

RESEARCH ON THE CROSS-SECTION GENERATING METHOD IN HTGR SIMULATOR BASED ON MACHINE LEARNING METHODS

Zeguang LI^{1*}, Jun SUN¹, Chunlin WEI¹, Zhe SUI¹, Xiaoye QIAN²

¹Institute of Nuclear and New Energy Technology, Collaborative Innovation Center of Advanced Nuclear Energy Technology, Key Laboratory of Advanced Reactor Engineering and Safety of Ministry of Education, Tsinghua University, Beijing, 100084, China

²Business school, Sichuan University, Chengdu, Sichuan, 610065, China

lizeguang@tsinghua.edu.cn, sunjun@tsinghua.edu.cn, weichunlin@tsinghua.edu.cn,
suiz@tsinghua.edu.cn, xyqian@scu.edu.cn

ABSTRACT

With the increasing needs of accurate simulation, the 3-D diffusion reactor physics module has been implemented in HTGR's engineering simulator to give better neutron dynamics results instead of point kinetics model used in previous nuclear power plant simulators. As the requirement of real-time calculation of nuclear power plant simulator, the cross-sections used in 3-D diffusion module must be calculated very efficiently. Normally, each cross-section in simulator is calculated in the form of polynomial by function of several concerned variables, the expression of which was finalized by multivariate regression from large number scattered database generated by previous calculation. Since the polynomial is explicit and prepared in advance, the cross-sections could be calculated quickly enough in running simulator and achieve acceptable accuracy especially in LWR simulations. However, some of concerned variables in HTGR are in large scope and also the relationships of these variables are non-linear and very complex, it is very hard to use polynomial to meet full range accuracy. In this paper, a cross-section generating method used in HTGR simulator is proposed, which is based on machine learning methods, especially deep neuron network and tree regression methods. This method first uses deep neuron networks to consider the non-linear relationships between different variables and then uses a tree regression to achieve accurate cross-section results in full range, the parameters of deep neuron networks and tree regression are learned automatically from the scattered database generated by VSOP. With the numerical tests, the proposed cross-section generating method could get more accurate cross-section results and the calculation time is acceptable by the simulator.

KEYWORDS: HTGR, engineering simulator, cross section, machine learning

1. INTRODUCTION

The modular High-Temperature Gas-cooled Reactor (HTGR) [1] is one type of the Gen-IV advance nuclear reactors, which has been researched and developed specifically in China since 1970s. With the rapid progress of the Shandong Shidao Bay High-Temperature gas-cooled Reactor Pebble-bed Module (HTR-PM) demonstration power plant, the engineering simulation system (ESS) [2] has been studied to

model the pebble-bed type reactor and implemented into the full scope simulator of HTR-PM according to the operator training and licensing purpose.

According with corresponding industry standards, the nuclear power plant simulator in required to perform real-time calculation and achieve simulation results with high fidelity compared to design parameters or operational data in different stages. To balance the accuracy and efficiency in ESS, the reactor physics modules in ESS are normally developed based on simple diffusion (1D or 1+2D) and point kinetics methods. In the simple diffusion calculation, the macro cross-sections on every node are needed to be calculated according to temperature, burnup and other variables. Normally, to increase the calculating efficiency of cross-sections, each kind of macro cross-sections used in ESS is calculated in the form of polynomial function of the concerned variables, the expression of which was finalized by multivariate regression analysis from prepared scattered macro cross-section tables generated by assembly code. This method could achieve fast and acceptable cross-section generating, and has been widely used and tested in LWR ESS.

To meet the increasing needs of accurate simulation, the 3-D diffusion reactor physics module [3] has been implemented instead of simple diffusion module in HTGR ESS, which needs much more cross-section calculations and impossible to use cross-section table interpolation to achieve real-time simulation. However, when implementing the polynomial functions to calculate cross-sections, it occurs different problems from LWR ESS. In HTGR, some new variables, such as buckling, need to be considered when fitting the polynomial functions of cross-sections, and the relationships between these variables and cross-sections are more complicated [4]. Also, some variables in HTGR have larger scope compared with LWR, which makes it much harder to meet full range accuracy using polynomial functions. Thusly, the traditional polynomial method, which is well used in LWR ESS, couldn't achieve the same effects in HTGR ESS.

