

UNCERTAINTY MANAGEMENT IN SCIENTIFIC NUMERICAL COMPUTATION

SHIGEO KAWATA¹, TAKASHI ISHIHARA^{1,2}, DAISUKE BARADA¹, JIANG XIE²,
HUIRAN ZHANG², DOMBO DAI², YASUHIKO MANABE³, YOSHIKAZU HAYASE⁴,
HITOHIDE USAMI⁵, TAKAYUKI TERAMOTO⁶, AND MASAMI MATSUMOTO⁷

Abstract. The uncertainty in scientific computing has induced serious accidents and disasters. New proposals are presented toward an uncertainty inference or management in scientific numerical computing. The first important point is to share the knowledge on uncertainty with other researchers. Sharing known uncertainties contributes to the uncertainty management. In addition, for specific problems, one can prepare multiple programs with different rounding methods and / or with different precisions, and can compare the results among the programs generated for the specific problems. If differences appear among the results, it suggests that the specific program may have some uncertainty problems, and has a lack of the precision or a lack of the digit number. This second method provides a simple and effective inference method of the uncertainty. Uncertainty comes from various sources from physical and mathematical model errors, unknown input data, numerical model errors, insufficient numerical precision, floating point precision, programming errors, data processing errors, visualization errors, etc., as well as human errors. Another uncertainty comes from the discretization step size of Δt or Δx in numerical computations. The discretization step size of Δt or Δx must be selected appropriately in numerical computations in order to describe short waves or fast phenomena concerned to the target problems. This uncertainty would be also reduced by multiple program computations with the different size of Δt and Δx to find out the appropriate step size under keeping numerical stabilities. These uncertainty reduction mechanisms are proposed in this paper.

Key words. scientific computing, uncertainty, validation, verification, computer assisted science, PSE (problem Solving Environment), uncertainty quantification.

1. Introduction

Scientific numerical computing has become a powerful and key method to study scientific issues, to develop and design new products, to discover new physics in science and to provide a perspective for decision making, together with theoretical and experimental methods. For example, global warming problems have been discussed and studied based on computer simulation predictions on temperature increase, sea level elevation, human population, economical trend, energy resource consumption, etc. [1]. Computer simulations have also contributed to scientific discoveries, innovations and new findings. In physics, chemistry and other disciplines, mathematical equations including PDEs (partial differential equations) are employed to model phenomena concerned. The mathematical equations might be discretized so that the equations can be treated and solved on computers. In computer simulation, then numerical data are obtained and analyzed on physical quantities of interests (QOI). Not always but frequently QOI is visualized.

On one hand, numerical computations and simulations are always under threats of uncertainty, that includes model errors, numerical errors, bugs, data analysis errors, etc. In 2009 Air France447 met blocking of all the Pitot tubes by which airplanes measure their speed [2]. The Pitot tube holes were blocked by the condensed super cooled moisture. That means that the speed of the airplane becomes low, and

the computers started to speed the airplane up. The three pilots could not find the reason for the acceleration. Finally at the steep attack angle the airplane stalled and was crashed in the ocean. All 228 people were killed by the accident. In this disaster, the input data was wrong from the Pitot tubes to the computers. Another accident happened at the Gulf war in 1991, and a missile killed unpredictably 28 people [3]. This accident came from an insufficient software precision.

On the other hand, PSE (Problem Solving Environment) studies have been explored intensively to support users of software and hardware for problem solving. PSE studies were started in 1970s to provide initially a higher-level programming language rather than Fortran, etc. in scientific computations. PSE is defined as follows: A system that provides all the computational facilities necessary to solve a target class of problems. It uses the language of the target class and users need not have specialized knowledge of the underlying hardware or software by John Rice [4]. PSE provides integrated human-friendly innovative computational services and facilities to enrich science and our society. In the PSE concept, human concentrates on target problems themselves, and a part of problem solving process, which can be performed mechanically, is performed by computers or machines or software.

