DEVELOPMENT OF A CORE THERMO-FLUID ANALYSIS CODE FOR PRISMATIC GAS COOLED REACTORS

NAM-IL TAK*, SUNG NAM LEE, MIN-HWAN KIM, HONG SIK LIM, and JAE MAN NOH
Korea Atomic Energy Research Institute
Daedeok-Daero 989-111, Yuseong-gu, Daejeon 305-353, Korea
*Corresponding author. E-mail : takni@kaeri.re.kr

Received February 25, 2014
Accepted for Publication June 16, 2014

A new computer code, named CORONA (Core Reliable Optimization and thermo-fluid Network Analysis), was developed for the core thermo-fluid analysis of a prismatic gas cooled reactor. The CORONA code is targeted for whole-core thermo-fluid analysis of a prismatic gas cooled reactor, with fast computation and reasonable accuracy. In order to achieve this target, the development of CORONA focused on (1) an efficient numerical method, (2) efficient grid generation, and (3) parallel computation. The key idea for the efficient numerical method of CORONA is to solve a three-dimensional solid heat conduction equation combined with one-dimensional fluid flow network equations. The typical difficulties in generating computational grids for a whole core analysis were overcome by using a basic unit cell concept. A fast calculation was finally achieved by a block-wise parallel computation method. The objective of the present paper is to summarize the motivation and strategy, numerical approaches, verification and validation, parallel computation, and perspective of the CORONA code.

KEYWORDS : Prismatic Core, Fuel Temperature, Fluid Flow Network, Gas Cooled Reactor, CORONA, VHTR

1. INTRODUCTION

A prismatic gas cooled reactor is a candidate design for national research programs such as the next generation nuclear plant (NGNP) project of the U. S. [1], and the nuclear hydrogen production and demonstration (NHDD) project of Korea [2]. For the smooth success of the NHDD project, the Korea Atomic Energy Research Institute (KAERI) has been developing key technologies for the design of a prismatic very high temperature reactor (VHTR) [3]. As one of the crucial technologies for the NHDD project, the development of a core thermo-fluid analysis code, named CORONA (Core Reliable Optimization and thermo-fluid Network Analysis), started in 2009. The CORONA code is targeted for a whole core thermo-fluid analysis of a prismatic gas cooled reactor with fast computation and reasonable accuracy.

Figs. 1 and 2 show the conceptual view of a typical prismatic reactor core and its fuel block respectively [4]. The active part of the prismatic core consists of a large number of vertical stacks of a fuel block. Accurate prediction of the fuel temperature is crucial to ensure the barrier against the release of fission products into the primary coolant. However, the complex geometry of a prismatic fuel block hinges accurate evaluations of the temperature distribution without elaborate numerical calculations.

Therefore, during the development of existing prismatic gas-cooled reactor designs (e.g., MHTGR of the General Atomics [5] and HTTR of Japan [6]), efficient numerical methods were applied to analyse the thermo-fluid behaviour of prismatic cores. The key idea of these efficient methods is to solve a three-dimensional solid heat conduction equation combined with one-dimensional fluid flow equations. A combined method was adopted in the DEMISE code of General Atomics [7] and the FLOWNET/TRUMP codes of Japan [8]. Recently the existing idea was improved by the present authors [9] and implemented into the CORONA code. Moreover, the fluid flow model described in [9] has been further improved through a one-dimensional network model.

The present paper summarizes the motivation and strategy, major outcomes, and perspective of the development of the CORONA code.

2. DEVELOPMENT MOTIVATION AND STRATEGY

The recent rapid development of computational fluid dynamics (CFD) technology provides a powerful tool to carry out elaborate numerical calculations with complex geometries for nuclear applications. Some CFD codes such as CFX [10] and STAR-CD [11] are commercially available. Thanks to the commercial CFD codes, the ap-
Application areas of CFD technology have been growing. Commercial CFD codes are user-friendly and well validated against most industrial problems which have a single phase fluid. In spite of the remarkable development of CFD technology, however, a whole prismatic core CFD analysis with geometric details is still a challenging task. Since fine meshes are required for accurate CFD calculations, most CFD applications are limited to local behaviours (e.g., single fuel column [12,13], and seven fuel columns [14]) in the design of a prismatic reactor. In 2009, Pointer and Thomas [15] published the STAR-CD results for a whole core of a prismatic reactor. It is clear that such a CFD analysis requires tremendous computational efforts. Additional significant efforts have to be devoted to generate computational grids of high quality for a good convergence. Such efforts are valuable for a final calculation but are too taxing for a designer who wants a large number of calculations with various design options.

