Study on Adjoint Based Node Sensitivity Analysis of Two Phase Thermal-Hydraulic System Analysis Code

Jae Jun Lee, Min-Gil Kim, Seongmin Son, Jeong Ik Lee*
Department of Nuclear and Quantum engineering, Korea Advanced Institute of Science and Technology (KAIST)
291 Daehak-ro, (373-1, Guseong-dong), Yuseong-gu, Daejeon 34141, Republic of KOREA
**Corresponding author: jeongiklee@kaist.ac.kr*

1. Introduction

In nuclear engineering, thermal-hydraulic (TH) system analysis code plays a crucial role in safety analysis of nuclear power plant (NPP). For the analysis, nodalization should be determined by users, and it is well known that this node configuration affects the simulation results, i.e. user effect. To obtain reliable results, user effect should be analyzed and this can be done through sensitivity analysis. Using the adjoint method instead of the forward method, it is possible to perform the sensitivity analysis efficiently for many parameters such as the geometric position of the nodes. In this paper, the authors perform an adjoint based node sensitivity analysis in two phase TH system analysis code (MARS-KS 1.4) and compare that the sensitivities derived from forward and above method are equivalent.

2. Methods

In this section, governing equations of MARS-KS 1.4 are presented in section 2.1. The adjoint based sensitivity analysis procedure with discretized governing equations is presented in section 2.2. In section 2.3, implementation process for the sensitivity calculation in the code is described.

2.1 Governing equations of MARS-KS 1.4

The MARS (Multi-dimensional Analysis of Reactor Safety) code is developed by KAERI for a multi-dimensional and multi-purpose realistic thermal-hydraulic system analysis of light water reactor transients [2]. It is basically consisted of hydrodynamic equations containing non-condensable gas and conduction equation for heat structure. Following expanded 6 hydrodynamic equations are used when there is only water in the simulated system. Independent variables (void fraction α_g , pressure P , internal energy U_f, U_g , velocity v_f, v_g) at new time step are calculated by solving these equations at every time step.

Expanded sum density equation.

$$\alpha_g \frac{\partial \rho_g}{\partial t} + \alpha_f \frac{\partial \rho_f}{\partial t} + (\rho_g - \rho_f)\frac{\partial \alpha_g}{\partial t} + \frac{1}{A}\frac{\partial}{\partial x}(\alpha_g \rho_g v_g A + \alpha_f \rho_f v_f A) = 0 \ldots (1)$$

Expanded difference density equation.

$$\alpha_g \frac{\partial \rho_g}{\partial t} - \alpha_f \frac{\partial \rho_f}{\partial t} + (\rho_g + \rho_f)\frac{\partial \alpha_g}{\partial t} + \frac{1}{A}\frac{\partial}{\partial x}(\alpha_g \rho_g v_g A - \alpha_f \rho_f v_f A) + \frac{2\left[\frac{P_s}{P} H_{is}(T^s - T_g) + H_{if}(T^s - T_f)\right]\frac{P_s}{P}}{h_g^* - h_f^*} - 2\Gamma_{\mathrm{w}} = 0 \ldots (2)$$

Expanded sum momentum equation.

$$\alpha_g \rho_g \frac{\partial v_g}{\partial t} + \alpha_f \rho_f \frac{\partial v_f}{\partial t} + \frac{1}{2}\alpha_g \rho_g \frac{\partial v_g^2}{\partial x} + \frac{1}{2}\alpha_f \rho_f \frac{\partial v_f^2}{\partial x} + \frac{\partial P}{\partial x} - \rho_m B_x + \rho_g \mathrm{FWF} v_g + \rho_f \mathrm{FWF} v_f + \Gamma_{\mathrm{g}}(v_g - v_f) = 0 \ldots (3)$$

Expanded difference momentum equation.

$$\frac{\partial v_g}{\partial t} - \frac{\partial v_f}{\partial t} + \frac{1}{2}\frac{\partial v_g^2}{\partial x} - \frac{1}{2}\frac{\partial v_f^2}{\partial x} + \left(\frac{1}{\rho_g} - \frac{1}{\rho_f}\right)\frac{\partial P}{\partial x} + \mathrm{FWG} v_g - \mathrm{FWF} v_f - \frac{\Gamma_{\mathrm{g}}\left(\rho_m v_I - (\alpha_g \rho_g v_f + \alpha_f \rho_f v_g)\right)}{\alpha_f \rho_f \alpha_g \rho_g} + \rho_m FI(v_g - v_f) + C\frac{\rho_m^2}{\rho_g \rho_f}\frac{\partial (v_g - v_f)}{\partial t} - \frac{\rho_m}{\rho_g \rho_f}(\rho_f - \rho_g) B_y \frac{\partial y}{\partial x} = 0 \ldots (4)$$