To overcome the limitations of traditional interpolation and polynomial functions used in the cross-section calculation of 3-D diffusion module of HTGR ESS, a new cross-section calculation method based on machine learning is proposed in this paper. The new method is developed based on deep neural network and regression tree, and could achieve high accuracy of cross-section calculation while fast enough to maintain the real-time feature. In Section 2, the machine learning methods and the proposed cross-section calculation method is generally described. Then, some numerical results, including comparison of cross-section results and 3-D diffusion results, are given in Section 3, which could show the feasibility of the proposed method.

2. DESCRIPTION OF THE METHODS

According to the complex nonlinear patterns of the relationships between reactor variables and macro cross-sections in HTGR ESS, the machine learning methods could be regarded as the most relevant methods to solve this problem. Although machine learning methods have been widely used in solving nonlinear problems, for HTGR cross-section calculation problem with complex curves, it is still very hard to get high accuracies just using simple machine learning methods. In this paper, a new cross-section calculation method based on deep neural networks (DNNs) and tree regression (TR) has been proposed. With the specific combination of these two machine learning methods, the proposed method first uses DNNs to consider the non-linear relationships between different variables and then uses a TR to achieve accurate cross-section results in full range, the parameters of DNNs and TR are learned automatically from the scattered database generated by VSOP. In this section, general descriptions of DNN, TR and also the proposed cross-section calculation method are given.

2.1. Deep Neural Network

DNN [5] is an application, nonlinear with respect to its parameters θ that associates to an entry x and an output $y = f(x, \theta)$. This application f has a particular form that we will precise. DNN could be used for regression or classification. As usual in statistical learning, the parameters θ are estimated from the training samples. In DNN, the basic unit is called artificial neuron. An artificial neuron is a function f_j of the input $x = (x_1, \dots, x_d)$ weighted by a vector of connection weights $w_j = (w_{j,1}, \dots, w_{j,d})$, completed by a neuron bias b_j , and associated to an activation function ϕ , which can be presented as

$$y_j = f_j(x) = \phi(\langle w_j, x \rangle + b_j), \quad (1)$$

normally, several activation functions could be considered, such as the identity function, the sigmoid function, the hyperbolic tangent function, the hard threshold function and the Rectified Linear Unit (ReLU). A deep neural network is a structure composed by several hidden layers of neurons where the output of a neuron of a layer becomes the input of a neuron of the next layer. On the last layer, called output layer, a different activation function as for the hidden layers could be applied depending on the type of problems handed: regression or classification. Fig. 1 shows the structure of a deep neural network. The strength of deep learning lies in the depth (number of hidden layers) of the networks.

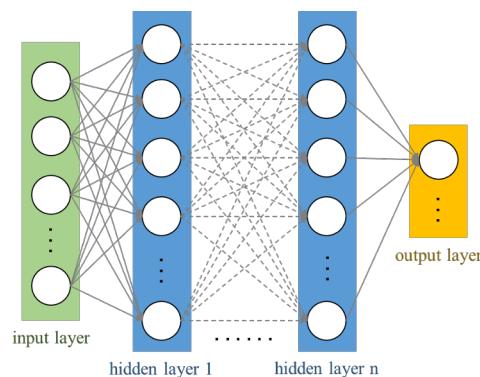


Figure 1. A basic deep neural network structure.

2.2. Tree Regression

All regression techniques contain a single output (response) variable and one or more input (predictor) variables. The general TR [6] may be considered as a variant of decision trees, designed to approximate real-valued functions, instead of being used for classification methods. In which, the decision tree is generated when each decision node in the tree contains a test on some input variable's value, and the terminal nodes of the tree contain the predicted output variable values.

A TR is built through a process known as binary recursive partitioning, which is an iterative process that splits the data into partitions or branches, and then continues splitting each partition into smaller groups as the method moves up each branch. Initially, all records in the training set are grouped into the same partition. The algorithm then begins allocating the data into the first two partitions or branches, using every possible binary split on every field, which is shown in Fig. 2. The algorithm selects the split that minimizes the sum of the squared deviations from the mean in the two separate partitions. This splitting rule is then applied to each of the new branches. This process continues until each node reaches a user-specified minimum node size and becomes a terminal node.