The other available option at KAERI could be the use of system codes such as MARS-GCR [16] and GAMMA+ [17]. It should be noted, however, that the system codes are targeted for system transients. Their meshes are too coarse for a detailed thermo-fluid analysis of a prismatic fuel block. The accuracy of the system code calculations is not satisfactory for the core thermal design of a prismatic reactor (e.g., hot spot analysis).

Therefore, a new computer code named CORONA has been developed to overcome the difficulties in the available tools at KAERI, i.e., CFD codes and system codes. The CORONA code is intended for whole-core thermo-fluid analysis of a prismatic gas cooled reactor with fast computation and reasonable accuracy. Fig. 3 summarizes the thermo-fluid analysis codes and their application areas for the design of prismatic gas-cooled reactors at KAERI. The position of CORONA lies in between CFD and system codes. Its position is similar to that of a sub-channel analysis code such as COBRA [18] used in a light water cooled reactor design.
Major applications of the CORONA code are (1) a steady-state hot spot analysis, (2) steady-state analysis for off-design conditions such as a coolant channel blockage accident, (3) some transient scenarios (e.g., rod ejection), (4) a coupled analysis (e.g., with neutronics) for high fidelity, (5) design optimization calculations. The CORONA code is written in C++ and can be run on either Windows or Linux operating system.

3. NUMERICAL APPROACHES

3.1 Grid Generation Using Basic Unit Cells

The first step in the numerical calculation of thermo-fluid phenomena is to set up a grid. The unique feature of the CORONA code is the use of the basic unit cells for the generation of computational grids. The concept of basic unit cells adopted in the CORONA code enables the effective generation of unstructured grids for prismatic fuel blocks without a special mesh generator. The revision of the generated grids is also simple and convenient.

Efficient Numerical Method:

The use of an efficient numerical method is the most important requirement for a new code to achieve its goal. The CORONA code adopts the existing idea of the efficient numerical method used for the design of MHTGR and HTTR (e.g., DEMISE and FLOWNET/TRAMP). This efficient method solves a three-dimensional heat conduction equation for a solid like a CFD code and one-dimensional conservation equations for a fluid flow like a system code. In 2012, Travis and El-Genk [19] verified the efficiency of this combined method using STAR-CD calculations.

Efficient Grid Generation:

For a whole prismatic core thermo-fluid simulation, grid generation can also be a time-consuming process. Therefore, an efficient grid generation method using the concept of basic unit cells was developed by the present authors [9,20]. This concept of the basic unit cells adopted in the CORONA code enables the effective generation of unstructured grids for prismatic fuel blocks without a special mesh generator. The revision of the generated grids is also simple and convenient.

Parallel Computation:

Reliance on a parallel computing technique is inevitable for a whole prismatic core simulation since significant numbers of meshes (e.g., in the order of ten million) are required. Parallel computation leads not only to a speed up of the calculations but also allows the use of a normal personal computer (PC).
3.2 Numerical Model for Solid

In order to simulate the heat conduction through the CORONA code has a similar level as the CFD analysis. However, it should be noted that the computational expense of a solid heat conduction equation (i.e., Eq. (1)) is much less than that of fluid conservation equations in a whole core CFD analysis of a prismatic core.

**Fig. 4.** Typical Unit Cell Arrangements in Prismatic Fuel Blocks

**Fig. 5.** Basic Unit Cell Types and Examples of Computational Grids

**Fig. 6.** Basic Unit Cell Assignment to Generate Computational Grids of Standard Fuel Block

**Fig. 7.** Example of Computational Grids Generated Using the Basic Unit Cells for the Standard Fuel Block

Then, using an axial combination of the basic unit cells, a hexagonal (or pentagonal) body called a ‘pin’ is made. Three-dimensional computational grids for a hexagonal fuel (or reflector) block are generated using a two-dimensional combination of pins. Modeling of a whole core is finally finished using a combination of hexagonal fuel and reflector blocks.

