

**Simulation of neutron diffusion and transient analysis of MSR\***ZHU Liang (朱亮),<sup>1,2</sup> PU Peng (蒲鹏),<sup>1</sup> DU Sha (杜莎),<sup>1</sup> and HAN Ding-Ding (韩定定)<sup>1,†</sup><sup>1</sup>*School of Information Science and Technology, East China Normal University, Shanghai 200241, China*<sup>2</sup>*Shanghai Institute of Applied Physics, Chinese Academy of Sciences, Shanghai 201800, China*

(Received October 30, 2013; accepted in revised form December 18, 2013; published online March 20, 2014)

Molten salt reactor (MSR) is a potential nuclear power reactor of Generation IV. The working process of the primary loop of an MSR is studied in this paper. A physical model is established to describe the coupled heat transfer for the MSR core channels, the temperature negative feedback and the neutron characteristics. The simulation code, NDP1D, has been developed with the object-oriented method, conducting the neutron diffusion and transient analysis in a parallel way. The simulation data and diagrams of neutron, power, flow rate and temperature can be obtained via graphical user interface. The simulation results can be used for further study on MSRs of larger dimensions and more complicated geometry.

Keywords: Molten salt reactor, Neutron diffusion, Transient analysis, Simulation software

DOI: [10.13538/j.1001-8042/nst.25.020601](https://doi.org/10.13538/j.1001-8042/nst.25.020601)

**I. INTRODUCTION**

The Generation IV International Forum (GIF) on nuclear energy aims at developing technologies to achieve safety performance, waste reduction and increased proliferation resistance, and to provide a nuclear energy option that is economically competitive [1]. Of the six future nuclear power plants under the framework of GIF, only the Molten Salt Reactor (MSR) is of the liquid fuel type. The concept of MSR was initially proposed by Oak Ridge National Laboratory (ORNL) [2, 3] where the feasibility was proved by both theories and experiments. In 2011, Chinese Academy of Sciences launched the project of Thorium Molten Salt Reactor Nuclear Energy System (TMSR). As part of the pre-study project, the molten salt reactor transient analysis considers key parameters (the power, flow rate, temperature and the controlling orders) and the physical properties of the core (flow rate of the initial zero-power and low-power mode, temperature and the primary loop pressure). The transient analysis software also judges status of the reactor via different parameters. Therefore, the simulation methods are of much concern.

Current researches on MSR mainly focus on finding different descriptive approaches of the reaction procedures. Lapenta *et al.* [4] analyzed neutronics in a fluid fuel system through a point-kinetics model. Dulla *et al.* [5, 6] used the quasi-static method, which considered a prescribed velocity field in one direction without taking the temperature cross-section feedback into account. Yamamoto *et al.* [7, 8] and Suzuki *et al.* [9] analyzed the steady-state and transient state for a small molten salt reactor by coupling the neutron diffusion equations with the heat transfer in fuel salt and graphite. Krepel *et al.* used the DYN1D-MSR [10] and DYN3D-MSR [11] codes for solving the two-group neutron diffusion equation with a point kinetics model, and models of delayed

neutrons drift and heat release distribution were included. The prompt fission neutron source was calculated by means of the same nodal flux expansion method. Part of the neutrons was released with a time delay when the precursor core drifted with the liquid fuel along axial direction. An extended delayed neutrons model was developed. Transient simulating of 3D neutronics and parallel channel thermal hydraulics were used.

Lecarpentier *et al.* [12] developed the Cinsf1D program, which replaced MSRE at Oak Ridge to become the European benchmark of the transient simulation, and to study the AMSTER system [13]. Both steady-state and transient state could be simulated by the Cinsf1D program. After modeling the salt fuel and the graphite in the reactor, the axial and radial power distribution of the core were calculated by the DRAGON program [14]. Based on the diffusion theory with two energy groups, the neutron flux and the distribution of delayed neutron precursors were described in neutronics according to the cross sections. The thermal negative feedback depending on the cross sections was related to the temperatures of various mediums. Thermo hydraulics could be described by the energy balance between the salt fuel and graphite. A simplified geometric model represented the typical core channel in the molten salt reactor. Also, a point kinetic model was included, where graphite temperature was a single average value, and the relationship between the temperature of the molten salt and the height of the core was linear.