So far, many PSEs have been developed and have contributed to solve problems: program generation support PSE [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], job execution support PSE on cloud or grid [16, 17], an education support PSE [18], etc. Though computer simulation is a powerful tool to solve and study scientific problems, powerful computer simulations are always facing to the threat of uncertainty. The threat of the uncertainty does not always appear but cannot be ignored [19]. Uncertainty in scientific computing could be managed by PSEs [20].

Toward the uncertainty management in scientific computing, we will discuss the origin and characteristics of the uncertainty and present proposals based on the PSE concept in this paper. In this paper three methods are presented toward the reduction of uncertainty: uncertainty knowledge sharing, implication of the precision or digit number required, and selection of the appropriate discretization step size.

2. Origin of Uncertainty in Scientific Numerical Computation

The uncertainty in computer simulation has a wide variety. Below is the summary of the origin of uncertainty:

- Physical model error or uncertainty, including unknown physics.
- Mathematical model error.
- Computing model error including computing algorithm error, discretization errors, etc.
- Numerical error, including errors in floating-point computations and rounding error. Numerical stability is also essential to obtain reasonable results.
- Input data uncertainty, including unknown input data and boundary conditions.
- Output data processing error, including visualization error.
- Measurement error, when computations are cooperated with measurement equipments.
- Human errors.

From real physical phenomena, physical model is constructed to find out which physics is concerned. In this process, some physics involved could be missed, and it may lead to uncertainty to describe the real phenomena. From physical model, mathematical model is derived. Mathematical model does not always present

the real world. Sometimes exact equations are not known, or some perturbations are ignored, which may be essentially important in the phenomena. The mathematical model may often include PDEs, which should be discretized to be solved on computers. In the discretization, well-known numerical instabilities may appear. In the computing program the numerical stability condition must be always fulfilled during a whole computation. If the stability condition is violated during computation, the numerical results do not meet the validity. The floating-point error is another issue in computer simulation, because recently long computing time on super computers becomes common to obtain meaningful results. A finite digit number is used in computers to describe real numbers and to perform arithmetic operations. This induces the floating-point errors, including rounding errors and truncation errors. For the computations, input data and boundary data should be prepared. Sometimes the input data is measured, and in this case the measurement itself may have some errors. It is difficult to find the exact input data, which may induce another source of uncertainty. We may approximate the input data. After or during computations, output data come up and the data processing is needed to find characteristics of QOI or so. We also often perform scientific visualization. In the visualization process, we could find some uncertainty depending on the visualization method, precision and so on [21]. Sometimes hidden important structures or features could not be found in the simple visualization, or a surface position may not be exact. Human errors also share the contribution to uncertainty with other issues discussed above. When a software gives a wrong result for users, it may cause some difficulties, errors and accidents, depending on target problems. The validation and verification mechanism is essentially important as useful softwares. This was pointed out by J. Rice and his colleagues [22]. Standardization and benchmark problems in each field may help to perform the validation and verification. In addition, uncertainty management must be addressed intensively in order to avoid serious accidents and disasters in our society. PSE is one of candidates to manage the uncertainty in a relatively easy way [20].

3. Toward uncertainty management

Uncertainty, verification and validity in scientific computing have been recently studied intensively [19, 20, 23, 24, 25, 26, 27]. On the other hand, each uncertainty has its own origin, and has very different characteristics with one another, as we have discussed in Section II. Just one solution might be insufficient to manage all the uncertainty. This consideration suggests us multiple solutions in the various directions of the uncertainty characteristics. One promising way is to develop a PSE for the uncertainty management [20, 27, 28].

3.1. A PSE for Sharing Uncertainty Knowledge. The PSE proposed in this paper should have multiple solutions. For example, the PSE for uncertainty management should have known solutions, including self-validating method, program reliability test function, floating-point error control, as well as mechanical program generation assistant function [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. Each known solution is not yet perfect and is still under study. PSEs for mechanical program generation are also not almighty, but work for a specific limited area, for example, FEM (finite element method)-based program generation, PSEs for PDEs-based problems by FDM (finite difference method), MPI-based parallel program generation support, document generation support, etc. However, some part of uncertainty could be already covered or managed by the present PSE research results.

