



Adhika Satyadharma \* D and Harinaldi

Department of Mechanical Engineering, Universitas Indonesia, Kampus UI Depok 16424, Indonesia; harinald@eng.ui.ac.id

\* Correspondence: adhika.satyadharma91@ui.ac.id

**Abstract**: Although the grid convergence index is a widely used for the estimation of discretization error in computational fluid dynamics, it still has some problems. These problems are mainly rooted in the usage of the order of a convergence variable within the model which is a fundamental variable that the model is built upon. To improve the model, a new perspective must be taken. By analyzing the behavior of the gradient within simulation data, a gradient-based model was created. The performance of this model is tested on its accuracy, precision, and how it will affect a computational time of a simulation. The testing is conducted on a dataset of 36 simulated variables, simulated using the method of manufactured solutions, with an average of 26.5 meshes/case. The result shows the new gradient based method is more accurate and more precise then the grid convergence index(GCI). This allows for the usage of a coarser mesh for its analysis, thus it has the potential to reduce the overall computational by at least by 25% and also makes the discretization error analysis more available for general usage.

**Keywords:** computational fluid dynamics; verification and validation; discretization error; uncertainty quantification; grid convergence index



As the usage of computational fluid dynamics (CFD) is getting more common in the engineering field, it is important to ensure that the result of these CFD simulations is indeed accurate. As CFD generally works by modeling a governing equation and then solving it, both of these processes must be done accurately to ensure an overall accurate simulation. Although it is important that both processes are accurate, the accuracy of the solver/calculation should be prioritized since the process of checking that a model is accurate depends on the accuracy of the calculation itself [1,2].

The process of ensuring that a model is well calculated which is also known as the verification process [3], usually consists of two parts which is the code verification and solution verification [4]. Code verification is the process to check how accurate the implementation of the underlying algorithm within the code is [5], and solution verification is a process to check the influence of the numerical errors within the solution [6]. Since verification could be classified as a process to assess the errors in the calculation of a model [7], it is also possible to separate the two types of verification by the error that it determines. Code verification is to check if there is any programming error, and solution verification determines the effect of the numerical error within the solution.

The numerical error in CFD usually consists of discretization error, iterative error, and round-off error [8]. Among these errors, discretization error, especially spatial discretization error is the most significant error, and it is difficult to quantify [9]. This error originates from the change of the governing equation in which is meant to work in a continuum domain into a discrete domain, allowing a computer could handle the calculation. It is during this change of domain, which usually includes some simplification, that results in some information loss, thus resulting in this error.



**Citation:** Satyadharma, A.; Harinaldi The Performance of a Gradient-Based Method to Estimate the Discretization Error in Computational Fluid Dynamics. *Computation* **2021**, *9*, 10. https://doi.org/10.3390/ computation9020010

Received: 17 December 2020 Accepted: 12 January 2021 Published: 24 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Although there are a lot of ways to estimate the spatial discretization error [10], the most-used method is the grid convergence index (GCI) [11]. This method is robust, easy to use, and it does not require any change to the CFD code. This method is also recommended by some well-known organizations such as the American Society of Mechanical Engineers (ASME) [12] and also the American Institute of Aeronautics and Astronautics (AIAA) [13]. Albeit popular, the usage of GCI should be taken with caution as it does have some flaws.

The first problem is that the GCI which is based on Richardson Extrapolation [14,15] assumes asymptotic convergence, in which unless this condition is achieved, the GCI analysis is questionable [16]. This leads to second problem in which the asymptotic convergence is indicated by the local order of convergence. This is a problem since the calculation of the local order of convergence is very sensitive to any external errors [17] in which even a small error could change the local order of convergence is hard to achieve, especially for low quality/coarse meshes or high Reynolds number turbulence flow [18]. This usually leads to the requirement of a high density mesh, which significantly increases the computational load. Apart from that, when the local order of convergence is high, the GCI result tends to have an unreasonably small value. On the other hand, if the local order of convergence is small, then the GCI analysis would be reasonable, but the value itself would become less meaningful.

Although there are several research studies dedicated to improving the GCI formula [19–24], most of them are designed to solve one or two problems which have been previously mentioned. For example, Xing and Stern [22] only consider the problem of the unreasonable small value of GCI and about the confidence level of GCI. Another example is from Eca and Hoekstra [23], which creates a new formulation which is designed to overcome the GCI sensitivity problem. However, these modifications usually come with a side effect. As shown by Phillips [25] that if the GCI becomes more conservative it is due to the formula of having a higher uncertainty, which makes the uncertainty less meaningful. This shows that it is difficult to obtain both a higher accuracy and high precision for the GCI.

The difficulty of improving the GCI is not only shown by how hard it is to balance accuracy and precision, but also shown by the gap of years between each publication. The last publication with an improvement to the GCI itself was done by Phillips [24] in 2016. The more recent publications are usually more focused on the error transport equation method [26–28], which is another type of discretization error estimator that is harder to execute than the GCI.

Within this research, a new perspective on how the GCI should be calculated is presented with a target to overcome all the previously mentioned problems. This perspective has the same roots as the GCI which is Richardson Extrapolation. However, instead of calculating an error estimate to obtain the uncertainty, the uncertainty is obtained by bounding the possible values of the exact solution judging by the behavior of the gradient. Using this approach, it is possible to obtain an estimate even by avoiding the usage of local order convergence which many of the problems are dependent on. The goal of this research is to evaluate the performance of this model, both of its accuracy and its precision which is estimated to be both higher than the GCI model.