Expanded liquid energy equation.

$$-(\rho_f U_f + P)\frac{\partial \alpha_f}{\partial t} + \alpha_f U_f \frac{\partial \rho_f}{\partial t} + \alpha_f \rho_f \frac{\partial U_f}{\partial t} + \frac{1}{A}\left[\frac{\partial}{\partial x}(\alpha_f \rho_f U_f v_f A) + P\frac{\partial}{\partial x}(\alpha_f \rho_f A)\right] - \left(\frac{h_f^*}{h_g^* - h_f^*}\right)\frac{P_s}{P} H_{ig}(T^s - T_g) - \left(\frac{h_g^*}{h_g^* - h_f^*}\right) H_{if}(T^s - T_f) - \left(\frac{P - P_s}{P}\right) H_{gf}(T_g - T_f) + \left[\left(\frac{1+\varepsilon}{2}\right)h_g' + \left(\frac{1-\varepsilon}{2}\right)h_f'\right]\Gamma_{\mathrm{w}} - Q_{wf} - DISS_f = 0 \ldots (5)$$

Expanded vapor energy equation.

$$(\rho_g U_g + P)\frac{\partial \alpha_g}{\partial t} + \alpha_g U_g \frac{\partial \rho_g}{\partial t} + \alpha_g \rho_g \frac{\partial U_g}{\partial t} + \frac{1}{A}\left[\frac{\partial}{\partial x}(\alpha_g \rho_g U_g v_g A) + P\frac{\partial}{\partial x}(\alpha_g \rho_g A)\right] + \left(\frac{h_f^*}{h_g^* - h_f^*}\right)\frac{P_s}{P} H_{ig}(T^s - T_g) + \left(\frac{h_g^*}{h_g^* - h_f^*}\right) H_{if}(T^s - T_f) + \left(\frac{P - P_s}{P}\right) H_{gf}(T_g - T_f) - \left[\left(\frac{1+\varepsilon}{2}\right)h_g' + \left(\frac{1-\varepsilon}{2}\right)h_f'\right]\Gamma_{\mathrm{w}} - Q_{wg} - DISS_g = 0 \ldots (6)$$

These above equations are discretized with semi-implicit scheme and could be expressed in Eq.7 with independent variable vector set **X** and parameter **N**.

$$\boldsymbol{F}(\boldsymbol{X},\boldsymbol{N}) = \boldsymbol{A}(\boldsymbol{X}^{\boldsymbol{n}},\boldsymbol{N})\boldsymbol{X}^{\boldsymbol{n}+\boldsymbol{1}} + \boldsymbol{B}(\boldsymbol{X}^{\boldsymbol{n}},\boldsymbol{N}) = \boldsymbol{0} \; \dots (7)$$

where $\boldsymbol{X} = (\boldsymbol{\alpha}_{\boldsymbol{g}}, \boldsymbol{P}, \boldsymbol{U}_{\boldsymbol{f}}, \boldsymbol{v}_{\boldsymbol{f}}, \boldsymbol{U}_{\boldsymbol{g}}, \boldsymbol{v}_{\boldsymbol{g}})$

2.2 Adjoint based sensitivity analysis procedure

Adjoint method is a widely used technique for sensitivity computation of parameter. It retrieves derivatives of a cost function respect to parameters efficiently. When objective function as $\boldsymbol{G}(\boldsymbol{X},\boldsymbol{N})$, parameters i.e. node position as $\boldsymbol{N}$. The sensitivity is defined as Eq.8

$$\frac{d\boldsymbol{G}(\boldsymbol{X},\boldsymbol{N})}{d\boldsymbol{N}} = \frac{\partial \boldsymbol{G}(\boldsymbol{X},\boldsymbol{N})}{\partial \boldsymbol{N}} + \frac{\partial \boldsymbol{G}(\boldsymbol{X},\boldsymbol{N})}{\partial \boldsymbol{X}}\frac{d\boldsymbol{X}}{d\boldsymbol{N}} \; \dots (8)$$

