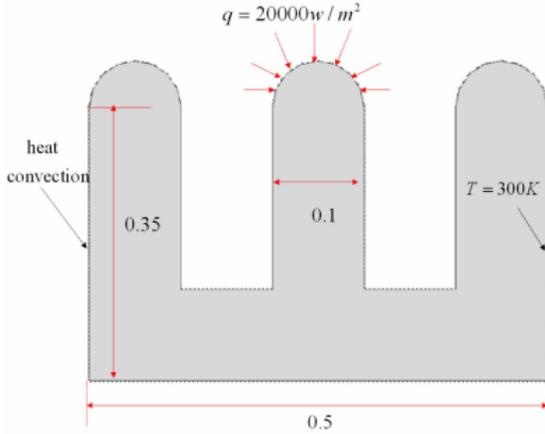
Fig. 1. Geometry and boundary conditions (dimensions are in meter)
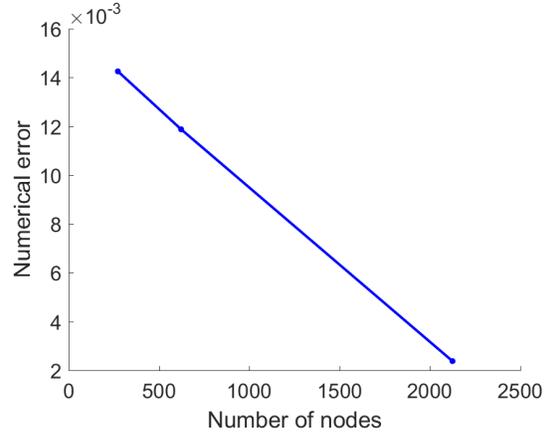
Fig. 2. Convergence of equivalent thermal energy (see Eq. (29)) with respect to the number of nodes

The value of h is h = 100 W/(m²K) and that of k is k = 100 W/(mK). The convergence of the equivalent thermal energy with respect to number of nodes is displayed in Fig. 2. Due to the lack of analytical solution, the result obtained by a fine mesh of 4464 elements (4675 nodes) is used as reference to evaluate the numerical error. It is observed that with the mesh of 546 elements (see Fig. 3), numerical error is only 1.2%. In linear heat transfer analysis, which is the case being considered, the same convergence would be recorded for other values of h and k. Therefore, it is acceptable to use the mesh of 546 quadrilateral elements to generate the training data.

The training data, i.e. the snapshot matrix defined in Eq. (16), is generated by finite element analysis (FEA) of direct problems, using

Fig. 3. Finite element mesh and location of 10 reference points

the following sets: k = 1, 6, 11, ..., 196, 201 W/(mK) and h = 1, 6, 11, ..., 196, 201 W/(m²K). In fact, the lower bound and upper bound of design parameters shall be guessed. Uniform discretization of design space is a basic and common approach. In
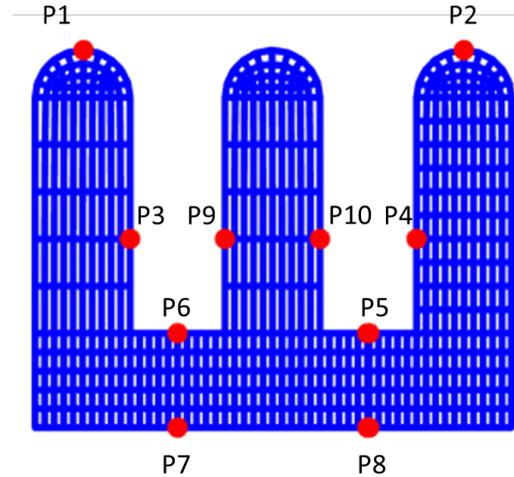
order to reduce the number of training data, the Taguchi's method for design of experiments can be employed, as presented by [33]. However, this method is not within the scope of the current work.

Temperature at 10 points (marked by dots in Fig. 3) are taken as reference data, with k = 87.25 W/(mK) and h = 103.5 W/(m^{2}K). Minimization of error rate between reference temperature and the values retrieved by the trained POD-RBF is the objective of the optimization block using Grey Wolf Optimizer. In order to mimic measurement error, 5% noise is added into the finite element solution, i.e. the "measured" temperature at each point is assumed to be within the range $0.95T_{FEM} \leq T \leq 1.05T_{FEM}$. Reference temperature at each point is the averaged value of 5 "measurements". Details are presented in Tab. 1. A comparison study is conducted between two variants of Grey Wolf Optimizer: the original one as described in [14], denoted by *GWO*, and the improved one, denoted by *VW-GWO*. In VW-GWO, variable weights (Eq. (8)) and the exponential-decay control parameter (Eq. (10)) are used.