Being more accurate and precise has several advantages. First, it would allow the use of a coarser mesh to obtain the same quality of estimation. This alone may speed up the required simulation time that would be lower. This would be later proven in this study as a time comparison analysis would also be presented. Another advantage of the coarser mesh is that within the creation of the mesh itself, a coarser mesh requires less time to create. As such, more time saving would occur. Second is that there might be a possibility that this new method might be practically applicable for a more complex simulation or a higher Reynolds number flow. This would allow for more simulations to have its result verified and have a higher quality result. For the structure of this paper, first the GCI method would be explained. This would be immediately followed by the explanation of the gradient-based method from the basic premise, the assumptions used, and the derivation, until the formula is obtained. The next part would then describe of how to compare the performance of the gradient-based method to the GCI method especially the metrics and the datasets used for comparison. This would be followed by a discussion in which the performance of each method would be compared and discussed in detail. An example would then be presented to show the performance of the model in real application. Finally, the conclusion would be presented which would describe the overall performance of the gradient-based method.

## 2. Discretization Error Estimation

A solution solved in a grid spacing of h is usually expressed by the generalized Richardson extrapolation (GRE) as shown in Equation (1). As the mesh is refined, the obtainable solution will get closer to a zero error solution with an order of p. However, this method does assume that the higher order terms (*HOT*) are small compared to  $ah^p$ , thus allowing it to be ignored. This assumption is also called the asymptotic convergence, since if only the first terms remains, the solution would asymptotically converge to  $f_0$  upon refinement.

$$f_h = f_0 + ah^p + HOT \tag{1}$$

$$GCI = SF \left| \frac{f_2 - f_1}{r^p - 1} \right| \tag{2}$$

$$\frac{f_3 - f_2}{r_{23}^p - 1} = (f_2 - f_1) \left( \frac{r_{12}^p}{r_{12}^p - 1} \right)$$
(3)

Using the GRE as a basis, Roache [11] proposed the GCI to calculate an uncertainty of a solution obtained at grid spacing *h*. This uncertainty is obtained by multiplying the error estimate of the GRE by a safety factor (*SF*). The GCI was designed so that the value of  $f_0$  is between  $f_1 \pm GCI$  with a 95% confidence interval. The equation is GCI as shown in Equation (2), in which since there are three unknowns in Equation (1), three solutions are needed to solve it. As for the value of *p* which is the local order of convergence, it is calculated using Equation (3) [29].

One of the first main concerns in GCI is the value of *SF*. At first, Roache [11] proposed the value of 3, but then revised it at 1.25 [29]. The value of 1 was discarded since there is a 50% chance of the value of  $f_0$  being outside the range of  $f_1 \pm GCI$ . This is true, but only if the value of  $f_1$  is the center of the estimate. By taking another value which is possibly closer to  $f_0$ , it is possible to use a safety factor of 1 or maybe less, while still maintaining its accuracy. To give a comparison to this idea, during an experiment, the uncertainty due to random error is usually not meant or given for individual data, but the average of the data, which is a better approximation of the real value. This is the basis of the suggested model, which is to calculate an uncertainty for a more accurate representation of the real value. This comparison is also illustrated in Figure 1.

In the suggested model there are two things that needs to be calculated which are  $f_{0,guess}$ , and the uncertainty. In essence, these two values are needed to calculate the possible range of  $f_0$  such as that the minimum possible value and the maximum possible would be known. Apparently, this relationship also applies in a reverse manner in which by knowing the minimum and maximum possible range of  $f_0$ , the value of  $f_{0,guess}$ , and the uncertainty could be obtained. This relationship is shown in Equations (4) and (5) in which the variables *a* and *b* denote either the maximum or minimum possible range of  $f_{0,guess}$ . By using these equations, the problem now lies on estimating the range of  $f_0$  in which some assumptions are going to be made to obtain theses values.

$$f_{0,guess} = \frac{a+b}{2} \tag{4}$$

$$U = \left| \frac{a - b}{2} \right| \tag{5}$$

Based on GRE, if the solutions are within the asymptotic range, then the function of  $f_h$  should be in the most part monotonic. There would be some wiggles in the data due to the *HOT*, but within the scale of  $f_2 - f_1$ , these *HOT* should be able to be ignored. The monotonic assumption would also apply to the derivative of  $f_h$  which is denoted by g. Using this monotonic assumption as a basis, if 1 < p, then the limiting value of  $f_0$  could be modeled as Equation (6), with the value of  $g_{12}$  calculated with Equation (7). This modeling is possible since when p > 1, the value of  $|g(h \le h_1)|$  would decrease as the cell size approaches zero. The limiting value of  $f_0$  is then taken by considering  $g(h \le h_1) = g_{12}$  and  $g(h \le h_1) = 0$ , in which upon integration resulted in Equation (6). This limiting value is also illustrated in Figure 2, in which line A1 represents the condition of  $g(h \le h_1) = g_{12}$  and line A2 represents  $g(h \le h_1) = 0$ .

$$if \ 1$$

$$g_{12} = \frac{f_2 - f_1}{h_2 - h_1} \tag{7}$$

$$g_{23} = \frac{f_3 - f_2}{h_3 - h_2} \tag{8}$$



**Figure 1.** An illustration of how uncertainty is calculated: (**a**) within computational fluid dynamics (CFD); (**b**) within an experiment.



**Figure 2.** A visual representation of how the limiting value is calculated with the suggested model: (**a**) on the simulation data; (**b**) on the gradient.

Using the same approach, which is a gradient based analysis, the limiting values of  $f_0$  for the condition of  $0 could also be obtained. One of these limiting value is <math>g(h \le h_1) = g_{12}$ . As for the other limiting value, some additional modeling is required. This is due to  $|g(h \le h_1)|$  in which it always increases as the cell size approaches zero. A quick glance would results in  $|g(h \le h_1)| = \infty$ , which is unacceptable. As such, it is required to find an estimation of  $g_0$ , so that the possible values of  $g(h \le h_1)$  is limited.