In this paper, we report our work on simulating neutron diffusion and transient analysis in an MSR. Optimization of the model based on Cinsf1D will be discussed in Sec. II, with a focus on the transient analysis, which is separated from the point kinetic model. The implementation of object-oriented programming is discussed in Sec. III. The software architecture is improved to meet the standards of software-engineering, high-performance computing and management via graphical user interface, and to acquire the simulation data and diagrams in visualized forms. The results are shown in Sec. IV. In Sec. V, we conclude that the neutron diffusion and transient analysis of the MSR are simulated in a precise and efficient way.

\* Supported by the National Nature Science Foundation of China (Nos.11075057, 11035009 and 10979074)

† Corresponding author, [ddhan@ee.ecnu.edu.cn](mailto:ddhan@ee.ecnu.edu.cn)

## II. THEORETICAL MODEL OF THE MSR PROCESS

A vertical-section diagram of the primary loop is shown in Fig. 1. The molten reactor core consists of graphite reflector. The molten salt fuel contains beryllium fluoride, lithium fluoride, zirconium fluoride, and uranium or thorium fluoride. With the fissile materials and convertible molten salt at 600 °C, the molten salt fuel exits from the core. The fuel salt flows sequentially from the left of the core to Pipe 1, next to the heat exchanger, then to Pipe 2, and finally back to the core. The molten salt reactor core is a cylindrical cavity with 199 parallel hexagonal graphite elements, each of which has a circular coolant channel. Each channel has a fuel passage in the middle. The calculation of the reactor core is divided into three zones, with  $h_1$ ,  $h_2$ , and  $h_3$  describing the length of each zone,  $r_g$  being the channel diameter and  $r_f$  being the inner diameter. The liquid salt serves as both the coolant and fuel. The graphite moderator in the reactor is heated by the neutrons and gamma rays produced in the fission reaction process. The heat between the graphite moderator and coolant is exchanged through the heat exchanger. Controlling of the reaction is modeled as follows.

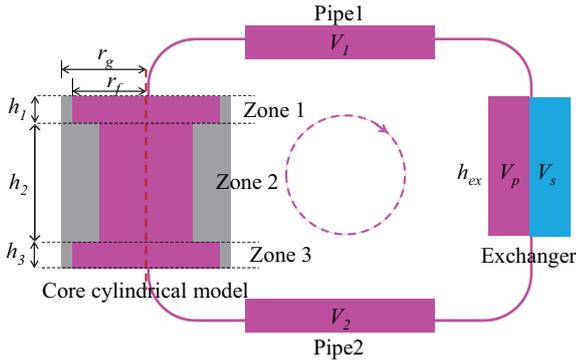


Fig. 1. (Color online) The vertical-sectional diagram of the primary loop: the fuel salt flows as the dashed-circle is indicated.

### A. Neutron Diffusion Modeling

In the neutron diffusion modeling, the neutrons are of two energy groups (the fast and thermal neutron energy groups) and one spatial dimension ( $z$ ). The neutron energy groups are cut at 1.8 eV. The flux of the molten salt reactor is calculated by the thermal spectrum. It has been shown that the axial direction ( $z$ ) and radial direction ( $x$ ) are relatively decoupled. The main object of the model is to compare the influence of liquidity for the delayed neutron precursors, which are produced only in the core, drifted along axial ( $z$ ) with the MSR core movement. The life spans of these neutrons are much shorter than the molten salt fuel cycle, so the neutrons are not affected by the velocity field [15, 16].

The neutron and precursors dynamics equations in the fuel material of molten salt reactor are derived from the particle

conservation equations as follows:

$$\left\{ \begin{array}{l} \frac{1}{\nu_1} \frac{\partial \phi_1}{\partial t} = \frac{\partial}{\partial z} \left( D_1 \frac{\partial \phi_1}{\partial z} \right) \\ \quad + (1 - \beta) \left( \nu_1 \sum f_1 \phi_1 + \nu_2 \sum f_2 \phi_2 \right) \\ \quad - \left( \sum \alpha_1 + \sum r + D_1 B^2 \right) \phi_1 + \sum_{i=1}^6 \lambda_i C_i, \\ \frac{1}{\nu_2} \frac{\partial \phi_2}{\partial t} = \frac{\partial}{\partial z} \left( D_2 \frac{\partial \phi_2}{\partial z} \right) - \left( \sum \alpha_2 + D_2 B^2 \right) \phi_2 \\ \quad + \sum r \phi_1, \\ \psi = \nu_1 \sum f_1 \phi_1 + \nu_2 \sum f_2 \phi_2, \\ \frac{\partial C_i}{\partial z} = \beta_i \psi(z, t) - \lambda_i C_i - \frac{\partial V(z, t) C_i}{\partial z}, \\ z \in [0, H], t > 0, i = 1, \dots, 6. \end{array} \right. \quad (1)$$