Since the tree is grown from the Training Set, a fully developed tree typically suffers from over-fitting. This over-fitting results in poor performance on real life data. Therefore, the tree must be pruned using

the validation set, which is divided from the prepared data. The cost complexity factor at each step during the growth of the tree and decides the number of decision nodes in the pruned tree are calculated. The cost complexity factor is the multiplicative factor that is applied to the size of the tree.

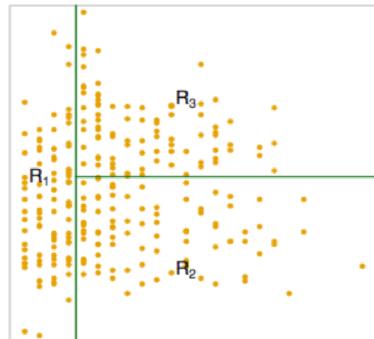


Figure 2. The partition of data in TR.

2.3. Proposed Method for Cross-section Calculation

In the reactor physics module of HTGR ESS, there are 18 types of macro cross-sections in the reactor core and 14 in the graphite reflectors (the same without fission cross-sections), which are

- Diffusion cross-section (for four energy groups): $\Sigma_{tr1}, \Sigma_{tr2}, \Sigma_{tr3}, \Sigma_{tr4}$;
- Absorption cross-section (for four energy groups): $\Sigma_{a1}, \Sigma_{a2}, \Sigma_{a3}, \Sigma_{a4}$;
- Fission cross-section (for four energy groups): $v\Sigma_{f1}, v\Sigma_{f2}, v\Sigma_{f3}, v\Sigma_{f4}$;
- Scattering cross-section (between different energy groups): $\Sigma_{1>2}, \Sigma_{1>3}, \Sigma_{1>4}, \Sigma_{2>3}, \Sigma_{2>4}, \Sigma_{3>4}$.

Considering the special reactor physics characteristics of HTGR, several reactor variables are considered to calculate the cross-sections, which are

- The burnup level: B_u ;
- The temperature of fuel: T_f ;
- The temperature of moderator: T_m ;
- The buckling: DB^2 , and
- The concentration of Xenon: N_{Xe} .

The present method to calculate cross-section in HTGR ESS is polynomial correlation method based on multi-variant linear regression, in which two more variables of square roots of T_f and T_m are considered besides the above variables. For different kinds of cross-sections (total 32 kinds), the regression formulas are pre-trained using large datasets generated by VSOP in the following format:

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4 + b_5 x_5 + b_6 x_6 + b_7 x_7, \quad (2)$$

in which, b_i are the regression coefficients calculated based on large datasets and x_i are the variables mentioned above. Using polynomial correlations to calculate cross-sections could achieve very fast speed, but the accuracies of cross-section results and reactor physics results, such as eigenvalues and temperature coefficients, are not so good.

Based on the machine learning methods, especially DNN and TR, a new method for cross-section calculation in HTGR ESS is proposed. The kernel architecture of proposed method is shown in Fig. 3. In the architecture, first several fully connected layers with artificial neurons of DNN are used to consider the non-linear relationships between different variables, and then a TR network is used following the fully connected layers to get the regression result of cross-section.

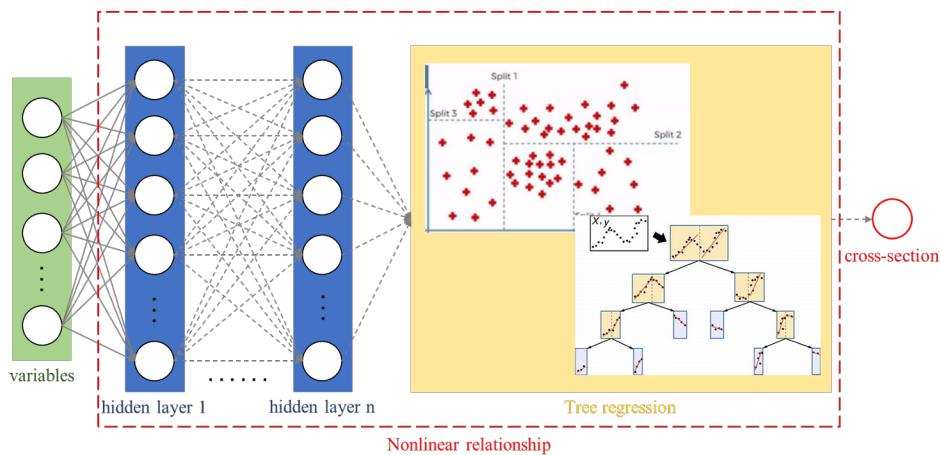


Figure 3. The architecture of proposed method.