After examining several datasets which satisfy the condition of 0 , it is found that the inverse of the gradient <math>(1/g) tends to behave linearly. Using this as a basis, Equation (9) is formed in which it shows how to calculate  $g_0$ . A linear relationship with either  $h_1$ ,  $h_2$ , or  $h_3$  is not done, since both  $g_{12}$  and  $g_{23}$  which is calculated by using Equations (7) and (8) are both estimates of the real gradient itself (e.g.,  $g_{12} \neq g_1$ ,  $g_{12} \neq g_2$ ). To ensure that the estimated  $g_0$ would lead to a conservative estimate of  $f_0$ , some modifications are needed, which in this case it is done by estimate of h which equal to  $g_{12}$  and  $g_{23}$ , respectively (e.g.,  $|g_{h12,min}| \leq |g_{12}| \leq |g_{h12,max}|$ ). These values are shown in Equations (10) and (11).

After obtaining the value of  $g_0$ , the other limiting value of  $f_0$  could then be calculated, in which it occurs at  $g(h \le h_1) = (g_{12} + g_0)/2$ . This average form is taken due to its simplicity and robustness. Although as stated before that the inverse of the gradient tends to behave linearly, this model is not used as the value  $g_0$  is still considered as a guess. This could lead to some conservative issues, in which the model of  $g(h \le h_1) = (g_{12} + g_0)/2$  is preferred as it is more conservative. Hence, the limiting values of  $f_0$  for the condition of 0 < p< 1 is  $g(h \le h_1) = g_{12}$  and  $g(h \le h_1) = (g_{12} + g_0)/2$ . Upon integration, it would result in Equation (12) and, respectively, shown in Figure 2 as line B2, and B1.

$$\frac{1}{g_0} = \frac{1}{g_{12}} - \left(\frac{1}{g_{23}} - \frac{1}{g_{12}}\right) \frac{1}{[h_{23,min}/h_{12,max}] - 1}$$
(9)

$$h_{12,max} = \frac{h_1 + h_2}{2} \tag{10}$$

$$h_{23,min} = \frac{5r_{23} + 7}{(r_{23} + 11)(r_{23} + 1)}(h_3 - h_2) + h_2$$
(11)

$$if \ 0 (12)$$

Some issues persist with Equations (6)–(12). Firstly, the estimate in the new model is discontinuous when p = 1. As some cases do tend to behave as  $p \approx 1$ , this discontinues estimate could lead to some sensitivity issues. To alleviate the problem a safety factor was used so that the condition of where Equation (6) applies is set to 1.1 < p and for Equation (12) it only applies if 0 . Furthermore, to alleviate some concerns of the misinterpretation regarding the value of <math>p due to its sensitivity, the condition of where Equation (6) applies is set to  $1.1 |g_{12}| \le |g_{23}|$ , and Equation (12) applies if  $1.1 |g_{12}| > |g_{23}|$ . For the sake of simplicity though, the condition of  $1.1 |g_{12}| \le |g_{23}|$  would be referred as condition A, and the condition of  $1.1 |g_{12}| > |g_{23}|$  would be referred as condition B.

For the second problem, when  $p \approx 0$ , the estimation of  $g_0$  becomes unrealistic. All gradients ( $g_{12}$ ,  $g_{23}$ , and  $g_0$ ) should have the same sign, either positive or negative. A change in sign would suggest an oscillation in the simulated data (f), which breaks the basic assumption of this method. However, when the value of p is near 0, there is a tendency that the value of  $g_0$  would have a different sign from both  $g_{12}$  and  $g_{23}$ . To alleviate this problem, Equation (13) is used.

$$if \ g_0 g_{12} < 0 \ \to \left\{ \begin{array}{l} a = -99f_1 \\ b = 101f_1 \end{array} \right. \tag{13}$$

Due to the difficulty in estimating a reliable estimate in the condition of  $p \approx 0$ , within Equation (13), a wide estimate range is taken to ensure the conservativeness of such case. The value of  $-99f_1$  and  $101f_1$  is taken to have the value of  $f_1$  to be the center of the estimate, and also it is considered to be less likely that the value of  $f_0$  would have a change of

sign from the value of  $f_1$ . Though there may not many useful practical usages from this equation, since the uncertainty is guaranteed to be 20000%, this equation is meant to evade any misleading estimate due to an unrealistic  $g_0$  value.

The third problem is the possibility if a mesh used for the simulation is not dense enough that it has a behavior of condition B, which upon refinement, changes to condition A. This leads to some conservative issues in Equation (12), which to ensure a conservative estimate, it is best to substitute the limiting value of  $f_1 - g_{12}h_1$  in Equation (12) with  $f_1$ . The final problem is again in Equation (12), in which to ensure conservativeness, the variable of  $h_1$  should be substituted by  $h_{12max}$ .

To summarize, the complete model of the suggested method is shown in Equations (4), (5), (14), and (15). The values of  $g_{12}$ ,  $g_{23}$ ,  $g_0$ ,  $h_{12max}$ , and  $h_{23mn}$ , are calculated using Equations (7)–(11), respectively. The definition of condition A is if  $1.1 |g_{12}| \le |g_{23}|$  and  $g_0g_{12} > 0$ , while the definition of condition B is  $1.1 |g_{12}| > |g_{23}|$  and  $g_0g_{12} > 0$ .

$$a = \begin{cases} f_1 & if \ Condition \ A \ or \ B \\ 101 f_1 & if \ g_0 g_{12} < 0 \end{cases}$$
(14)

$$b = \begin{cases} f_1 - g_{12}h_1 & \text{if Condition A} \\ f_1 - \left(\frac{g_{12} + g_0}{2}\right)h_{12max} & \text{if Condition B} \\ -99f_1 & \text{if } g_0g_{12} < 0 \end{cases}$$
(15)

# 3. Evaluation Method

For the evaluation of the suggested method, a dataset of 11 different cases obtained by using the method of manufactured solution (MMS)[30] is prepared. From these cases, a total of 36 variables are taken for evaluation. For the detail of each case, the variables that are being evaluated, the number of mesh used for each case, and also the MMS equation; all of this could be seen in Table 1.