where,  $i$  is the delayed neutron group number.  $V_{j=1,2}$  indicates the neutron speed of group  $j$ , with  $j = 1$  being the fast neutron energy group and  $j = 2$  the thermal neutron energy group.  $\phi_j(z, t)$  is the neutron flux of group  $j$ ,  $t$  is the time,  $D_j(z, t)$  is the diffusion coefficient of group  $j$ .  $\beta_j$  is the proportion of emitted heat from the delayed neutron of group  $i$ , and  $\beta$  is the sum of all  $\beta_j$ ,  $\sum f_j(z)$  and  $\sum \alpha_j(z)$  are the macroscopic neutron fission cross-section and the macroscopic neutron absorption cross-section of the group  $j$ , respectively;  $\sum_r(z)$  is the neutron diffusion cross-section from the fast neutron group towards the thermal group;  $B^2$  is the radial leakage coefficient;  $\lambda_j$  is the delayed neutron decay constant of group  $i$ ;  $C_j(z, t)$  is the linear precursor concentration of group  $i$ .  $\psi(z, t)$  is the number of neutrons emitted by fission per unit height of the reactor; and  $V(z, t)$  is the molten salt speed vector. Due to the long decay time of the delayed neutrons, they are supposed to drift with the molten salt, which means that the pioneer nuclear has the same speed with the molten salt without considering the interaction of them.

### B. Thermal Negative Feedback

The diffusion cross-section can be written as a function of  $z$  and temperature. Thus, according to the diffusion theory, the cross-section as a function of  $z$  at a certain temperature can be calculated through the interpolation using standard cross-section data. Thus, Eq. (1) can be solved to obtain the flux and power released. The reactivity is affected by temperature and balance configuration of salt and graphite, and the control rod depth, but preset data can be prepared to evaluate the diffusion process simulation.

### C. Thermal Hydraulics

The calculation is divided into two parts, the core and the heat exchanger, using lumped parameter model.

The core thermal hydraulics calculation is calculated in two steps: (1) regarding the core graphite and fuel salt as heat sources and calculating the amount of heat generated; and (2) calculating the heat transferred by the heat source. The power generated is calculated by  $P_t = \kappa \left( \sum_{f_1} \phi_1 + \sum_{f_2} \phi_2 \right)$ . The amount of power emitted into the graphite, denoted as  $P_g$ , is calculated by the effective heat transfer coefficient [17, 18]. The power released into salt is denoted as  $P_s$ . The other part of the heat released is generated by delayed neutrons which are reduced by the radioactivity in the salt fuel. Thus, the calculation of heat source is expressed as Eq. (2). Then the heat transport calculation is a simple fixed heat source problem and will not be discussed in this paper.

$$\begin{cases} P_s(z, t) = (1 - \xi)P_t(z, t) + \sum_i \lambda_{th,i}(1 - \xi_i)C_{th,i}(z, t), \\ P_g(z, t) = \xi P_{total}(z, t) + \sum_i \lambda_{th,i}\xi_i C_{th,i}(z, t), \\ \frac{dC_{th,i}}{dt}(z, t) = \beta_{th,i}P_t(z, t) - \frac{\partial V(z, t)C_{th,i}(z, t)}{\partial z} \\ \quad - \lambda_{th,i}C_{th,i}(z, t), \end{cases} \quad (2)$$

where,  $\xi_i = \exp(-\lambda_i \Delta t)$ ;  $C_{th,i}(z, t)$  is the density per unit length for delayed precursors of group  $i$ ; and  $\lambda$ ,  $\beta$  and  $V$  are of the similar definitions to those in Eq. (1).