In addition, the sharing function of known uncertainties and their solutions is significantly important to avoid errors and accidents so that one can find what kind of uncertainty could happen and what is known solutions. PSE is good at working for sharing the uncertainty.

For the uncertainty knowledge sharing, we have propose one mechanism in a PSE, called PSE Park [20], which is a kind of PSE for PSE: PSE Park is a meta PSE system or framework enabling us to construct PSEs easily. In the near future, many users may start to do simulations for their hobbies, for a better lifestyle, etc. PSE Park is a framework to enable the construction of PSE by combining those existing functions and functions that are newly developed. In PSE Park users can select Cores which are components for a PSE, and connect them to build up a new PSE. A new function against the uncertainties in scientific computing is also implemented. When users register a new Core, that is, a function composing a PSE, PSE Park requests the user to input information for the uncertainties relating to the function. For example, an applicable range of the method employed or the parameters specified in the Core, etc. This new function of PSE Park enhances to share the information or knowledge about the uncertainties, and it may contribute to reduce the uncertainty risk.

This idea could contribute also to reduce human errors. Human is not always perfect. We cannot always avoid human errors. The knowledge sharing function in PSE can also contribute to share the knowledge.

3.2. Uncertainty Implication Based on Different Rounding Methods and

Precisions. One of uncertainty sources is the rounding error. In IEEE 754, the four rounding methods are specified: Round to nearest (Even) (RZ), Round Upward (RP), Round Downward (RM), Round toward 0 (RZ). Not always but in many cases, computation results by vulnerable or sensitive computations are strongly influenced by the rounding methods. Here we call a software containing uncertainty a sensitive software. So we would presume that computational results from the sensitive software may be influenced by the rounding methods.

An example result is shown in Fig. 1 for $f_1(x) = x$ and

$$f_2(x) = \{-\ln(\exp(-x^{-4}))\}^{-1/4}$$

based on the IEEE rounding methods [19]: Round to nearest (Even) (RZ), Round Upward (RP), Round Downward (RM) and Round toward 0 (RZ). $f_2(x)$ should be equal to $f_1(x) = x$ mathematically. In Fig. 1 the computation was performed with the double precision. In this case there are significant differences among the results depending on the rounding methods. This result shows that the double precision is not sufficient to obtain a sufficiently accurate result, and the digit number used was not sufficient for this specific case.

Figure 2 shows the results for $f_1(x) = x$ and $f_2(x) = \{-\ln(\exp(-x^{-4}))\}^{-1/4}$ in the quad precision. There is no difference among the results. This means that the quad precision is sufficient for $f_2(x) = \{-\ln(\exp(-x^{-4}))\}^{-1/4}$. The results in Figs. 1 and 2 demonstrate that the rounding method would work as a probe to detect the sufficient digit number used for specific numerical problems.

Based on this consideration, we propose to use a PSE for automatic program generation [5, 6, 7, 9, 10, 13, 14, 15, 16, 17] to detect the sensitivity of the software to the digit number or the precision. This could be relatively easy in PSE. When a program is generated by the PSEs, for example, the 4 programs are generated corresponding to the 4 rounding methods. The 4 computer programs provide independent numerical results for one specific problem. If the PSE has a comparison