No	Case	Variable	Total Mesh	MMS Equation
1	1D Heat Equation (First and Second	т	26	$- T = 50exp(-t/20)\sin(\pi x)$
2	order accurate)	1	35	
3	Laplace Equation	∬ u dxdy	25	$u = ysin(\pi x)\sin(\pi y^2)$
4	1D Convection+ Difusion Equation	Т	44	T = (1 - 0.5t)a $a = (100 - \exp(x \ln(90)))$
5 6 7	Hagen–Poiseuille	$v_{max} \Delta P$		$u = -48y^2 + 0.03$ $v = 0$
8 9	(First and Second order accurate)(Uniform Mesh)	$v_{max} \Delta P$	32	
10		$ au_{wall}$		
11 12 13	Hagen–Poiseuille (Non-Uniform Mesh)	$rac{arpi_{max}}{\Delta P}  au_{wall}$	21	
14 15 16	Modified Hagan-Poissuilla-	$\Delta P \\ \overline{u}$	20	<i>if</i> $y < -0.02$ : u = 0.052 + 2.08y <i>if</i> $u > 0.02$ :
17 18	piecewise x-velocity	$\tau_{wall}$ $\partial u / \partial y_{wall}$	50	u = 0.052 - 2.08y $else: u = -52y^2 + 0.0312$
19 20 21 22	Annulus Flow	$\overline{u} \ \Delta P \  au_{wall} \ v_{max}$	19	$u = -10^{3}(a - 0.045)^{2} + 0.625$ $a = \sqrt{y^{2} + z^{2}}$

Table 1. Details of the method of manufactured solution (MMS) simulations.

No	Case	Variable	Total Mesh	MMS Equation
23		$\overline{u}$		
24	Boundary Layer	$\overline{\partial u}/\partial x_{y=0.2}$	20	$u = u_{\infty} - u_{\infty} \exp(-50y)$ $v = -\frac{30}{2}u_{\infty} (a+1)$
25	Doundary Enger	$ au_{wall}$	52	$a = -(50y + 1) \exp(-50y)$
26		$\Delta P$		$u = (00y + 1) \exp(-00y)$
27		$\Delta P$		$u = -48y^2 + 3$
28		$v_{max}$		v = 0
29	Diffuser	$\Delta P$	17	$u = -400a^3y^2 + a$
30		$\partial u / \partial y_{wall}$		$v = -2400a^2y^2 + 4a^2y$ $P = -96\mu_{water}x$
31		$v_{max}$		a = 0.25/(x + 0.25)
32		P <sub>inlet</sub>		$u = -3a^2y^2 + 3$
33	Nozzle	$u_{max}$	25	$v = 0.4a^3y^3$ $P = 50x$
34		$\partial u / \partial y_{max}$		a = 5/(-x+1.25)
35	Consister Floor	u <sub>lid,max</sub>	20	1
36	Cubity Flow	$\overline{u_{lid}}$	28	$u_{lid} = 1$

Table 1. Cont.

For each case in Table 1, it is simulated in one of two different applications which are also operated between two different computers. The application includes a self-coded python code in JetBrains PyCharm and Fluent. As for the computers, one is using an 8-core processor (Intel i5-8256U) with an 8 GB of RAM, and the other with a 24-core processor (AMD Threadripper 3960x) with a 32 GB of RAM. As each application is set on its default settings, the number of cores available for the calculation is 7 or 23 cores for JetBrains PyCharm and 1 core for Fluent.

To ensure the validity of the simulation, first the simulations are conducted by using a 64-bit floating number (double precision) to evade any interference from the round-off error [31] and simulated to machine accuracy limit to evade any potential iterative error. The result of this simulation is then compared to the MMS equation to ensure that it is indeed solving the right equation.

To be able to measure the time to simulate each mesh, the computer is set to safe mode to prevent any kind of interference, and also the simulation is conducted multiple times to ensure that the measured time is valid. Although the simulations are conducted on different programs and computers, for each case in Table 1, both of these factors would remain the same, thus it is possible to do a time comparison for each of the meshes in each case.

As some of the data obtained within this simulation do show oscillating behavior, some of the data are bound to be rejected. This rejection only occurs in the coarsest mesh of each simulation, indicating that those meshes are outside the asymptotic range. After rejection, on average, each variable would have data from 23 different meshes. As every non-oscillating triplet would be used for analysis, the number of triplets that is going to be used is 95,992 triplets.

There are three overall tests that are going to be conducted by accuracy, precision, and required computational time. The accuracy test measures how often the  $f_0$  estimate does contain the real  $f_0$  value. This could easily be done as the MMS method allows us to know the analytical solution of the simulation. The precision test is measuring how tight of an estimate is given by a method. This test is conducted by measuring how often the uncertainty or the GCI of a specific method is below a standard statistical acceptance of 10%, 5%, and 1%. For the computational time analysis, this measures the minimal real time that is needed for a computer to simulate three meshes, so that the data have an uncertainty below 10%, 5%, and 1%, if they are analyzed by a specific method. For the sake of convenience, the accuracy test is going to be measured by the conservativeness

parameter defined in equation 16, with the value of  $f_{0,guess} = f_1$  and U = GCI for the GCI method. As for the precision test, this is going to be measured by a probability parameter shown in equation 17, with  $U_{Target}$  is set to 10%, 5%, and 1%. All of these tests are going to be conducted with the gradient based method, Roache's method, and also some improvement of Roache's method which is proposed by Oberkampf and Roy [16], Stern-Wilson [19,21], and Xing and Stern [22]. Each of these methods will be denoted as  $U_{gradient}$ ,  $GCI_{R}$ ,  $GCI_{OR}$ ,  $GCI_{SW}$ , and  $GCI_{XS}$ .