The heat exchange between the salts in the primary and secondary loops is described by thermal convection equations, taking into account the heat emission due to the decrease of delayed thermal precursors. The flow rate  $D$  is related with volume  $V$ , temperature  $T$ ,  $z$  and time  $t$  as:

$$\begin{aligned} \frac{dT_{j,1}}{dt} &= \sum_k \lambda_{th,k} C_{th,k} \frac{L}{\rho_1 C_{p1} V_{j1}} - \frac{hS}{\rho_1 C_{p1} V_{j1}} (T_{j,1} - T_{j,2}) \\ &+ D_{j,1} \frac{L}{V_{j1}} \frac{\partial T_{j,1}}{\partial t}, \end{aligned} \quad (3)$$

where footnote 1 stands for the primary loop and 2 for the secondary loop.  $\rho$  is salt density, and  $c_p$  is the heating capacity at constant pressure,  $S$  is the exchange surface, and  $h$  is a coefficient of exchange per unit surface area. The total salt flow in the primary loop is calculated by  $D(t) = \sum_j D_{j,1}(t)$ , ( $j = 1 \rightarrow 6$ ).

Between the core outlet and the inlet of heat exchanger, the fuel salt passes a high-temperature zone without heat transfer, namely Pipe 1 in Fig. 1; similarly, between the bottom outlet of the heat exchanger and the core inlet the fuel salt passes a low-temperature zone, namely Pipe 2. Therefore, the heat exchanger is calculated in three parts: above the heat exchanger, the heat exchanger and below the heat exchanger. Each of the parts is calculated on the assumption that temperatures of the boundary and precursor concentrations in all parts are continuous.

### III. THE REALIZATION OF THE SIMULATION SOFTWARE

According to the physical model and the geometric model of the MSR, simulation code Neutron Diffusion Program for 1D (NDPID) for one-dimensional neutron diffusion and transient analysis was finished with the following features:

- To analyze and design the structure of the coupled program of neutron diffusion and transient analysis. Based on physical function of the neutron diffusion process, design the data structure and file operations in a hierarchical and modular way;
- To achieve the function of each module and to complete optimization of the program algorithm and computational efficiency;
- To transplant the original code into an architecture suitable for parallelization;
- To reserve open programming interface for further improvements the transient analysis program on multi-group and complicated geometry.

NDPID is written in C++, the object-oriented language, combined with the linear algebra numerical library ublas (part of the library Boost, launched by the C++ Standards Committee Library Working Group members), shared memory parallel library OpenMP, as well as the cross-platform graphical user interface framework tools Qt.

#### A. Calculation Process of NDPID

The NDPID is divided into critical and transient calculations. Their flow charts are shown in Figs. 2 and 3, respectively.

The salt fuel temperature along the axis  $z$  is calculated. The macroscopic cross-section is linearly interpolated to calculate the thermal characteristics of molten salt and the graphite by adjusting the radial leakage and the uranium concentration and by using the input table of reaction cross-section data. At a fixed temperature, the neutron flux equations containing a concentration of delayed neutrons are resolved, and the flux is normalized by the core power. In transient calculations, the depth and speed of control rod is considered. Flow rate, coupled with heat exchange rate, is determined by the status of reactor. And the input table of reaction cross-section data is linearly interpolated to get the macroscopic cross-section at time  $t$ .

Then the neutron flux and delayed neutron precursors concentration at the next time step are calculated by the cross-section and flux at time  $t$ . By using the distribution of temperature and the power released in the time interval to calculate temperature of the core salt, temperatures of the graphite and heat exchanger are calculated at the next moment. Considering situations where some more vectors, matrix operations,

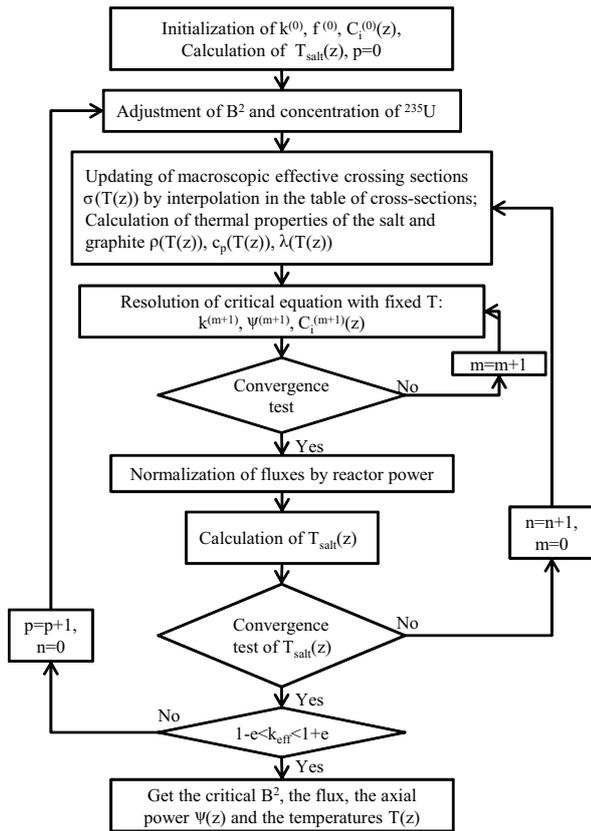


Fig. 2. Flow chart for critical calculation.