The number of computational grids for a solid used in
orthogonal. At a fluid boundary, the following equation is governed:

\[
a_i(T_{s,i} - T_{s,j}) - s_{i,non} = q_{i,conv}^{conv}
\]  

Equations (3) and (6) can be rearranged into matrix form as

\[
a_i^* T_{s,i,p} = \sum_j a_i^j T_{s,j} + b_p
\]  

Equation (7) can be solved simultaneously for all nodes if the material properties and \( q_{i,conv}^{conv} \) are known. The detailed derivation from Eq. (1) to Eq. (7) is not given in this paper since it is lengthy and widely available in a CFD textbook dealing with unstructured meshes (e.g., see Versteeg and Malalasekera [21]).

3.3 Numerical Model for Fluid Flow

In contrast to the solid geometry of the fuel blocks, the fluid in the prismatic fuel blocks flows through simple circular channels. A fraction of the core flow bypasses the fuel block coolant channels and passes through the gaps between the hexagonal columns. Such gaps can also be considered as rectangular channels connected to each other. For the fluid flow within a prismatic core, a one-dimensional approach can therefore be a good approximation. Such an approximation avoids fine meshes near the walls as well as turbulence conservation equations which are one of the major challenges in a CFD analysis. A one-dimensional approximation for the duct flows enables a fast calculation with reasonable accuracy. In addition, there is potential for an exchange of coolant between the bypass gaps and the coolant channels due to cross flow gaps which are small spaces between the horizontal faces of the blocks. Therefore, a one-dimensional network model has been widely used for a fluid flow in a prismatic core [7].
The first step to solve Eqs. (11) and (12) is to guess the pressures at all nodes. These values are treated as preliminary values and denoted by $\bar{P}$. The correction $\delta P$ is defined as the difference between the correct pressure field and the guessed pressure field $\bar{P}$.

$$ P = \bar{P} + \delta P $$ (13)

Similarly, the volumetric flow rate correction $\delta Q$ and density correction $\delta \rho$ are defined as:

$$ Q = \bar{Q} + \delta Q $$ (14)

$$ \rho_f = \bar{\rho}_f + \delta \rho_f $$ (15)

The density of the gas can be obtained using the ideal gas law.

$$ \rho_f = \frac{P}{z_c RT} $$ (16)

Where $R$ is the gas constant and $z_c$ is the compressibility factor. Substitution of Eqs. (13)~(16) into the momentum equation (i.e., Eq. (12)) and neglecting the terms involving the products of the corrections, yield the following equation for the volumetric flow rate correction.

$$ \delta Q_j = \frac{1}{2C_j \bar{\rho}_{f,j} |\bar{Q}_j|} \times \left\{ \left[1 - G_j \delta P_{i,uj} - (1 + G_j) \delta P_{i,dj} \right] + \bar{P}_{i,uj} - \bar{P}_{i,dj} - C_j \bar{P}_{f,j} \bar{Q}_{j} |\bar{Q}_j| \right\} $$ (17)

where

$$ C_j = \frac{1}{2} \left( \frac{f_j L_j}{D_j} + K_j \right) \frac{1}{A_{f,j}^2} $$ (18)

$$ G_j = \frac{C_j \bar{Q}_{j} |\bar{Q}_j|}{2z_c RT_j} $$ (19)

Finally, the substitution of Eqs. (14), (15) and (17) into the continuity equation (i.e., Eq. (11)) yields the following equation for the pressure correction.

$$ \sum_{j \in T_i} a_j + \sum_{j \in T_i} b_j \delta P_j + \sum_{j \in T_i} c_j \delta P_{i,uj} + \sum_{j \in T_i} d_j \delta P_{i,dj} = e_i $$ (20)
Where \( a_j, d_j, c_j, \) and \( d_j \) are the rearranged coefficients in terms of each \( \delta P \) node respectively and \( e_j \) is the source term.

With the known field of \( \delta P, \delta Q \) can be obtained using Eq. (17). Then, new values for \( P, Q, \) and \( \rho_{ij} \) are determined using Eqs. (13), (14), and (16). The newly updated values are now considered as the preliminary values for the next iteration and the whole process is repeated a number of times after which the fluid energy conservation equation is solved.