$$\Psi = \frac{\sum Cases with \left(f_{0,guess} - U \le f_0 \le f_{0,guess} + U\right)}{\sum Total Number of Cases}$$
(16)

$$\Gamma = \frac{\sum Cases with(U \text{ or } GCI \leq U_{Target})}{\sum Total Number of Cases}$$
(17)

### 4. Results

For the result of the accuracy test, the value of  $\Psi$  can be seen in Table 2. There are three different values of  $\Psi$  in this analysis, which aim to minimize the effect of any possible kind of bias within the analysis, for example due to the different number of data points on each case. For  $\Psi_{\text{Overall}}$ , this is the value of  $\Psi$  from all of the 95,922 triplets. For  $\Psi_{\text{Case}}$ , this is the average value of  $\Psi$  on every case. For  $\Psi_p$ , this is the averaged value of  $\Psi$  from a group of 31 intervals of *p*, each with an interval of 0.1 in the range of 0 and a separate individual group for <math>3 < p.

Table 2. Result of the Accuracy Test.

Method	$\Psi_{Overall}$	$\Psi_{Case}$	$\Psi_p$
GCI <sub>R</sub>	90.49%	87.27%	81.11%
GCI <sub>OR</sub>	96.95%	96.13%	91.84%
GCI <sub>SW</sub>	96.11%	94.68%	90.16%
GCI <sub>XS</sub>	97.85%	95.05%	91.92%
Ugradient	98.94%	96.75%	92.56%

From the result of the accuracy analysis it could be seen that for every variation of  $\Psi$ , the value of  $\Psi$  from  $U_{gradient}$  is the highest of them all. This consistent result across all variation of  $\Psi$ , shows that the gradient-based method does indeed perform better than all the other methods. This means that even though the gradient-based method uses the value of  $f_{0,guess}$  as the center of the estimate, as the model is built upon this assumption, the accuracy is not affected at all, and this method manages to estimate the value of  $f_0$  better than the other methods.

To understand where each method failed, the failure rate of each method based on its *p* value is shown in Figure 3. In general, the failures could be divided into three parts, which are  $0 . For the range of <math>0 , the failure could only be seen for <math>U_{gradient}$ . This failure is due to the estimation of  $g_0$  which is sometimes not conservative enough. For the range of 2 < p, the failures are mainly caused by triplets that include data from some very coarse mesh with have a behavior that is different from the rest. These data might as well be considered outside the asymptotic range, thus by including them in the analysis, they are expected to produce a high failure rate. The other source of failure only applies to  $GCI_R$ , in which it could be seen that in this range, the failure rate is significantly higher than the other methods. The cause of this failure is due to the value of *p* that exceeds the formal order of accuracy ( $p_f$ ) which is usually 1 or 2. This would cause the value of  $GCI_R$ , in which the other variations of GCI have tried to prevent. As such, they tend to have a lower failure rate.



Figure 3. Failure rate of each method, based on the value of *p*.

As for the range of  $1 , for <math>GCI_R$ , the failures are caused by the same reasons as the failures for the range of 2 < p, which are some coarse meshes that behaves differently and also a p value that exceeds the value of  $p_f$ . This is possible since the failure at the range of 2 < p has the properties of  $p_f = 2$ . However, within the range  $1 , the failures are caused by the data that have the properties of <math>p_f = 1$ . As such the same reason of failures applies, although there are some differences that need to be addressed. First, within the data with  $p_f = 2$  there are some data with the p value lower than what tends to be accurate. This would have an effect of the failures in the range of 1 , which is not as significant as the failures in the range of <math>2 < p. Second, is that for the data with a p value that exceeds  $p_f = 1$ , the deviation is usually not that far off. As such the value of p is found to be limited to p < 2. This caused a drastic reduction in failure rate from  $1 to <math>2 . As for the variations of <math>GCI_R$ , the same reasons also apply. However, as each method has tried to improve upon  $GCI_R$ , the failures are usually less than  $GCI_R$  with a varying degree of success.

For  $U_{gradient}$ , within the same range of 1 , the failures are caused mainly bya wrong usage of the equation for condition A, in which the data do mathematicallyfit the description of condition A, although the overall trend shows that condition Bshould actually be applied. With the boundary between condition A and B is around 1.1,this caused the failures to be focused around this value and above it. As such, a highfailure rate is not observed for <math>1 < p, but the failure rate is not only focused on 1 ,but also spreads to <math>1 . However, it should be known that the misinterpretation ofthe conditions is more likely to occur near <math>p = 1. As such the failure rate would decrease on a higher value of p.

For the precision test, this result is shown in Table 3. It could be seen that there are again 3 variations of  $\Gamma$ , which is again to minimize any possible bias. These variations have a similar definition as the  $\Psi$  variations in the accuracy test. It should be known that the data in Table 3 are required to be conservative, since if the data are not conservative, regardless if the uncertainty is high or low, it would be meaningless if it is not accurate. From all of these variations, there are two things that have to be considered. First is that the value of  $\Gamma$  would be higher for a higher  $U_{Target}$ . This is obvious since a tighter estimate would be harder to obtain, thus the value of  $\Gamma$  would be lower. The second one is that it could be once again seen that  $U_{gradient}$  has the highest  $\Gamma$ , which indicates that this method does tend to produce the tightest estimate of  $f_0$ , or the lowest uncertainty of the  $f_0$  estimate. This is again caused by design, as the gradient based method uses the value of  $f_{0,guess}$  to be the center of the estimate, thus allowing for a lower uncertainty.