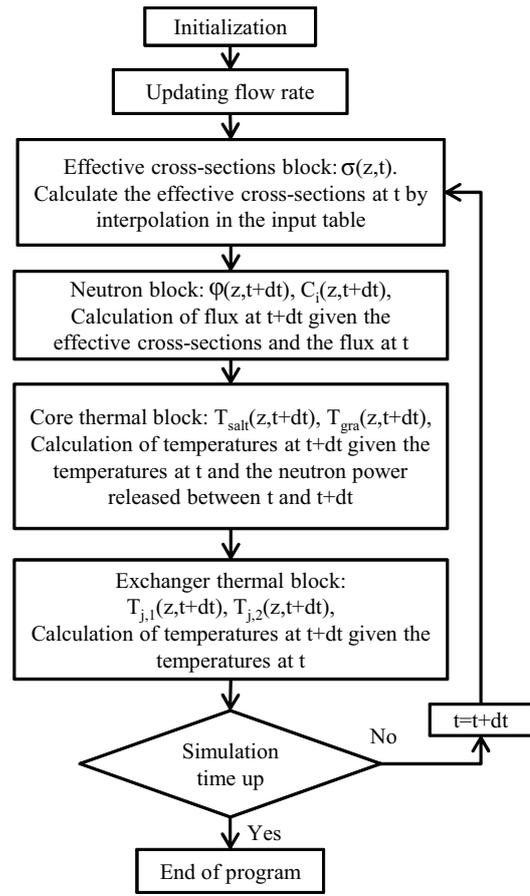


Fig. 3. Flow chart for transient calculation.

file reading and writing operations are involved in the calculation and frequently repeated calls, a certain amount optimization and parallelization have been done to adapt the multi-dimensional data simulation.

### B. Optimization of NDPID

Based on the Cinsf1D code, the program is optimized in three aspects: functionality, syntax and data structures.

- The calculation of diffusion model has been separated according to functionality into several relatively independent modules, which can be tested separately. Coupling of the program is reduced to improve the readability, being conducive to expansion and the second revision;
- With the unique pointer characteristics in C++, the data transfer procedure of the program is optimized to avoid the operations in Cinsf1D written in Fortran, which only supports transferring the real parameters instead of formal parameters;
- The stringstream object of C++ is used in NDPID to simplify the type and upgrade file reading and writing to the memory-level and object-level, hence the reading speed and security is greatly improved.

### C. Parallelization of NDPID

For one-dimensional case, the code can be run on just a single computer or one node of a computing cluster, so it is parallelized using shared memory model, i.e. OpenMP. The parallel execution of the model is the Fork-Join. With time analysis, physical models and code analysis as well as the OpenMP parallelization, the performance of this program has been improved and CPU load can be equalized.

#### 1. Parallel analysis of NDPID

Combined with the above physical model, parameters of the model and calculation involved have been analyzed:

- a. The mathematical equations resolution: MATHFONC :: SOLVETSYS, MATHFONC :: SOLVES\_SYSTEM are respectively the LU decomposition and the main-element elimination method, solving the linear equations  $AX = Y$ . In both methods, the former can be paralleled in the process of back substitution; the latter can be in the elimination process of the primary line to other lines. Due to the usage of lower proportion of LU decomposition algorithm, smaller size of the one-

TABLE 1. Upgraded FECs threshold consistency test

	Number of simultaneously active cores <sup>a</sup>	
	1	2
Before	96.1%	3.9%
After	79.8%	20.2%

<sup>a</sup> Tested on a 2-cored computer

dimensional program, and the larger cost of parallelization, we have considered the main-element elimination method to get effective parallelization.

b. File reading and writing: the program inputs data from the neutron, thermo, transient, geometry and other files; for outputing codes, results are written to some data files. Since the reading and writing of files does not interfere with each other, the CPU can exchange data with the cache in parallel, and then read and write the hard disk to reduce the waiting time between peripherals. On the other hand, file reading and writing is in parallel with other calculations in order to effectively use the waiting time of reading and writing.

c. The main cycle in transient simulation: pipeline parallelism is a better choice concerned for continuous cycle of code. Because the loop of the transient simulation in main program contains a large number of interdependent variables which are in serial execution and each cycle is highly dependent on the calculation results of the last cycle, it is very difficult to parallelize and the model and algorithm modified should be sought to accommodate parallelization.