In Eq.9, differentiating the discretized governing equations with respect to the parameters, an approach called discrete adjoint approach, is used.

$$\frac{d\boldsymbol{F}(\boldsymbol{X}^{\boldsymbol{n}+\boldsymbol{1}},\boldsymbol{X}^{\boldsymbol{n}},\boldsymbol{N})}{\partial \boldsymbol{X}^{\boldsymbol{n}+\boldsymbol{1}}}\boldsymbol{\phi}^{\boldsymbol{n}+\boldsymbol{1}} + \frac{d\boldsymbol{F}(\boldsymbol{X}^{\boldsymbol{n}+\boldsymbol{1}},\boldsymbol{X}^{\boldsymbol{n}},\boldsymbol{N})}{\partial \boldsymbol{X}^{\boldsymbol{n}}}\boldsymbol{\phi}^{\boldsymbol{n}} = -\frac{d\boldsymbol{F}(\boldsymbol{X}^{\boldsymbol{n}+\boldsymbol{1}},\boldsymbol{X}^{\boldsymbol{n}},\boldsymbol{N})}{\partial \boldsymbol{N}} \; \dots (9)$$

where $\frac{d\boldsymbol{X}^{\boldsymbol{n}+\boldsymbol{1}}}{d\boldsymbol{N}} = \boldsymbol{\phi}^{\boldsymbol{n}+\boldsymbol{1}}, \frac{d\boldsymbol{X}^{\boldsymbol{n}}}{d\boldsymbol{N}} = \boldsymbol{\phi}^{\boldsymbol{n}}$

Eq.9 can be expressed with identity matrix **I** in Eq.10.

$$\begin{bmatrix} \frac{dF(X^{n+1},X^n,N)}{\partial X^{n+1}} & \frac{dF(X^{n+1},X^n,N)}{\partial X^n} & 0 & \cdots & 0 \\ 0 & \frac{dF(X^n,X^{n-1},N)}{\partial X^n} & \frac{dF(X^n,X^{n-1},N)}{\partial X^{n-1}} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{dF(X^1,X^0,N)}{\partial X^1} & \frac{dF(X^1,X^0,N)}{\partial X^0} \\ 0 & 0 & \cdots & 0 & I \end{bmatrix} \begin{bmatrix} \phi^{n+1} \\ \phi^n \\ \vdots \\ \phi^1 \\ \phi^0 \end{bmatrix} = \begin{bmatrix} -\frac{dF(X^{n+1},X^n,N)}{\partial N} \\ -\frac{dF(X^n,X^{n-1},N)}{\partial N}) \\ \vdots \\ -\frac{dF(X^1,X^0,N)}{\partial N} \\ \phi^0 \end{bmatrix} \dots (10)$$

Eq.12 shows that the node sensitivity can be obtained with adjoint function $\boldsymbol{\lambda}$ that satisfying Eq.11

$$\begin{bmatrix} \frac{dF(X^{n+1},X^n,N)}{\partial X^{n+1}} & \frac{dF(X^{n+1},X^n,N)}{\partial X^n} & 0 & \cdots & 0 \\ 0 & \frac{dF(X^n,X^{n-1},N)}{\partial X^n} & \frac{dF(X^n,X^{n-1},N)}{\partial X^{n-1}} & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \frac{dF(X^1,X^0,N)}{\partial X^1} & \frac{dF(X^1,X^0,N)}{\partial X^0} \\ 0 & 0 & \cdots & 0 & I \end{bmatrix}^T \begin{bmatrix} \lambda_{n+1} \\ \lambda_n \\ \vdots \\ \lambda_1 \\ \lambda_0 \end{bmatrix} = \begin{bmatrix} \frac{\partial G(X,N)}{\partial X^{n+1}} \\ \frac{\partial G(X,N)}{\partial X^n} \\ \vdots \\ \vdots \\ \frac{\partial G(X,N)}{\partial X^0} \end{bmatrix} \dots (11)$$