U <sub>Target</sub> (%)	Method	Γ <sub>Overall</sub>	Γ <sub>Case</sub>	Гр
	GCI <sub>R</sub>	76.29%	60.46%	57.04%
	GCIOR	76.74%	63.37%	59.08%
10%	GCI <sub>SW</sub>	78.56%	61.98%	59.67%
	GCI <sub>XS</sub>	78.94%	62.02%	59.47%
	Ugradient	84.76%	67.59%	66.82%
	GCI <sub>R</sub>	72.61%	56.29%	52.39%
	GCI <sub>OR</sub>	72.35%	57.72%	53.04%
5%	<b>GCI</b> <sub>SW</sub>	74.36%	57.68%	54.66%
	GCI <sub>XS</sub>	73.82%	56.54%	52.64%
	Ugradient	82.02%	64.42%	62.58%
	GCI <sub>R</sub>	58.82%	43.39%	37.85%
	GCI <sub>OR</sub>	50.85%	38.35%	21.35%
1%	GCI <sub>SW</sub>	55.46%	40.92%	34.47%
	GCI <sub>XS</sub>	53.18%	39.16%	22.70%
	Ugradient	67.35%	50.66%	44.11%

Table 3. Result of the Precision Test.

As for the time analysis, this result is shown in Table 4, in which it is showing the average time saving of each method compared  $to GCI_R$ . The positive sign in this data shows that the method does on average requires less real time to simulate a certain case, and a negative sign means that the method, on average requires more real time for its simulation. It should be known that once again, the data used for this analysis must be conservative.

Table 4. Average Computational Time Saving Compared to Roache's Method.

Method	$U_{Target} = 10\%$	$U_{Target} = 5\%$	$U_{Target} = 1\%$
GCIR	0%	0%	0%
GCI <sub>OR</sub>	-182.86%	-76.39%	-84.42%
GCI <sub>SW</sub>	-24.61%	-17.12%	-44.45%
GCI <sub>XS</sub>	-138.40%	-107.16%	-165.81%
Ugradient	26.23%	27.76%	14.83%

In Table 4, it could be seen that  $U_{gradient}$  is the only method that manage to reduce the required computational time, in which it is reduced by 15%–30%. It should be known that this is the averaged value. There are cases where the other methods also save time, and there are cases where  $U_{gradient}$  requires more time. Some example of the time required to simulate a case, and obtain an acceptable uncertainty is shown in Table 5.

Table 5. Several Examples of Required Simulation Time.

Method	Time (s)				
Wiethou	Case 10 ( <i>U<sub>Target</sub></i> = 10%)	Case 20 $(U_{Target} = 5\%)$	Case 24 ( <i>U<sub>Target</sub></i> = 1%)	Case 35 ( <i>U<sub>Target</sub></i> = 1%)	
GCI <sub>R</sub>	47.646	3594.817	5522.303	17,708.15	
GCIOR	51.846	3523.34	15333.32	30,509.99	
GCI <sub>SW</sub>	50.591	3523.34	15333.32	24,218.85	
GCI <sub>XS</sub>	50.591	6464.425	5564.398	50,600.98	
Ugradient	44.878	674.898	534.6054	26,107.87	

The general reason why  $U_{gradient}$  manages to save time is due to the possibility of the method to analyze data from a more coarse mesh, and does produce an accurate and precise estimate. This is a very big advantage, as the computational time would increase significantly as the mesh is gets denser. As such, even by just evading one more mesh refinement, there could be a significant time saving.

Observing the data, it seems that the saving would be much more significant if the overall data is on condition B. Some examples of the cases that fits this condition is shown in Figure 4, in which for case 24 itself, the time saving for this case is 50.15%, 50.15%, and 90.32%, respectively, for  $U_{Target} = 10\%$ , 5%, and 1%. On the other hand, if the data are in condition A, especially if the data are parabolic ( $p \approx 2$ ), the time saving would be less significant, even there might not be any time saving at all. The examples of the cases that fit this condition are shown in Figure 5, where the most extreme example is case 3 which has a time saving of -13.63%, -31.78%, and -35.65%, respectively, for  $U_{Target} = 10\%$ , 5%, and 1%. The cause of this difference is apparently due to precision issues.



Figure 4. The MMS simulation result that fits condition B: (a) Case 12; (b) Case 24; (c) Case 30.



Figure 5. The MMS simulation result that fits condition A: (a) Case 3; (b) Case 6; (c) Case 32.

Although previously it has been shown that the  $U_{gradient}$  gave the tightest estimate, or it is the more precise method, this is not true across all conditions. On a higher value of p, the other methods tend to be better than  $U_{gradient}$  if the accuracy issue is ignored. To prove this, the precision test would be conducted once again, but this time the dataset would consist of multiple random values of p,  $r_{12}$ ,  $r_{23}$ , and also  $f_3/f_1$ . The interval of p is set to be  $0 , while all the other variables is set to <math>1 < r_{12}$ ,  $r_{23}$ ,  $f_3/f_1 \le 2$ . A total of  $3 \times 10^7$  random data are taken and the  $\Gamma$  value is calculated once again. This result which is shown in Figure 6 shows that upon 1.8 < p, some other methods tend to be equal or even better than  $U_{gradient}$ . This is especially true for GCI<sub>R</sub>. A word of caution though, as these data do not take the accuracy into consideration. As seen in Figure 3, that especially for  $GCI_R$  with 2.2 < p, the failure is already very significant, that it could not be recommended for normal usage.