Figure 3. Finite element mesh and location of 10 reference points

Table 1. The 10 reference points

Points	Coordinates	Ref. Temperature [K] (FEA solution, without noise)	Ref. Temperature [K] (FEA solution, 5% noise)
P1	[0.05, 0.4]	253.2039	256.8404
P2	[0.45, 0.4]	300.0839	296.5855
P3	[0.1, 0.2]	261.5149	256.1581
P4	[0.4, 0.2]	300.8999	296.8543
P5	[0.35, 0.1]	313.1507	315.2422
P6	[0.15, 0.1]	299.1058	291.6881
P7	[0.15, 0]	298.4134	298.1384
P8	[0.35, 0]	311.9706	312.8024
P9	[0.2, 0.2]	361.0809	357.3288
P10	[0.3, 0.2]	361.2115	356.1655

Two cases are consider: (a) Reference data are obtained without noise and (b) Reference data are obtained with 5% noise. For each case, the inverse analysis is run 10 times by both GWO and VW-GWO. In all cases, the number of grey wolves is 10.

Results are presented in Tab. 2. It is observed that for both cases (i.e. zero noise and 5% noise in reference data), VW-GWO exhibits better performance than GWO. Although the mean values of estimated k and h are almost the same, the standard deviation in VW-GWO is much lower. For comparison, the results obtained by Genetic Algorithm (GA) are also presented. Agreement between the three algorithms can be observed, although the performance of GA is slightly behind. The possible reason is that the information of the best agents are taken into account by the two GWO variants, but not by GA.

For case (a), i.e. zero noise, the values of k and h by the surrogate model are almost equal to the true ones. For case (b), i.e. 5% noise in reference data, error rates of the mean

values of estimated k and h, compared with the correct ones (i.e. $k = 87.25$ and $h = 103.5$), are 4.87% and 5.40%, respectively. These error rates are very close to the noise existed in reference data. The above results have demonstrated the accuracy of inverse analysis using trained POD-RBF and GWO.

Table 2. Parameters estimated by the proposed model in both cases (a) and (b). The true values of k and h are: $k = 87.25$ W/(mK) and $h = 103.5$ W/(m^{2}K). Results obtained by Genetic Algorithm (GA) are also presented

		k		h	
		Mean	Standard deviation	Mean	Standard deviation
Case (a) (zero noise)	GWO	87.2527	0.0350	103.5016	0.0772
	VW-GWO	87.2506	0.0023	103.5006	0.0011
	GA	86.5116	4.5390	105.2710	6.0082
Case (b) (5% noise)	GWO	91.5073	0.0743	108.8373	0.1280
	VW-GWO	91.4993	0.0084	108.8626	0.0267
	GA	81.09318	6.6859	111.7144	5.8421

Figs. 4 and 5 present the mean convergence curves of 10 runs, each run with 10 agents, achieved by GWO, VW-GWO amd GA for case (a) and case (b), respectively. In both cases, the optimization process using VW-GWO tends to converge with much less iterations than GWO. Fig. 4 clearly exhibits the efficiency of VW-GWO, as best fitness quickly drops to zero after more than 50 iterations. After 100 iterations, the best fitness obtained by GWO is still higher than that by VW-GWO. Similar observation is recorded in Fig. 5 for case (b). VW-GWO requires much smaller number of iteration than GWO to
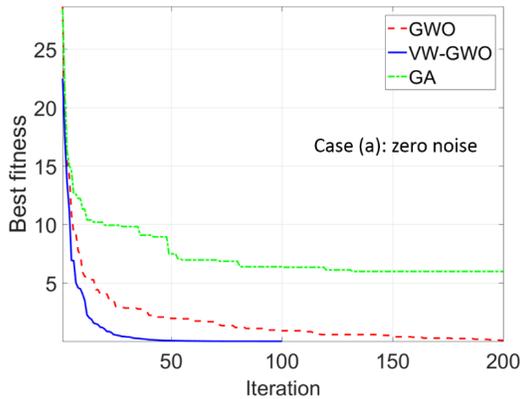
Fig. 4. Convergence curve obtained by GWO and VW-GWO for case (a): zero noise in reference data
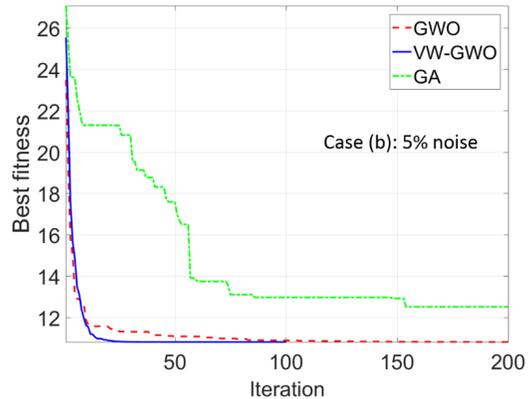
Fig. 5. Convergence curve obtained by GWO and VW-GWO for case (b): 5% noise in reference data