$$\frac{d\boldsymbol{G}(\boldsymbol{X},\boldsymbol{N})}{d\boldsymbol{N}} = \frac{\partial \boldsymbol{G}(\boldsymbol{X},\boldsymbol{N})}{\partial \boldsymbol{N}} + [\lambda_{n+1} \quad \lambda_n \quad \cdots \quad \cdots \quad \lambda_0] \begin{bmatrix} -\frac{d\boldsymbol{F}(\boldsymbol{X}^{\boldsymbol{n}+\boldsymbol{1}},\boldsymbol{X}^{\boldsymbol{n}},\boldsymbol{N})}{\partial N} \\ -\frac{d\boldsymbol{F}(\boldsymbol{X}^{\boldsymbol{n}},\boldsymbol{X}^{\boldsymbol{n}-\boldsymbol{1}},\boldsymbol{N})}{\partial N}) \\ \vdots \\ -\frac{d\boldsymbol{F}(\boldsymbol{X}^{\boldsymbol{1}},\boldsymbol{X}^{\boldsymbol{0}},\boldsymbol{N})}{\partial N} \\ \boldsymbol{\phi}^{\boldsymbol{0}} \end{bmatrix} \dots (12)$$

2.3 Implementation process on the code

For node sensitivity analysis, necessary data are extracted and saved for adjoint sensitivity module from MARS-KS 1.4 and sensitivity is calculated in adjoint sensitivity module code written in MATLAB.

Figure.1 is the algorithm for adjoint based sensitivity calculation.

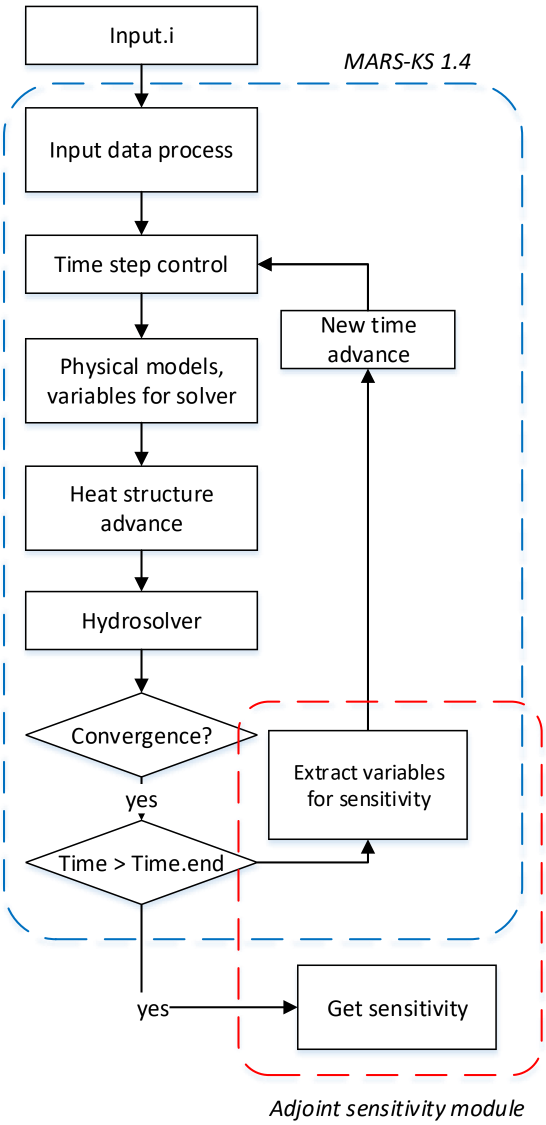

Fig. 1. Algorithm of adjoint based sensitivity calculation

3. Results

In this section, the calculated node sensitivities with each forward method and adjoint method are presented. We tested two similar problems under single phase flow condition first. The flow is in the pipe and it is driven by pressure difference. In case 1, water flows through the pipe. In case 2, only vapor is flowing.

3.1 Description of cases.

A nodalization of problem is shown in Figure. 2. Horizontal flow occurs due to the pressure difference between each side. Left time dependent volume (TMDPVOL) has higher pressure than right time dependent volume. Table. 1 shows problem conditions.