The effect of this result in the time analysis is that even if 2.2 < p, since there is a lot of combination on each case, there would be at least 1 triplet that is conservative. As such, these data are used for the time analysis. This would result in  $GCI_R$  to have the lowest uncertainty estimate, thus this method could utilize a more coarse mesh to obtain



an uncertainty equal or below  $U_{Target}$  and the overall time would be lower than all the other methods.

Figure 6. The result of a random data test: (a)  $U_{Target} = 10\%$ ; (b)  $U_{Target} = 5\%$ ; (c)  $U_{Target} = 1\%$ .

Considering Figures 3 and 6, for 1.8 < p, where there are several methods that perform better in terms of accuracy than  $U_{gradient}$  and do not have a low failure rate; there is a chance that these methods do have a lower computational time than  $U_{gradient}$ . This suggests that the usage of  $U_{gradient}$  to save time would be more significant if the data has a value of p < 1.8. This trend apparently occurs quite often in our dataset, especially when the meshes are coarse enough. However, upon refinement, there is a tendency that the data would behave as  $p \approx 2$  which have a minimal time saving. This switch in behavior is one of the cause that the time saving for  $U_{gradient}$  at  $U_{Target} = 1\%$  is lower compared to the rest. Some examples of the cases with switching behavior are shown in Figure 7, in which for case 35 itself, the time savings are 15.86%, 38%, and -47.43%, for, respectively,  $U_{Target} = 10\%$ , 5%, and 1%. All of this suggests that the usage of  $U_{gradient}$  is much better aimed at coarse meshes, with p < 1.8.



Figure 7. The MMS simulation result that has switching conditions: (a) Case 20; (b) Case 35.

Judging by the data from Table 4, in which the data on this table show the averaged time saving from all across the *p* spectrum, it could be inferred that for data with *p* < 1.8, then the time saving should be higher. As such, as engineers usually use a 5% uncertainty, then for a simulation data with *p* < 1.8, the time saving by using  $U_{gradien}$  compared to  $GCI_R$  is on average at least 25%. This is a significant amount of time saving and it could potentially be more. The maximum time saving that is observed in this dataset is up to 89.22%, which occurs in case 1 for  $U_{Target} = 5\%$ .

# 5. Study Case

To show the performance of the gradient based method on a real simulation, a study case would be presented. A simulation of an air flow passing through a 100 mm diameter cylinder is conducted. This flow is single phased, and have a uniform velocity distribution of 5 m/s, in which overall it correlates to a Reynolds number of 34,300. A total of 16 meshes are going to be used ranging from 1536 cells to 230,496 cells, of which the cell distribution is shown in Figure 8 and for each mesh the mesh is refined in a systematic manner. The simulation is solved in a steady-state manner using a standard k- $\varepsilon$  turbulence model with a near wall formulation is used for the cells near the wall. For the discretization, it is done by using a finite volume method with a second order accurate method is used to interpolate the face values. The simulation is then iterated until it reaches machine accuracy, in which then the drag coefficient ( $C_D$ ) could be obtained.



Figure 8. The cell distribution in the simulation of a flow passes through a cylinder.

The result of this simulation could be seen in Figure 9. First it could be seen that the data do behave monotonically and no oscillation is observed. This means that every triplet in this simulation could be calculated using the GCI and the gradient-based method. Although the accuracy test could not be done for these data since the analytical solution of this case is unknown, the precision test and the required computational time could still be done. Both of these test results are shown, respectively, in Tables 6 and 7.



**Figure 9.** The simulation result of a flow passes through a cylinder: (a)  $C_D$ ; (b) time.

Method	Γ			
Witthou	$U_{Target} = 10\%$	$U_{Target} = 5\%$	$U_{Target} = 1\%$	
GCI <sub>R</sub>	100.00%	98.35%	38.74%	
GCI <sub>OR</sub>	96.98%	70.88%	0.82%	
GCI <sub>SW</sub>	100.00%	95.05%	16.48%	
GCI <sub>XS</sub>	100.00%	93.68%	18.41%	
Ugradient	100.00%	100.00%	55.77%	

Table 6. The precision test result of the simulation of a flow passes through a cylinder.

Table 7. The required computational time of the simulation of a flow passes through a cylinder.

Method	Time (s)			
Wiethou	<i>U<sub>Target</sub></i> = 10%	$U_{Target} = 5\%$	$U_{Target} = 1\%$	
GCI <sub>R</sub>	765.964	1537.095	40,103.05	
GCI <sub>OR</sub>	1983.451	3161.402	45,013.19	
GCI <sub>SW</sub>	765.964	1993.79	41,529.36	
GCI <sub>XS</sub>	765.964	1993.79	75,110.02	
Ugradient	765.964	765.964	19,444.85	

The data on Tables 6 and 7 have the same tendencies as the result from the MMS simulations. The value of the  $\Gamma$  from  $U_{gradient}$  tends to be always the highest, which means that  $U_{gradient}$  is the most precise method. This precision also translates to the required computational time, in which for  $U_{Target} = 5\%$ , it saves 50.17% over  $GCI_R$  and for  $U_{Target} = 1\%$ , it saves 51.51%. There is no time saving in  $U_{Target} = 10\%$  since all the uncertainties from both  $GCI_R$  and  $U_{gradient}$  even from the coarsest mesh is already below 10%. This does show that the result on the MMS simulation could also apply into a real engineering case simulation.