reach convergence. Computational time for each iteration is not much difference between GWO and VW-GWO. Therefore, with higher rate of convergence, there is potential to save elapsed time by using VW-GWO. The number of necessary iterations is not known beforehand. It is possible to define a lower limit for the number of iterations. After that limit, if fitness value (i.e. the value of objective function) repeatedly does not change within many iterations (e.g. 50 iterations), the optimization process can be considered as being converged and thus can be terminated.

5. CONCLUSION AND OUTLOOKS

In this paper, a trained POD-RBF system is coupled with Grey Wolf Optimizer to develop a surrogate model for estimation of thermal parameters. It is demonstrated that the proposed numerical scheme yields reliable output. When there is no noise in reference data, the error rate between predicted thermal parameters and the true ones is almost zero. When noise is included in the reference data, the parameters are predicted with an error rate within the range of noise.

Comparison between two variants of Grey Wolf Optimizer, i.e. the original one (namely GWO) and the improved one (namely VW-GWO) has been conducted. It is shown that by using VW-GWO, the convergence rate of the optimizing process is increased. Therefore, less number of iterations is required and as a result, computational time can be potentially saved.

There are still many issues left open. Improving computational efficiency of the optimization process is a constant demand. For the POD-RBF block, the size of training data would increase with respect to the number of the parameters to be identified. Loosely speaking, if identification of 1 parameter needs N samples, then identification of d parameters would need N^d samples. Special technique is necessary to handle with a large and multi-dimensional data. Experiments could be involved in both the preparation of training data and the collection of reference data. However, a large number of data is usually required for training. Therefore, a numerical data generator might be more practical. On the other hand, the numerical model has to be verified before it can be used for generation of training data. The reference data in practice shall be obtained from measurement. Obviously, the more number of sensors are placed, the more information could be gained. Unfortunately, in most of the cases, the number of sensors cannot be large due to the cost issues. Therefore, it is necessary to optimize the number of sensors and the positions where the sensors are located [34, 35]. This is also an interesting research topic which can be employed together with inverse analysis.

ACKNOWLEDGMENT

REFERENCES

[1] M. N. Özisik and H. R. B. Orlande. *Inverse heat transfer: fundamentals and applications*. Taylor & Francis, (2000).

[2] F. B. Liu. Inverse estimation of wall heat flux by using particle swarm optimization algorithm with Gaussian mutation. *International Journal of Thermal Sciences*, **54**, (2012), pp. 62–69. https://doi.org/10.1016/j.ijthermalsci.2011.11.013.

[3] H. L. Lee, W. J. Chang, W. L. Chen, and Y. C. Yang. Inverse heat transfer analysis of a functionally graded fin to estimate time-dependent base heat flux and temperature distributions. *Energy Conversion and Management*, **57**, (2012), pp. 1–7. https://doi.org/10.1016/j.enconman.2011.12.002.

[4] F. Mohebbi, B. Evans, A. Shaw, and M. Sellier. An inverse analysis for determination of space-dependent heat flux in heat conduction problems in the presence of variable thermal conductivity. *International Journal for Computational Methods in Engineering Science and Mechanics*, **20**, (3), (2019), pp. 229–241. https://doi.org/10.1080/15502287.2019.1615579.