The parameters of the DNN and TR in the proposed method are trained according to the cross-section datasets prepared by VSOP, with the aim to minimize the mean squared error (MSE) loss function:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^N (\mathbf{y}_n - \hat{\mathbf{y}}_n(\mathbf{x}_n, \theta))^2, \quad (2)$$

in which, θ means the networks parameters, \mathbf{x} , \mathbf{y} and $\hat{\mathbf{y}}$ mean the input reactor variables, cross-section result of VSOP and output of the network. In order to minimize the loss function $L(\theta)$, a stochastic gradient descent algorithm is used, in which the backpropagation algorithm is used to calculate the gradient. When training the networks, to avoid overfitting we randomly split the prepared labeled datasets partly as the training datasets and partly as the validating datasets. The training datasets are used to estimate the parameters of networks, and the validating datasets are used to assess the performance of trained network. Only when the loss function values are both very small on training and testing datasets, we could think that the networks are well trained.

To increase the accuracy of the algorithm, the values of input data (including data used for training and new data for diagnosis) need to be normalized. This is because normalization can help reduce the chance of getting stuck in local minima (i.e., not global minima among the several minimum points of error during the learning process) owing to different scales of variables. For the parameter with the values almost in the same order of magnitude, a simple min-max scaling method could be applied to adjust the values. If the values of a certain parameter have a large range of orders of magnitude, then before using min-max scaling method, we should apply logarithm operation to the values. After normalization using logarithm operation and min-max scaling, the values are scaled to range from 0 to 1.

3. NUMERICAL RESULTS

To model the method, a desktop computer with the following hardware configurations is used: NVIDIA GeForce GTX 1080 8GB GPU, 2×Intel Xeon V5-2690 v3 CPU and 256 GB memory, and the **TensorFlow**, **Scikit** library based on Python 3.5.3 is used to implementing the DNN and TR. When training the networks, we use the **adadelta** method to accelerate the gradient descent progress, and the training batch size and epochs are set as 32 and 20. For training and validating data splitting, we use 80%/20% random splitting strategy, which means 80% of datasets are randomly picked for training the network parameters and the other 20% datasets are predicted by the trained networks and compared with matched results to validate the networks. To show the effects of the proposed method, the interactions polynomial regressions are also used for the comparison. In this section, first the cross-section results of

the proposed method are shown and compared with present polynomial and interactions polynomial results, then the reactor physics results are given and compared.

3.1. Cross-section Results and Comparison

First, the cross-section results of proposed method, present polynomial regression used in HTGR ESS and interaction polynomial regression. For the proposed machine learning method, a DNN with 2 hidden layers is used in which each layer has 20 artificial neurons, and for the TR the minimum leaf size is chosen as 4. All these three methods are well trained with 5355 scattering datasets for each of different cross-sections, and the minimum MSEs have been achieved for the trainings.

Table I gives the MSE comparison results for these three methods to calculate different cross-sections in reactor core, from which we could easily notice that the proposed machine learning method could achieve the minimum MSE results, which means the proposed method has better calculation performance than present polynomial and interaction polynomial method. Also in Fig. 4, some of the cross-section comparisons are shown. In Fig. 4, the blue dots are the reference cross-section results prepared by VSOP and yellow dots are the calculated results by different methods, from which we could easily notice the accuracies of the proposed method are much better than the polynomial methods.

Table I. MSE results of three methods.