The steady-state one-dimensional form for the energy conservation equation governing the fluid flow network can be derived as

\[
\sum_{j \in I_i} \rho_{f,i} Q_j a_j - \sum_{j \in I_i} \rho_{f,i} Q_j a_j h_i - \sum_{j \in I_i} \rho_{f,i} Q_j a_j h_{i,j} + \sum_{j \in I_i} \rho_{f,i} Q_j a_j h_{d,j} = q_{conv,f,i} \tag{21}
\]

where \( h_i \) is the enthalpy of the fluid node \( i \) and \( a_j \) is the parameter for the donor property defined as

\[
\begin{align*}
\alpha_j & = 1 \text{ if } (Q_j \geq 0, j \in T_i) \text{ or } (Q_j < 0, j \in T_i) \\
& = 0 \text{ if } (Q_j < 0, j \in T_i) \text{ or } (Q_j \geq 0, j \in T_i) \tag{22}
\end{align*}
\]

The central differencing scheme can be used as an alternative option for Eq. (21). In the case of a single pipeline, Eqs. (20) and (21) can be solved using the tri-diagonal matrix algorithm (TDMA), while sparse matrix techniques can be applied in the case of complex flow networks.

### 3.4 Interfacing of Solid and Fluid Models

The coupling between the three-dimensional solid model and the one-dimensional fluid model uses a typical Nusselt number correlation such as:

\[
Nu = 0.021 Re^{0.8} Pr^{0.4} \left( \frac{T_s}{T_f} \right)^{-0.5} \tag{23}
\]

\[
Nu = 4.364 \quad \text{(for laminar flow)} \tag{24}
\]

Eq. (23) is the correlation of McEligot et al. [7, 25] which is one of the most popular correlations for a prismatic gas-cooled reactor core under normal operating conditions. The entrance effect is neglected due to little interest in the entrance region. Eqs. (6) and (21) are combined as follows:

\[
q_{conv,f,j} = Nu_j k_{f,j} A_{s,j} (T_{s,j} - T_{f,j}) \tag{25}
\]

\[
q_{conv,f,j} = \sum_i q_{conv,s,i,j} \tag{26}
\]

where \( i \) represents each surface of the solid contacted with the fluid node \( j \) and \( A_{s,j} \) is the heat transfer area between the solid surface \( i \) and fluid node \( j \).

The overall computational procedure for a steady-state analysis is shown in Fig. 10. When the maximum value of the fluid temperature change during the iteration is less than the user supplied parameter, the convergence is regarded as having been achieved.

### 4. VERIFICATION AND VALIDATION

#### 4.1 Heat Conduction in Unit Cells

In order to verify the multi-dimensional heat conduction within unit cells of the prismatic fuel block, two conceptual problems using unit cells were investigated [20]. The verification study started from a single unit cell problem which is one of the simplest test cases. The single unit cell consists of a fuel compact, helium gap, and graphite region. A uniform heat generation in the fuel compact was assumed...
4.2 Bypass Flow and Crossflow in Fuel Column

An air flow experiment performed by Kaburaki and Takizuka [26] was selected to validate the one-dimensional fluid flow network model implemented in CORONA [27]. Fig. 14 shows a schematic of their experimental apparatus. Four graphite blocks were stacked up in a column and surrounded by a steel shroud to simulate the bypass gap (~1.2mm) between fuel columns. The hexagonal fuel block has 12 coolant holes 20 mm in diameter. One artificial cross flow gap (~1 mm) was created in the middle of the column to simulate a crossflow gap between fuel blocks. The air at atmospheric pressure was drawn from the top of the test section and flowed down the channels and the bypass gap. The bottom of the bypass gap was completely sealed and the coolant leakage flow, which entered the bypass gap, was merged into the main coolant channels through the crossflow gap. The mass flow rate

and the temperature of the outmost boundary was fixed. Since it is believed that a commercial CFD code can solve a multi-dimensional heat conduction problem accurately, a commercial CFD code, CFX, is used for the verification of the CORONA code for the multi-dimensional heat conduction problem. Fig. 11 compares the predicted temperature profile by CORONA with that by CFX. It shows that CORONA accurately solves the multi-dimensional heat conduction equation. It is obvious that a finer mesh provides a more accurate result.

The second case considers seven unit cells composed of one coolant cell (designated as ‘C’) surrounded by six fuel cells (designated as ‘F’) as shown in Fig. 12. A uniform heat generation in the fuel compacts was assumed and the temperature at the coolant boundary was fixed. In addition, the adiabatic boundary condition was imposed on the outmost boundary of the unit cells.

Fig. 13 shows the predicted temperature profiles along Line A in Fig. 12. It was observed that when a fine mesh is adopted, the accuracy of CORONA is close to that of CFX.
and the static pressure distribution in each coolant channel were measured. The measurement uncertainties of the experiment were not published and the sensitivity calculations of CORONA confirmed later that the effect of the entrance and exit form losses was small.