[5] A. E. Segall, D. Engels, and C. Drapaca. Inverse determination of thermal boundary conditions from transient surface temperatures and strains in slabs and tubes. *Materials and Manufacturing Processes*, **27**, (8), (2012), pp. 860–868. https://doi.org/10.1080/10426914.2012.663130.

[6] M. Ebrahimian, R. Pourgholi, M. Emamjome, and P. Reihani. A numerical solution of an inverse parabolic problem with unknown boundary conditions. *Applied Mathematics and Computation*, **189**, (1), (2007), pp. 228–234. https://doi.org/10.1016/j.amc.2006.11.062.

[7] H. L. Zhou, X. H. Zhao, B. Yu, H. L. Chen, and Z. Meng. Firefly algorithm combined with Newton method to identify boundary conditions for transient heat conduction problems. *Numerical Heat Transfer, Part B: Fundamentals*, **71**, (3), (2017), pp. 253–269. https://doi.org/10.1080/10407790.2016.1277915.

[8] F. Mohebbi and M. Sellier. Estimation of thermal conductivity, heat transfer coefficient, and heat flux using a three dimensional inverse analysis. *International Journal of Thermal Sciences*, **99**, (2016), pp. 258–270. https://doi.org/10.1016/j.ijthermalsci.2015.09.002.

[9] Z. Ostrowski, R. Białecki, and A. J. Kassab. Solving inverse heat conduction problems using trained POD-RBF network inverse method. *Inverse Problems in Science and Engineering*, **16**, (1), (2008), pp. 39–54. https://doi.org/10.1080/17415970701198290.

[10] C. Y. Yang. Estimation of the temperature-dependent thermal conductivity in inverse heat conduction problems. *Applied Mathematical Modelling*, **23**, (6), (1999), pp. 469–478. https://doi.org/10.1016/s0307-904x(98)10093-8.

[11] R. Pourgholi, H. Dana, and S. H. Tabasi. Solving an inverse heat conduction problem using genetic algorithm: sequential and multi-core parallelization approach. *Applied Mathematical Modelling*, **38**, (7-8), (2014), pp. 1948–1958. https://doi.org/10.1016/j.apm.2013.10.019.

[12] R. Das and B. Kundu. Direct and inverse approaches for analysis and optimization of fins under sensible and latent heat load. *International Journal of Heat and Mass Transfer*, **124**, (2018), pp. 331–343. https://doi.org/10.1016/j.ijheatmasstransfer.2018.03.059.

[13] H.-L. Chen, B. Yu, H. L. Zhou, and Z. Meng. Improved cuckoo search algorithm for solving inverse geometry heat conduction problems. *Heat Transfer Engineering*, **40**, (3-4), (2019), pp. 362–374. https://doi.org/10.1080/01457632.2018.1429060.

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis. Grey wolf optimizer. *Advances in Engineering Software*, **69**, (2014), pp. 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[15] E. Emary, H. M. Zawbaa, and A. E. Hassanien. Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, **172**, (2016), pp. 371–381. https://doi.org/10.1016/j.neucom.2015.06.083.

[16] S. Eswaramoorthy, N. Sivakumaran, and S. Sekaran. Grey wolf optimization based parameter selection for support vector machines. *COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, (2016). https://doi.org/10.1108/compel-09-2015-0337.

[17] A. Lakum and V. Mahajan. Optimal placement and sizing of multiple active power filters in radial distribution system using grey wolf optimizer in presence of nonlinear distributed generation. *Electric Power Systems Research*, **173**, (2019), pp. 281–290. https://doi.org/10.1016/j.epsr.2019.04.001.

[18] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, and W. Cai. Grey wolf optimizer for parameter estimation in surface waves. *Soil Dynamics and Earthquake Engineering*, **75**, (2015), pp. 147–157. https://doi.org/10.1016/j.soildyn.2015.04.004.

[19] S. Zhang and Y. Zhou. Template matching using grey wolf optimizer with lateral inhibition. *Optik*, **130**, (2017), pp. 1229–1243. https://doi.org/10.1016/j.ijleo.2016.11.173.

[20] S. Zhang, Y. Zhou, Z. Li, and W. Pan. Grey wolf optimizer for unmanned combat aerial vehicle path planning. *Advances in Engineering Software*, **99**, (2016), pp. 121–136. https://doi.org/10.1016/j.advengsoft.2016.05.015.