2. Results of parallelization

The CPU time of the optimized code is 37.84 s, whereas it was 51.32 s before. In comparison, the speed-up ration is  $S \approx 1.36$ . Although it does not reach linear acceleration,  $p/\log 2p \leq S \leq p$ , yet it has been greatly improved. For current scale of the code, there are two trends for the parallelization results: when only a small amount of parallelization is implemented, the calculation speed gets higher, but CPU load is unbalanced, as shown in Table 1; when the parallelized part is greater, the calculation speed is not significantly increased, or even gets slower, but CPU load is more balanced. The reason is that the large serial part is unable to be parallelized. Also, the small code scale is of high cost of parallelization. A possible way to improve the application performance of the code is to increase the number of threads based on Amdahl's law [19], or to increase workload according to Gustafson's Law [20]. Another way of parallelization using GPU [21], specifically CUDA [22], will also be considered in future work.

IV. THE SIMULATING RESULTS OF NDPID

The graphical interface of NDPID is written in Qt, a C++ cross-platform graphical user interface development framework, as shown in Fig. 4. The diagrams displayed are generated by Root codes, which are integrated into NDPID.

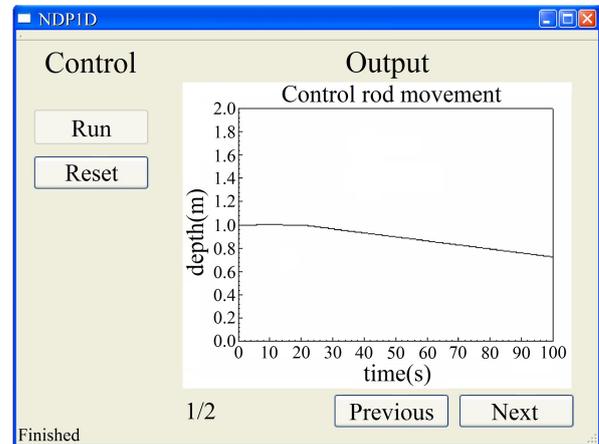


Fig. 4. (Color online) User interface of NDPID, showing the change of control rod depth.

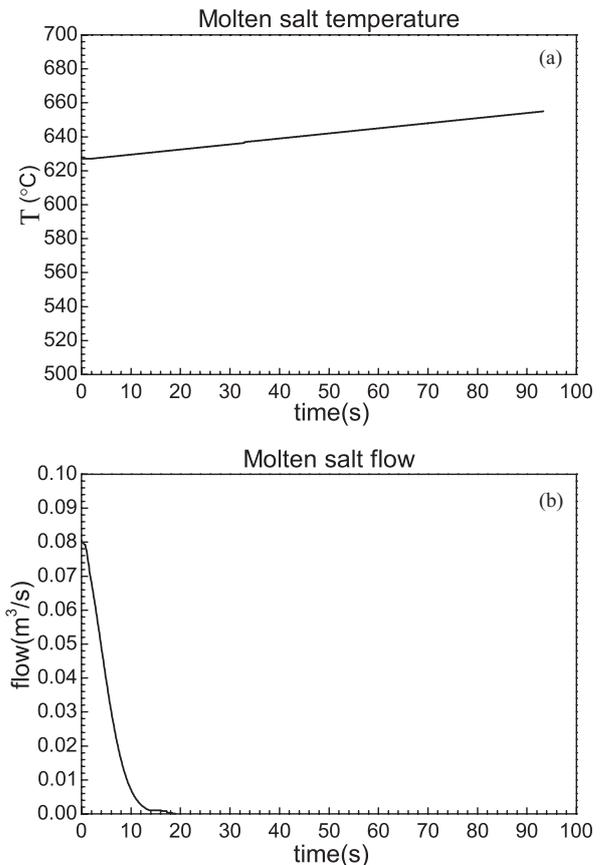


Fig. 5. Constant power shutdown with (a) increasing temperature of molten salt and (b) rapid decrease of molten salt flow rate.