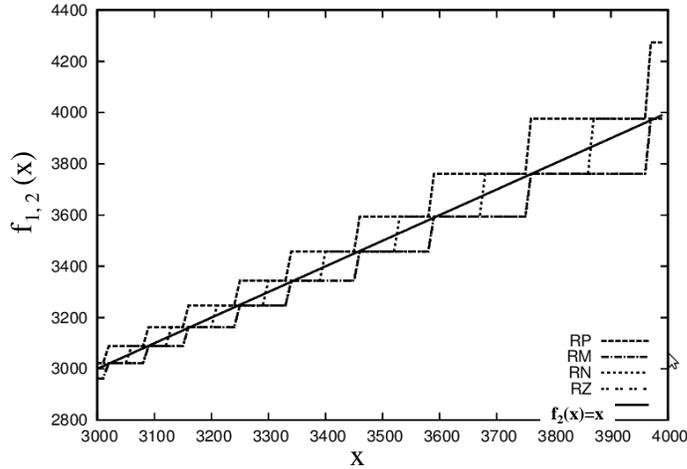


FIGURE 1. Numerical results in the double precision for $f_1(x) = x$ and $f_2(x) = \{-\ln(\exp(-x^{-4}))\}^{-1/4}$ [19] based on the IEEE rounding methods: Round to nearest (Even) (RZ), Round Upward (RP), Round Downward (RM) and Round toward 0 (RN). For the sensitive case of $f_2(x)$, a large difference appears among the numerical results based on the 4 IEEE rounding methods. Figure 1 demonstrates that the rounding method would work as a probe to detect the precision requirement, that means the digit number requirement.

function among the results, we can easily find the difference among the results. If the numerical results depend on the rounding methods, we can suspect that the software may have some uncertainty. That means that the software is sensitive against the rounding errors. In this case the software should be developed further to reduce the uncertainty by a larger number of digits used or by a higher precision.

This method is rather generic, and is easily implemented in a PSE framework. In addition to this uncertainty detection method, similarly we can also generate multiple programs by changing the computation precision, and the comparison among the generated programs with different precision provides a possibility of uncertainty existence in the specific program.

We applied this suggested method to estimate the numerical errors in a shock wave problem in a fluid. Figure 3a) shows the result for the shock wave propagation. At the single precision, the numerical results present significant differences depending on the rounding methods (see Fig. 3b). Figure 3c) shows the results at the double precision. The results at the double precision provide better results. The quadruple precision presents sufficiently accurate results as shown in Fig. 3d).

3.3. Discretization Step Size Control under Numerical Stability Condition. One of other uncertainties comes from the discretization step size, which is the time step Δt or the spacial step size Δx . Even for a fixed value of Δx one could obtain a numerical result; when Δx is too large to represent a spacial distribution,

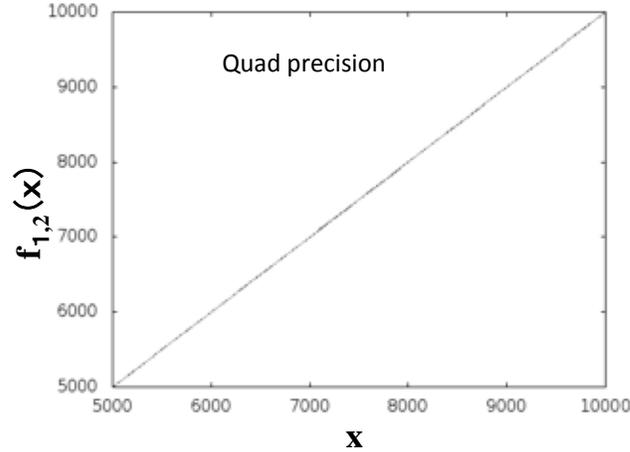


FIGURE 2. Numerical results in the quadrature precision for $f_1(x) = x$ and $f_2(x) = \{-\ln(\exp(-x^{-4}))\}^{-1/4}$ based on the IEEE rounding methods: Round to nearest (Even) (RZ), Round Upward (RP), Round Downward (RM) and Round toward 0 (RZ). For $f_2(x)$, no difference appears among the results. In this specific case, the quad precision is sufficient to eliminate the rounding errors.

short wavelength waves cannot be represented precisely. The insufficient discretization step size would result in one kind of numerical uncertainty. When the time step Δt is too large, short time scale phenomena cannot be expressed exactly. The insufficient Δt or Δx leads another kind of uncertainty in numerical computations.