[21] R. A. Białecki, A. J. Kassab, and A. Fic. Proper orthogonal decomposition and modal analysis for acceleration of transient FEM thermal analysis. *International Journal for Numerical Methods in Engineering*, **62**, (6), (2005), pp. 774–797. https://doi.org/10.1002/nme.1205.

[22] A. Fic, R. A. Białecki, and A. J. Kassab. Solving transient nonlinear heat conduction problems by proper orthogonal decomposition and the finite-element method. *Numerical Heat Transfer, Part B: Fundamentals*, **48**, (2), (2005), pp. 103–124. https://doi.org/10.1080/10407790590935920.

[23] X. Zhang and H. Xiang. A fast meshless method based on proper orthogonal decomposition for the transient heat conduction problems. *International Journal of Heat and Mass Transfer*, **84**, (2015), pp. 729–739. https://doi.org/10.1016/j.ijheatmasstransfer.2015.01.008.

[24] N. N. Minh, N. T. Nha, T. T. Thien, and B. Q. Tinh. Efficient numerical analysis of transient heat transfer by Consecutive-Interpolation and Proper Orthogonal Decomposition. *Science and Technology Development Journal*, **20**, (K9), (2017), pp. 5–14. https://doi.org/10.32508/stdj.v20ik9.1671.

[25] Z. Ostrowski, R. A. Białecki, and A. J. Kassab. Estimation of constant thermal conductivity by use of proper orthogonal decomposition. *Computational Mechanics*, **37**, (1), (2005), pp. 52–59. https://doi.org/10.1007/s00466-005-0697-y.

[26] C. A. Rogers, A. J. Kassab, E. A. Divo, Z. Ostrowski, and R. A. Bialecki. An inverse POD-RBF network approach to parameter estimation in mechanics. *Inverse Problems in Science and Engineering*, **20**, (5), (2012), pp. 749–767. https://doi.org/10.1080/17415977.2012.693080.

[27] M. Bocciarelli, V. Buljak, C. K. S. Moy, S. P. Ringer, and G. Ranzi. An inverse analysis approach based on a POD direct model for the mechanical characterization of metallic materials. *Computational Materials Science*, **95**, (2014), pp. 302–308. https://doi.org/10.1016/j.commatsci.2014.07.025.

[28] S. Khatir and M. A. Wahab. Fast simulations for solving fracture mechanics inverse problems using POD-RBF XIGA and Jaya algorithm. *Engineering Fracture Mechanics*, **205**, (2019), pp. 285–300. https://doi.org/10.1016/j.engfracmech.2018.09.032.

[29] Z. M. Gao and J. Zhao. An improved grey wolf optimization algorithm with variable weights. *Computational Intelligence and Neuroscience*, **2019**, (2019). https://doi.org/10.1155/2019/2981282.

[30] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current Science*, (2000), pp. 808–817.

[31] G. R. Liu. *Meshfree methods: moving beyond the finite element method*. Taylor & Francis, second edition, (2010).

[32] C. H. Thai, V. N. V. Do, and H. Nguyen-Xuan. An improved Moving Kriging-based mesh-free method for static, dynamic and buckling analyses of functionally graded isotropic and sandwich plates. *Engineering Analysis with Boundary Elements*, **64**, (2016), pp. 122–136. https://doi.org/10.1016/j.enganabound.2015.12.003.

[33] S. U. Hamim and R. P. Singh. Taguchi-based design of experiments in training POD-RBF surrogate model for inverse material modelling using nanoindentation. *Inverse Problems in Science and Engineering*, **25**, (3), (2017), pp. 363–381. https://doi.org/10.1080/17415977.2016.1161036.

[34] C. Leyder, V. Dertimanis, A. Frangi, E. Chatzi, and G. Lombaert. Optimal sensor placement methods and metrics–comparison and implementation on a timber frame structure. *Structure and Infrastructure Engineering*, **14**, (7), (2018), pp. 997–1010. https://doi.org/10.1080/15732479.2018.1438483.

[35] D. Dinh-Cong, H. Dang-Trung, and T. Nguyen-Thoi. An efficient approach for optimal sensor placement and damage identification in laminated composite structures. *Advances in Engineering Software*, **119**, (2018), pp. 48–59. https://doi.org/10.1016/j.advengsoft.2018.02.005.

APPENDIX A

The flow chart of the proposed procedure for inverse heat transfer analysis is given in Fig. A.1.

Fig. A.1. Flow chart