Cross-section	Proposed method	Present polynomial	Interaction polynomial
Σ_{tr1}	1.97918E-09	2.88830E-06	6.25074E-07
Σ_{tr2}	3.63795E-10	5.01554E-06	8.93022E-07
Σ_{tr3}	1.61840E-09	5.28031E-06	9.39802E-07
Σ_{tr4}	3.80998E-09	5.73372E-06	9.93249E-07
Σ_{a1}	1.99458E-16	4.72362E-13	1.77864E-13
Σ_{a2}	1.90543E-14	1.43143E-11	8.64436E-12
Σ_{a3}	2.54380E-12	7.34186E-10	1.80614E-10
Σ_{a4}	7.81759E-11	3.96015E-09	2.34191E-09
$v\Sigma_{f1}$	4.90047E-16	8.64009E-13	2.22637E-13
$v\Sigma_{f2}$	7.65116E-15	1.05433E-11	5.86950E-12
$v\Sigma_{f3}$	1.74118E-13	2.42471E-10	5.54669E-11
$v\Sigma_{f4}$	1.06900E-10	1.37088E-08	4.09887E-09
$\Sigma_{1>2}$	5.70958E-11	5.49774E-08	3.01623E-08
$\Sigma_{1>3}$	2.21434E-26	4.42258E-23	2.58395E-23
$\Sigma_{1>4}$	2.13529E-33	3.27196E-30	1.45560E-30
$\Sigma_{2>3}$	2.72365E-10	6.49937E-08	4.50464E-08
$\Sigma_{2>4}$	1.12478E-39	3.16151E-37	1.71929E-37
$\Sigma_{3>4}$	2.51926E-11	7.36492E-08	4.56098E-08

Also, the calculation efficiencies of proposed machine learning method and polynomial method are compared. When using the architecture mentioned above, the cross-section calculating speed of proposed method is around 320,000 / sec, and the speed of polynomial method is around 395,000 / sec. It is because

the DNN needs more calculation time than polynomial calculation, but the speeds of these two methods could all be accepted for real-time calculation of ESS.

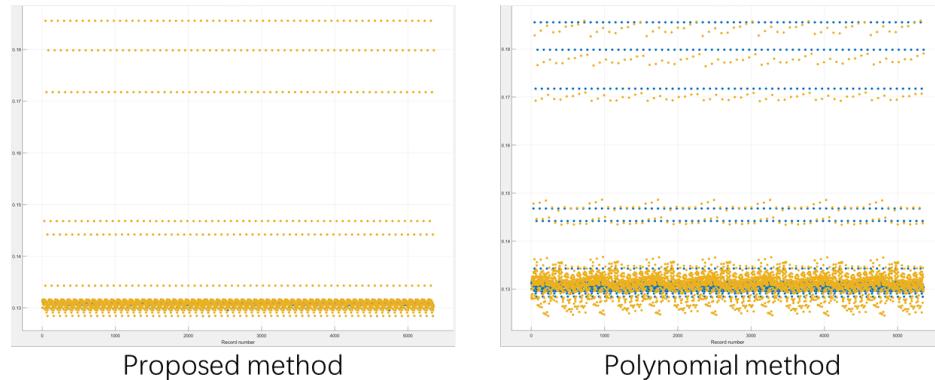


Figure 4. Comparison of proposed and polynomial method results of Σ_{tri} .

3.2. Reactor Physics Results and Comparison

Besides the direct comparison of cross-section results, the integral results of reactor, such as eigenvalues (k_{eff}) and temperature coefficients, are also compared to test the proposed machine learning method. The proposed method was implemented into the 3-D diffusion module of HTGR ESS, and the k_{eff} and temperature coefficient results are calculated. First, a normal operation state with steady temperature distribution is chosen as reference state, and the fuel and moderator temperatures are changed with +50K, +20K, -20K and -50K independently, which means totally 9 states are chosen for the comparison as listed in Table II. Fig. 5 shows the comparison of k_{eff} results in different states calculated by VSOP (reference results), polynomial method and proposed method, and Fig. 6 shows the comparison of fuel and moderator temperature coefficients calculated by VSOP, polynomial method. From these results, we could notice that the proposed method could achieve much more accurate integral reactor results than present polynomial method in 3-D diffusion module of HTGR ESS.

Table II. List of comparison states.