In order to reduce the uncertainty coming from the insufficiently large Δt or Δx , one should use a sufficiently small value for Δt or Δx . The problem is what the sufficiently small Δt or Δx is. The reasonable Δt or Δx cannot be obtained a priori. We also cannot go to the infinitesimally small values of Δt or Δx for a reasonable computation time.

In addition, numerical stability conditions must be fulfilled during the computation. For example, a simple diffusion equation $\partial f/\partial t = \partial^2 f/\partial x^2$ would be discretized explicitly as follows: $f_i^{n+1} = f_i^n + (\Delta t/\Delta x^2)(f_{i+1}^n - 2f_i^n + f_{i-1}^n)$. The superscript shows the time index, and the subscript shows the spatial index. In this discretization explicit method the stability condition is $\Delta t/(\Delta x)^2 < 1/2$. When Δt or Δx is reduced, the stability condition must be always fulfilled during the computations.

Table 1 shows relative differences among the numerical results for the diffusion problem, in which the discretization step size is changed under the stability condition of $\Delta t/(\Delta x)^2 < 1/2$. When the sufficiently small Δt or Δx are selected, the numerical results do not change much. Then one can assume the corresponding Δt and Δx are sufficiently small to describe the short waves and fast phenomena.

In a PSE framework, this uncertainty management procedure can be easily implemented: when a numerical program is generated by hand or by a program generation PSE, the programmer or scientists would include the numerical stability

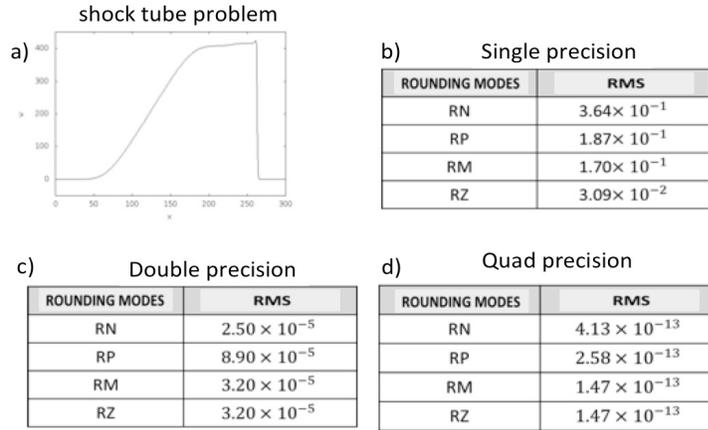


FIGURE 3. A shock tube problem in fluid dynamics is solved by multiple programs with the different rounding methods and precisions (see Fig.3a)). The single precision shows a significant difference among the computational results depending on the rounding method as shown in Fig. 1b). The double precision program provides better results as shown in Fig. 3c). However, the quadruple precision shows sufficiently accurate results (see Fig. 3d).

TABLE 1. Change in Numerical Results depending on Discretization Step Size

Δx	Δt	Relative Difference
$1/2\Delta x$	$1/4\Delta t$	1.11×10^{-2}
$1/4\Delta x$	$1/16\Delta t$	4.45×10^{-3}
$1/8\Delta x$	$1/64\Delta t$	2.06×10^{-3}

condition for the program. The PSE could submit several job for the specific problem with the different Δt and Δx together. After getting the numerical results, the users compare the results and find the appropriate Δt and Δx for the problem.

This uncertainty management method is also general for numerical computations to find out the appropriate Δt and Δx for specific problems.

4. Conclusions

In this paper we have discussed the origin of uncertainty in computing sciences first. Then we stressed three important issue and proposals: the first important point is to share the knowledge on uncertainty with other researchers and engineers. Sharing the previous or known uncertainties with others contributes to the uncertainty management. Probably this is the essentially important and realistic solution for the uncertainty management. The second is that PSEs can provide a reliable tool to manage the uncertainty to avoid future accidents and disasters originated from the uncertainty. For specific problems, the PSE provides multiple

programs with different precisions or with different rounding methods, and then we can compare the results among the programs generated for the specific problems. If some differences appear among the results by the multiple programs generated by the PSE, it suggests that the specific program may have some uncertainty problems. The third proposal is to supply the information for the appropriate discretization step sizes of Δt and Δx . When Δt and Δx are sufficiently small, the numerical results do not depend on the size of Δt and Δx . When the differences appear among the numerical data from the programs using the different Δt and Δx and the differences are visible, it suggests the users or scientists to reduce Δt and Δx to obtain the accurate results. These three approaches and proposals are realistic and rather general toward the uncertainty reduction.