An important simulating process is reactor shutdown. When the reactor is shut down, the power of the reactor goes from the level of operation down to the neutron source level. For validation, a benchmark test, the shutdown of constant power, is performed, and the results are shown in Fig. 5, with both the molten salt flow rate and temperature mainly tested. Also, the control rod depth is shown in Fig. 4. The simulation results of NDP1D program consists with the calculation results of analytical method.

## V. CONCLUSION

The simulation of neutron diffusion and transient analysis of the molten salt reactor has been discussed in this paper, which established an effective physical and geometric model and built a theoretical foundation for the development of the one-dimensional neutron diffusion and transient analysis calculation software NDP1D. The optimized program has reduced the degree of coupling among modules and the redundancy, in line with the software engineering. On the other

hand, the program's performance has been improved and the hardware load has been more balanced after parallelization. With the help of simulation software NDP1D, multiple results have been acquired, such as the simulation of constant power shutdown, including changes of molten salt flow rate, temperature and control rod depth. Although the software NDP1D we developed is just one-dimensional simulation, and some conditions are set close to the ideal states of transient analysis, the studying methods and framework of the software as well as the reactor physics and thermal coupling model can be the basis for follow-up projects to further explore transient analysis on molten salt reactors with higher dimension and more complicated geometry.

## ACKNOWLEDGMENTS

The authors acknowledge the support from CAI Xiang-Zhou, GUO Wei, ZHAN Zhi-Chao, YU Ming-Xiang and SHI Hong-Li.

- 
- [1] DOE. A Technology Roadmap for Generation IV Nuclear Energy Systems. Issued by the US DOE, 2003.
  - [2] Bettis E S, Schroeder R W, Cristy G A, *et al.* Nucl Sci Eng, 1957, **2**: 804–825.
  - [3] Rosenthal M W, Kasten P R, Briggs R B. Nucl Appl Technol, 1970, **8**: 107–117.
  - [4] Lapenta G, Mattioda F, Ravetto P. Ann Nucl Energy, 2001, **28**: 1759–1772.
  - [5] Dulla S, Ravetto P, Rostagno M M. Ann Nucl Energy, 2004, **31**: 1709–1733.
  - [6] Dulla S and Ravetto P. Nucl Sci Eng, 2007, **155**: 475–488.
  - [7] Yamamoto T, Mitachi K, Ikeuchi K, *et al.* Heat Transfer Asian Res, 2006, **35**: 434–450.
  - [8] Yamamoto T, Mitachi K, Suzuki T. JSME Int J B-Fluid T, 2005, **48**: 610–617.
  - [9] Suzuki N and Shimazu Y. J Nucl Sci Technol, 2008, **45**: 575–581.
  - [10] Krepel J, Grundmann U, Rohde U, *et al.* Ann Nucl Energy, 2005, **32**: 1799–1824.
  - [11] Krepel J, Rohde U, Grundmann U, *et al.* Ann Nucl Energy, 2007, **34**: 449–462.
  - [12] Lecarpentier D. Ph.D. Thesis, National Academy of Arts and Trades, 2001.
  - [13] Lecarpentier D and Carpentier V. Nucl Sci Eng, 2003, **143**: 33–46.
  - [14] Delpech M, Dulla S, Garzenne C, *et al.* Proceedings of the international conference GLOBAL, 2003, 2182–2187.
  - [15] Lapenta G. Nuclear Reactor Physics, CLUT, Turin, 2005, pp.195–210.
  - [16] Memoli V, Cammi A, Di Marcello V. Proceedings of the COMSOL Conference, Milan, Italy, 2009.
  - [17] Křepel J, Grundmann U, Rohde U. Proceedings of ICONE 12 Conference, Arlington, Virginia (Washington, DC), USA, 2004.
  - [18] Cammi A, Di Marcello V, Luzzi L, *et al.* Ann Nucl Energy, 2011, **38**: 1356–1372.
  - [19] Hill M D and Marty M R. Computer, 2008, **41**: 33–38.
  - [20] Gustafson J. Communications of the ACM, 1988, **31**: 532–533.
  - [21] Owens J D, Luebke D, Govindaraju N, *et al.* Comput Graph Forum, 2007, **26**: 80–113.
  - [22] Ruan J and Han D D. Microelectronics & Computer, 2009, **26**: 201–205.(in Chinese)