Fig. 15 shows a comparison of the CORONA prediction results with the measured pressures. A good agreement can be seen along all of the stacks. This was achieved by applying the correlation of Kaburaki and Takizuka developed based on their experiment. This means that the described one-dimensional fluid flow model is correctly implemented into the CORONA code and is able to simulate the bypass flow and crossflow phenomena reasonably.

4.3 Conjugate Heat Transfer in Fuel Column

In order to verify a conjugate heat transfer in a prismatic fuel block, two conceptual problems with single fuel columns were investigated [28, 29]. Two types of fuel blocks (i.e., standard and reserved shutdown control (RSC) fuel blocks) were considered. Since it is believed that a commercial CFD code is well verified against a single phase conjugate heat transfer problem, the verification of the CORONA code is made using a comparison with the results of CFX. However, it should be noted that a commercial CFD code also needs a sufficient validation for prismatic core conditions (i.e., high temperature helium flow at a high heat flux condition in a prismatic fuel block geometry).

Table 1 shows the major thermo-fluid parameters for the considered single fuel column problems under consideration. Fig. 16 shows the CFX meshes used to simulate the single fuel column problems. Using symmetry, 1/12 and 1/2 sections of the geometries were modeled for CFX calculations of the standard and RSC fuel blocks, respectively.
fair comparison of the computation time is not feasible due to differing convergence criteria and a differing number of meshes. However, it is obvious that significant reduction in the computation time can be achieved by CORONA and the difference in the computation time would dramatically soar in the case of a whole core analysis.

5. PARALLEL COMPUTATION

5.1 Block-Based Parallel Computation

Parallel computing capability is important for the CORONA code since a whole prismatic core simulation requires a huge computing cost and a significant amount of computer memory. For a simple implementation of parallel computing capability, a block-based parallel computation method was developed for the CORONA code [30]. The idea of the block-based parallel computation method is based on the fact that each fuel (or reflector) block is completely enclosed by the fluid boundaries due to the existence of bypass and cross flows as shown in Fig. 20. Therefore, the heat conduction matrix shown in Eq. (7) can be naturally decomposed into much smaller matrixes that represent each fuel (or reflector) block. The decomposed matrix can be solved independently using a parallel computation library.

5.2 Application to Whole Core Analysis

In order to examine the improved performance by the parallel computation, an example problem for a whole prismatic core was investigated. PMR600 [2] was selected as a reference core. The major thermo-fluid boundary conditions are shown in Table 2. The reactor power profile obtained by the neutronic analysis [31] was applied. The applied power distribution is shown in Fig. 21. The heat loss to the reactor cavity cooling system (RCCS) was neglected.

Fig. 22 shows the calculation model for the CORONA
reduced with the number of processors. The maximum number of processors per PC was limited to 4 due to the random access memory (RAM) capacity of the PCs.

Table 2. Major Thermo-fluid Parameters for PMR600 Benchmark Calculation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core thermal power (MW)</td>
<td>600</td>
</tr>
<tr>
<td>No. of fuel columns</td>
<td>108</td>
</tr>
<tr>
<td>No. of fuel blocks per column</td>
<td>9</td>
</tr>
<tr>
<td>No. of reflector columns with control rod hole</td>
<td>36</td>
</tr>
<tr>
<td>Core flow rate (kg/s)</td>
<td>251.3</td>
</tr>
<tr>
<td>Coolant inlet temperature (°C)</td>
<td>490</td>
</tr>
<tr>
<td>System pressure (MPa)</td>
<td>7</td>
</tr>
<tr>
<td>Active core height (m)</td>
<td>7.929</td>
</tr>
<tr>
<td>Bypass gap size (mm)</td>
<td>2</td>
</tr>
<tr>
<td>Crossflow gap size (mm)</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3. Specification of Two PCs for Parallel Computation Benchmark

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>No. of CPUs</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC 1</td>
<td>Intel Xeon X5690, Six Core, 3.47 GHz</td>
<td>2</td>
<td>144 GB</td>
</tr>
<tr>
<td>PC 2</td>
<td>Intel Xeon X5690, Six Core, 3.47 GHz</td>
<td>2</td>
<td>96 GB</td>
</tr>
</tbody>
</table>
blocks. The usefulness of the CORONA code was proved by the example calculations using a whole prismatic core model.