Acknowledgments

The work is partly supported by JSPS, MEXT, CORE (Center for Optical Research and Education, Utsunomiya Univ., Japan), Shanghai University, China, the Major Research Plan of NSFC [No. 91330116] in China, and PSE (Problem Solving Environment) Research group Japan. The authors would like also extend their thanks to Prof. Wu Zhang in Shanghai University for their fruitful discussions.

References

- [1] United Nations Environment Programme, Global Environment Outlook (GEO-5), www.unep.org/geo/pdfs/geo5/GEO5_report_full_en.pdf, 2012.
- [2] www.bea.aero/fr/enquetes/vol.af.447/vol.af.447.php.
- [3] www.fas.org/spp/starwars/gao/im92026.htm.
- [4] Gallopoulos, E., Houstis, E., Rice, J.R., Computer as Thinker/Doer: Problem-Solving Environments for Computational Science, IEEE Computational Science & Engineering, 1, 11-23, 1994.
- [5] Boonmee, C., Kawata, S., Computer-Assisted Simulation Environment for Partial-Differential-Equation Problem, 1. Data Structure and Steering of Problem Solving Process, Trans. of the Japan Society for Computational Engineering and Science, Paper No.19980001, 1998.
- [6] Boonmee, C., Kawata, S., Computer-Assisted Simulation Environment for Partial-Differential-Equation Problem: 2. Visualization and Steering of Problem Solving Process, Trans of the Japan Society for Computational Engineering and Science, Paper No.19980002, 1998.
- [7] Kawata, S., Boonmee, C., Fujita, A., Nakamura, T., Teramoto, T., Hayase, Y., Manabe, Y., Tago, Y., Matsumoto, M., Visual Steering of the Simulation Process in a Scientific Numerical Simulation Environment -NCAS-, Enabling Technologies for Computational Science, edited by E.Houstis and J.Rice, Kluwer Academic Pub., 291-300, 2000.
- [8] Gallopoulos, E., Houstis, E., Rice, J.R., Future Research Directions in Problem Solving Environments for Computational Science, Technical Report CSRD Report No.1259, Report of a workshop on Research Direction in Integrating Numerical Analysis, Symbolic Computer, Computational Geometry, and Artificial Intelligence for Computational Science, Washington DC., 1991.
- [9] Umetani, Y., DEQSOL A numerical Simulation Language for Vector/Parallel Processors, Proc. IFIP TC2/WG22, 5, 147-164, 1985.
- [10] Hirayama, Y., Ishida, J., Ota, T., Igai, M., Kubo, S., Yamaga, S., Physical Simulation using Numerical Simulation Tool PSILAB, 1st Problem Solving Environment Workshop Japan, 1-7, 1998.
- [11] Rice, J.R., Boisvert, R.F., Solving Elliptic Problems Using ELLPACK, Springer-Verlag, Springer Series in Computational Mathematics 2, New York, 1984.
- [12] Fujio, H., Doi, S., Finite Element Description System, as a Mid-Layer of PSE, Proc. of Conference on Computation Engineering and Science, 3, 441-444, 1998.
- [13] Okochi, T., Konno, C., Igai, M., High Level Numerical Simulation Language DEQSOL for Parallel Computers, Trans. of Information Processing Society of Japan, 35, 977-985, 1994.