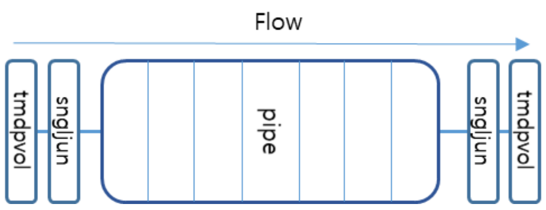

Fig. 2 Nodalization of Tests

Table I. Conditions of Tests

Geometry		Value
Pipe area	[m^2]	0.1
Pipe length	[m]	11
Roughness	[m]	0
Initial Conditions		
Liquid velocity in case 1	[m/s]	1.0
Vapor velocity in case 2	[m/s]	1.0
Pipe pressure	[MPa]	3.6
System temperature in case 1	[K]	323
System temperature in case 2	[K]	623
Boundary Conditions		
Left tmdpvol pressure	[MPa]	4.2
Right tmdpvol pressure	[MPa]	3.6
Tmdpvol temperature in case 1	[K]	323
Tmdpvol temperature in case 2	[K]	623
Problem Time Conditions		
Problem time	[sec]	30
Minimum time step	[sec]	1e-6
Maximum time step	[sec]	1e-4

3.2 Calculated sensitivity results

Response function of the test is the converged mass flow rate at last single junction. When pipe consists of uniformly distributed 5 nodes, liquid mass flow rate is 7696 kg/s in case 1 and vapor mass flow rate is 570 kg/s in case 2. When there are 10 nodes, the values of case 1 and case 2 are 7695 kg/s and 565 kg/s.

Table.2 shows that sensitivity obtained using forward method is converged as the Δx decreases but it diverges if the Δx is too small due to truncation error. The values converged sufficiently are marked with a bold and skewed font.

Table. 3 shows the sensitivity obtained by the adjoint method.

Table II. Node sensitivity with forward method

5 Nodes	**Δx= -0.1**	**Δx= +0.1**	**Δx= -0.01**
# 1	-0.0106	-0.0072	-0.0091
# 2	-0.0017	0.0016	-0.00020
# 3	-0.0025	0.0008	-0.0010
# 4	-0.4655	-0.4621	-0.4640
5 Nodes	**Δx= +0.01**	**Δx= -0.001**	**Δx= +0.001**
# 1	-0.0088	-0.0089	-0.0089
# 2	0.00013	-0.00005	-0.00002
# 3	-0.0007	-0.00087	-0.00084
# 4	-0.4636	-0.4638	-0.4638
5 Nodes	**Δx= -0.0001**	**Δx=+0.0001**	Case 1
# 1	***-0.0089***	***-0.0089***	
# 2	***-0.00004***	***-0.00004***	
# 3	***-0.00086***	***-0.00086***	
# 4	***-0.4638***	***-0.4638***	
5 Nodes	**Δx= -0.1**	**Δx= +0.1**	**Δx= -0.01**
# 1	-0.0901	-0.0879	-0.0891
# 2	-0.1474	-0.1453	-0.1464
# 3	-0.4135	-0.4115	-0.4126
# 4	-1.0633	-1.0612	-1.0624
5 Nodes	**Δx= +0.01**	**Δx= -0.001**	**Δx= +0.001**
# 1	-0.0889	-0.0890	-0.0890
# 2	-0.1462	-0.1463	-0.1463
# 3	-0.4124	-0.4125	-0.4125
# 4	-1.0621	-1.0623	-1.0622
5 Nodes	**Δx= -0.0001**	**Δx=+0.0001**	Case 2
# 1	***-0.0890***	***-0.0890***	
# 2	***-0.1463***	***-0.1463***	
# 3	***-0.4125***	***-0.4125***	
# 4	***-1.0622***	***-1.0622***	
10 Nodes	**Δx= -0.05**	**Δx= 0.05**	**Δx= -0.005**
# 1	-0.0053	-0.0036	-0.0045
# 2	-0.00085	0.00082	-0.0001
# 3	-0.00084	0.00083	-0.000085
# 4	-0.00084	0.00083	-0.000085
# 5	-0.00084	0.00083	-0.000085
# 6	-0.00084	0.00083	-0.000087
# 7	-0.00084	0.00083	-0.000087
# 8	-0.0017	-0.000041	-0.00096
# 9	-0.4689	-0.4672	-0.4682
10 Nodes	**Δx= 0.005**	**Δx= -0.0005**	**Δx= 0.0005**
# 1	-0.0044	-0.0045	-0.0044
# 2	0.000067	-0.000025	-0.000009
# 3	0.000082	-0.000009	0.000006
# 4	0.000082	-0.00001	0.0000066
# 5	0.000082	-0.000009	0.000007
# 6	0.000080	-0.000010	0.0000060