State	Ref.	T_f+50K	T_f+20K	T_f-20K	T_f-50K	T_m+50K	T_m+20K	T_m-20K	T_m-50K
No.	1	2	3	4	5	6	7	8	9

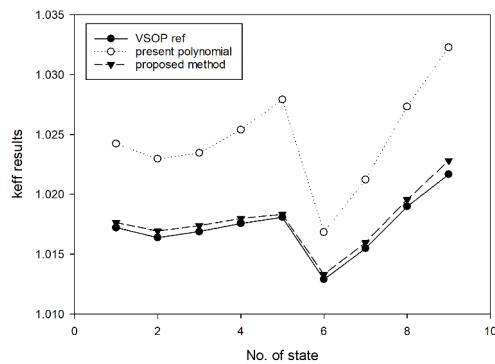


Figure 5. Comparison of k_{eff} results in different states.

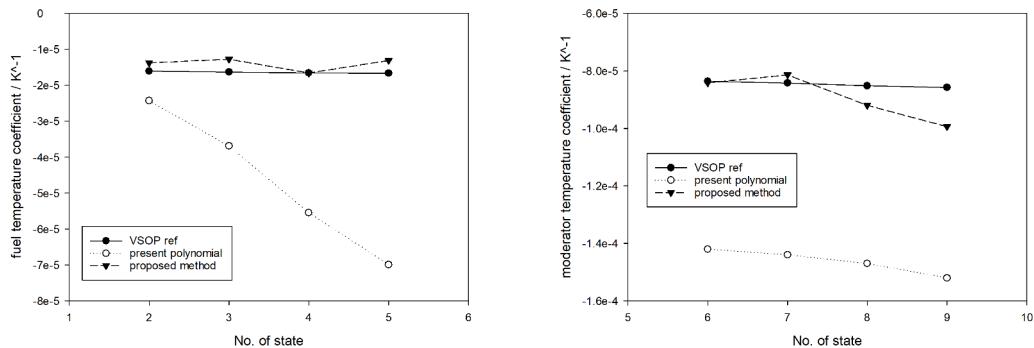


Figure 6. Comparison of temperature coefficient results in different states.

4. CONCLUSIONS

The relationships between cross-sections and reactor variables are more complex in HTGR and hard to get accurate results using traditional polynomial regressions. To calculate cross-sections fast and accurate for 3-D diffusion module in HTGR ESS, a new method based on machine learning, especially DNN and TR, is proposed in this paper. The proposed method uses prepared cross-section datasets from VSOP to train the networks, and could calculate the cross-sections accurately and also keep the real-time feature of ESS. The cross-section results and reactor integral results are compared with proposed method and traditional polynomial method, and the feasibility and effects of the new method has been proved. In the future work, this method will be tested using the ESS for operation problems coupled with thermal-hydraulics and system analysis.

ACKNOWLEDGMENTS

The research was funded by Chinese Major National S&T under Grant number ZX069, Project 11775126 /11545013/71872117 by National Natural Science Foundation of China, Young Elite Scientists Sponsorship Program by CAST (2016QNRC001), and Tsinghua University Initiative Scientific Research Program.

REFERENCES

1. Zhang, Z., Dong, Y., et al., “The Shandong Shidao Bay 200 MWe High-Temperature gas-cooled Reactor Pebble-bed Module (HTR-PM) demonstration power plant: an engineering and technological innovation,” *Eng.*, **2**, pp. 112-118 (2016).
2. Sui, Z., Sun, J., et al., “The engineering simulation system for HTR-PM,” *Nucl. Eng. Des.*, **271**, pp. 479-486 (2014).
3. Sui Z., Sun J., Wei C., et al., “Coupled Neutron Dynamic and Thermal Hydraulic Models of the HTR Simulator”, *Proceedings of the 2013 21st International Conference on Nuclear Engineering*, 2013.
4. C. Wei; Z. Sui; et al. “Application of improved quasi-static method to high temperature gas cooled reactor simulator,” *Nuc. Eng. Des.*, **271**, pp. 337-340 (2014).
5. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, **323(6088)**, pp. 533–536 (1986).
6. W. Y. Loh. “Classification and Regression Trees,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **1(1)**, pp. 14-23 (2011).