- [14] Houstis, E.N., Rice, J.R., Parallel ELLPACK, a development environment and problem solving environment for high performance computing machines, edited by Gaffney, In P., Houstis, E.N., Programming Environments for High-Level Scientific Problem Solving, North-Holland, Amsterdam, 229-241, 1992.
- [15] Fujita, A., Teramoto, T., Nakamura, T., Boonmee, C., Kawata, S., Computer-Assisted Parallel Program Generation System P-NCAS from Mathematical Model-Visualization and Steering of Parallel Program Generation Process-, Transof Japan Society for Computational Engineering and Science, Paper No. 20000037, 2000.
- [16] Fuju, H., Kawata, S., Sugiura, H., Saitoh, Y., Usami, H., Yamada, M., Miyahara, Y., Kikuchi, T., Kanazawa, H., Hayase, Y., Scientific Simulation Execution Support on a Closed Distributed Computer Environment, 2nd IEEE International Conference on e-Science and Grid Computing, Paper No. 27340112, 1-7, 2006.
- [17] Kanazawa, H., Itou, Y., Yamada, M., Miyahara, Y., Hayase, Y., Kawata, S., Usami, H., Design and Implementation of NAREGI Problem Solving Environment for Large-Scale Science Grid, 2nd IEEE International Conference on e-Science and Grid Computing, Paper No. 27340105, 2006.
- [18] Teramoto, T., Okada, T., Kawata, S., A Distributed Education-Support PSE System, 3rd IEEE International Conference on e-Science and Grid Computing, 516-520, 2007.
- [19] Kahan, W., Desparately Needed remedies for the Undebuggability of Large Floating-point Computations in Science and Engineering, IFIP Working Conf. on Uncertainty Quantification in Scientific Computing, www.eecs.berkeley.edu/~wkahan/Boulder.pdf, 2011.
- [20] Kawata, S., Kobashi, H., Ishihara, T., Manabe, Y., Matsumoto, M., Barada, D., Hayase, Y., Teramoto, T., Usami, H., Scientific Simulation Support Meta-System: PSE Park - with Uncertainty Feature Information -, Int. J. Intelligent Information Processing, 3, 66-76, 2012.
- [21] Johnson, C.R., Top Scientific Visualization Research Problems, IEEE Computer Graphics and Applications: Visualization Viewpoints, 24, 13-17, 2004.
- [22] Houstis, E., Gallopoulos, E., Bramley, E., Rice, J.R.' Problem-Solving Environment in Computational Science, IEEE Computational Science & Engineering, 4, 18-21, 1997.
- [23] Rump, S. M., Verification methods: Rigorous results using floating point arithmetic, Acta Numerica, 19, 287-449, 2010.
- [24] Enright, W.H., Reducing the Uncertainty when Approximating the Solution of ODEs, (Uncertainty Quantification in Scientific Computing, Springer, Proc. 10th IFIP WG 2.5 Working Conference, WoCoUQ 2011, Boulder, CO, USA, August 1-4, 2011), 280293, 2012.
- [25] Weimer, W.R., Exceptional Situation and Program Reliability, PhD Thesis, Univ. California, Berkeley, 2005.
- [26] The US National Academy of Sciences, Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification, 2012.
- [27] Dienstfrey, A.M., Boivert, R.F., edited, Uncertainty Quantification in Scientific Computing, Springer, 2012.
- [28] Einarsson, B., edited, Accuracy and reliability in Scientific Computing, SIAM (Society for Industrial and Applied Mathematics), 2005.

1. Department of Advanced Interdisciplinary Sciences, Utsunomiya University, Utsunomiya 321-8585, Japan

E-mail: kwt@cc.utsunomiya-u.ac.jp

2. School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China

E-mail: jiangx@shu.edu.cn, hrzhangsh@shu.edu.cn

3. Numazu National College of Technology, Numazu 410-8501, Japan

4. Toyama National College of Technology, Toyama 939-0293, Japan

5. Tamagawa university, Machida 194-8610, Japan

6. Tsuyama National College of Technology, Tsuyama 708-8509, Japan

7. Yonago National College of Technology, Yonago 683-8502, Japan