Further research is on-going to improve the physical models and increase the capability of the CORONA code. The major items are (1) coupling with the DeCART code [33] for a high fidelity calculation, (2) variable bypass (and crossflow) gap size model for consideration of the change in the size with the location as well as the burnup, and (3) transient simulation capability.

It is believed that the CORONA code can be practically used in a whole core thermo-fluid analysis for the design of a prismatic gas cooled reactor. In addition, it is envisaged that the excellence of the CORONA code will also be highlighted in a high fidelity analysis combined with a whole core neutron transport calculation.

### NOMENCLATURE

\( a_i \) = Influence coefficient at node \( i \), defined in Eq. (5)

\( a_i^* \) = Rearranged influence coefficient of \( a_i \), used in Eq. (7)

\( a_j \) = Coefficient for junction \( j \), used in Eq. (20)

\( a_0^P \) = Influence coefficient at node \( P \), defined in Eq. (5)

\( a_0^P \) = Rearranged influence coefficient of \( a_0^P \), used in Eq. (7)

\( A_s \) = Solid surface area \([\text{m}^2]\)

\( b_j \) = Coefficient for junction \( j \), used in Eq. (20)

\( b_p \) = Source term for node \( P \) at temperature matrix, used in Eq. (7)

\( c_j \) = Coefficient for junction \( j \), used in Eq. (20)

\( C_j \) = Resistance coefficient at fluid node \( j \), defined in Eq. (18)

\( C_s \) = Specific heat of solid \([\text{J/kg K}]\)

\( d_j \) = Coefficient for junction \( j \), used in Eq. (20)

\( D \) = Hydraulic diameter \([\text{m}]\)

\( e_i \) = Source term for node \( i \) at pressure matrix, used in Eq. (20)

\( f \) = Friction factor \([-]\)

\( g \) = Gravitational acceleration constant \([\text{m/s}^2]\)

\( G \) = Parameter defined in Eq. (19)

\( h \) = Fluid enthalpy \([\text{J/kg}]\)

\( I_i \) = Initiating junction of node \( i \)

\( k_f \) = Thermal conductivity of fluid \([\text{W/m K}]\)

\( k_s \) = Thermal conductivity of solid \([\text{W/m K}]\)

\( K \) = Form loss factor \([-]\)

\( L \) = Fluid channel length \([\text{m}]\)

\( N \) = Number of meshes

\( N_u \) = Nusselt number \([-]\)

\( P \) = Pressure \([\text{Pa}]\)

\( P_r \) = Prandtl number \([-]\)

\( q_g^{\text{conv}} \) = Power density generated in solid \([\text{W/m}^3]\)

\( q_f^{\text{conv}} \) = Convection heat transfer at fluid \([\text{W}]\)

\( q_s^{\text{conv}} \) = Convection heat transfer at solid \([\text{W}]\)

\( Q \) = Volumetric flow rate \([\text{m}^3/s]\)

\( R \) = Gas constant of helium \([\text{J/kg K}]\)

\( Re \) = Reynolds number \([-]\)

\( s_{\text{non}} \) = Cross-diffusion term

\( t \) = Time \([\text{s}]\)

\( T_f \) = Fluid temperature \([\text{K}]\)

---

### 6. CONCLUSIONS AND OUTLOOK

The development of the CORONA code was started in KAERI for the establishment of the key technologies for the design of a prismatic VHTR. The CORONA code is targeted for a whole core thermo-fluid analysis of a prismatic gas cooled reactor with fast computation and reasonable accuracy. In order to achieve the target, the development of CORONA focused on (1) an efficient numerical method using the combination of a three-dimensional heat conduction and one-dimensional fluid flow network, (2) efficient grid generation using basic unit cells, and (3) parallel computing capability.

The results of the verification and validation studies show that the CORONA code can provide a reasonably accurate result for the thermo-fluid analysis of prismatic fuel blocks.
\[ T_i \] = Terminating junction of node \( i \)
\[ T_s \] = Solid temperature [K]
\[ V \] = Volume [m³]
\[ w \] = Fluid velocity [m/s]
\[ z \] = One-dimensional coordinate
\[ z_c \] = Compressibility factor [-]

**Greek**
\[ \alpha \] = Parameter defined in Eq. (22)
\[ \rho \] = Density [kg/m³]

**ACKNOWLEDGEMENTS**
This work was supported by Nuclear R&D Program of the National Research Foundation of Korea which is grant funded by the Korean government (Grant code: NRF-2012M2A8A2025679).

**REFERENCES**