# 7	0.000080	-0.000011	0.0000046
# 8	-0.00079	-0.000890	-0.000861
# 9	-0.4680	-0.4681	-0.4681
10 Nodes	**Δx= -0.00005**	**Δx= 0.00005**	Case 1
# 1	***-0.0045***	***-0.0045***	
# 2	-0.000017	-0.000015	
# 3	-0.0000007	-0.0000016	
# 4	0.0000004	-0.0000014	
# 5	0.0000002	-0.0000038	
# 6	0.0000018	-0.0000004	
# 7	0.0000058	-0.0000022	
# 8	***-0.000875***	***-0.000882***	
# 9	***-0.4681***	***-0.4681***	
10 Nodes	**Δx= -0.05**	**Δx= 0.05**	**Δx= -0.005**
# 1	-0.0256	-0.0244	-0.0251
# 2	-0.0112	-0.0100	-0.0107
# 3	-0.0066	-0.0054	-0.0061
# 4	-0.0086	-0.0074	-0.0080
# 5	-0.0194	-0.0182	-0.0189
# 6	-0.0535	-0.0523	-0.0530
# 7	-0.1512	-0.1500	-0.1507
# 8	-0.4323	-0.4313	-0.4318
# 9	-1.0694	-1.0683	-1.0689
10 Nodes	**Δx= 0.005**	**Δx= -0.0005**	**Δx= 0.0005**
# 1	-0.0250	-0.0250	-0.0250
# 2	-0.0106	-0.0106	-0.0106
# 3	-0.0060	-0.0060	-0.0060
# 4	-0.0079	-0.0080	-0.0080
# 5	-0.0187	-0.0188	-0.0188
# 6	-0.0528	-0.0529	-0.0529
# 7	-0.1506	-0.1506	-0.1506
# 8	-0.4317	-0.4318	-0.4318
# 9	-1.0688	-1.0689	-1.0689
10 Nodes	**Δx= -0.00005**	**Δx= 0.00005**	Case 2
# 1	***-0.0250***	***-0.0250***	
# 2	***-0.0106***	***-0.0106***	
# 3	***-0.0060***	***-0.0060***	
# 4	***-0.0080***	***-0.0080***	
# 5	***-0.0188***	***-0.0188***	
# 6	***-0.0529***	***-0.0529***	
# 7	***-0.1506***	***-0.1506***	
# 8	***-0.4318***	***-0.4318***	
# 9	***-1.0689***	***-1.0689***	

4. Summary and Further Work

In this study, authors calculated sensitivities using the code written by authors and observe that sensitivities between adjoint and forward methods have similar directions in the single phase problems with MARS-KS 1.4. Update of the code is necessary for more accurate calculations. The method will be applied to the two phase mixture problem and will be presented in the conference.

Table III. Node sensitivity with adjoint method

5 Nodes	**Sensitivity**	**10 Nodes**	**Sensitivity**
# 1	-0.0297	# 1	-0.0143
# 2	-0.0214	# 2	-0.0101
# 3	-0.0223	# 3	-0.0099
# 4	-0.9353	# 4	-0.0099
Case 1		# 5	-0.0099
		# 6	-0.0100
		# 7	-0.0100
		# 8	-0.0115
		# 9	-0.9451
5 Nodes	**Sensitivity**	**10 Nodes**	**Sensitivity**
# 1	-0.2299	# 1	-1.2406
# 2	-1.6089	# 2	-0.7619
# 3	-3.0567	# 3	-0.1274
# 4	-4.0648	# 4	-0.6876
Case 2		# 5	-1.1648
		# 6	-1.6061
		# 7	-2.0283
		# 8	-2.4770
		# 9	-3.3774

ACKNOWLEDGEMENT

This work was supported by the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea. (No. 1603010)

REFERENCES

[1] D. G. Cacuci, Sensitivity and Uncertainty Analysis: Volume I Theory, Chapman & Hall/CRC, 2003.
[2] KAERI, "MARS CODE MANUAL", KAERI/TR-2811/2004, 2009.
[3] J. J. Lee, M. G Kim, S. M. Son and J. I. Lee, "A study on the applicability of adjoint based optimization method for nuclear thermal hydraulic system analysis code nodalization", BEPU 2018, May 13-19, 2018, Real Collegio, Lucca, Italy.