Regarding the application of  $U_{gradient}$  to a real engineering case, it should be noted that within this example, this is a simple case where there is a single phase flow that passes through a simple geometry at a rather low Reynolds number of 34,300. This is much simpler case than any kind of real world application of CFD. At a higher complexity, or higher Reynolds number flow, it is expected that the data tend to oscillates more, thus making it incompatible with the GCI and  $U_{gradient}$ . This would require some mesh refinement, in which sometimes, to make sure the data do not oscillate, the required mesh is not practical to be created or even simulated. This does limit the application of both GCI and  $U_{gradient}$ . However, with the advantages that  $U_{gradient}$  have, the required mesh is only required to not oscillate any more. If the mesh does not oscillate any more, which means that  $U_{gradient}$  can be applied, it has been shown that  $U_{gradient}$  could produce an accurate and precise estimate even from coarse meshes. For comparison the  $GCI_R$  would require the mesh to not only stop oscillating but also be well behaved ( $p \approx p_f$ ) to ensure that the estimate is both accurate and precise.

#### 6. Conclusions

In this study, a new method to estimate a discretization error in CFD is presented. The method is based on the behavior of the gradient in a CFD simulation result. This method is tested on an MMS simulation dataset, in which the results show that even by changing the center of estimation from  $f_1$  to  $f_{0,guess}$ , the gradient-based method can still estimate the value of  $f_0$  even to a degree that is more accurate than the other method tested in this study. This gradient-based method is also shown to be more precise, especially for p < 1.8, which usually occurs on coarser meshes. This higher precision is a significant advantage for the gradient based method since it does allow the usage of a coarser mesh to obtain a certain level of uncertainty, thus reducing the overall computational time required for the simulation. For the range that this method is recommended for, which is p < 1.8, the time

saving is compared to the standard GCI with a targeted uncertainty of 5%, is on average at least 25%. This significant advantage would be useful for analyzing CFD simulation results.

The standard GCI is only practical for a low Reynolds number flow. However, with the advantage that the gradient-based method has, it might be practical up to a medium Reynolds number flow. It is not sure that this method would be suitable for a high Reynolds flow simulation or a more complex simulation, since the result of such simulation usually has a high tendency to oscillate which does break the assumptions of the gradient-based method.

Although it might not be perfect, nor it is applicable to every CFD simulation, the gradient-based method that has been shown is capable of performing better than the standard GCI. It is more accurate, more precise, and may reduce the overall computational time by a significant amount. As such, we do recommend the use of this method for analyzing the results of CFD.

**Author Contributions:** Conceptualization, A.S.; methodology, A.S.; software, A.S.; validation, A.S.; formal analysis, A.S.; writing—original draft preparation, A.S.; writing—review and editing, A.S.; visualization, A.S.; supervision, H.; project administration, H.; funding acquisition, A.S. and H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Universitas Indonesia in PUTI Q2 2020, grant number NKB-1712/UN2.RST/HKP05.00/2020.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## Abbreviations

- ASME The American Society of Mechanical Engineers
- AIAA American Institute of Aeronautics and Astronautics
- CFD Computational fluid dynamics
- GCI Grid convergence index
- GRE Generalized Richardson extrapolation

MMS Method of manufactured solutions

## Nomenclature

a.b	Constants	НОТ	Higher Order Terms
$C_D$	Drag Coefficient	u	Viscosity
$f_0^D$	Analytical result	p	Order of convergence
f 0.ouess	Guessed analytical result	$p_f$	Formal order of accuracy
fh	Simulation result of mesh with cell size <i>h</i>	P	Pressure
f1.2.3	Simulation result of mesh index 1,2, or 3	Ψ	Conservativeness
80	Gradient at cell size = $0$	r <sub>ab</sub>	Cell size ratio of mesh index <i>a</i> and <i>b</i>
ga	Gradient at mesh index <i>a</i>	SF	Safety Factor
Sab	Gradient of mesh index <i>a</i> and <i>b</i>	t	Time
GCI	Grid Convergence Index	Т	Temperature
Г	Probability of U or $GCI \leq U_{Target}$	τ	Shear Stress
ha	Cell size of mesh index <i>a</i>	<i>u,v,w</i>	Velocity
h <sub>12max</sub>	Maximum possible <i>h</i> value that has a gradient equal to $g_{12}$	Ŭ	Uncertainty
$h_{23min}$	Minimum possible <i>h</i> value that has a gradient equal to $g_{23}$	<i>U<sub>Target</sub></i>	Targeted uncertainty value

### References

- 1. Roy, C.J. Grid Convergence Error Analysis for Mixed-Order Numerical Schemes. AIAA J. 2003, 41, 595–604. [CrossRef]
- ASME Committee V&V 20. Standard for Verification and Validation on Computational Fluid Dynamics and Heat Transfer; The American Society of Mechanical Engineers: New York, NY, USA, 2009.

- 3. American Institute of Aeronautics and Astronautics (AIAA). *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations;* American Institute of Aeronautics and Astronautics: Reston, VA, USA, 2002.
- 4. Roache, P.J. Verification of Codes and Calculations. AIAA J. 1998, 36, 696–702. [CrossRef]
- 5. Rider, W.; Witkowski, W.; Kamm, J.R.; Wildey, T. Robust verification analysis. J. Comput. Phys. 2016, 307, 146–163. [CrossRef]
- 6. Roy C.J. Review of code and solution verification procedures for computational simulation. J. Comput. Phys. 2005, 205, 131–156. [CrossRef]
- Oberkampf, W.L.; Trucano, T.G. Verification and validation in computational fluid dynamics. *Prog. Aerosp. Sci.* 2002, 38, 209–272. [CrossRef]
- 8. Roache, P.J. Verification and Validation in Computational Science and Engineering, 1st ed.; Hermosa Publishers: Albuquerque, NM, USA, 1998.
- 9. Roy, C.J.; Oberkampf, W.L. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Comput. Methods Appl. Mech. Eng.* 2011, 200, 2131–2144. [CrossRef]
- 10. Roy, C.J. Review of Discretization Error Estimators in Scientific Computing. AIAA Pap. 2010, 2010–2126. [CrossRef]
- Roache, P.J. Perspective: A Method for Uniform Reporting of Grid Refinement Studies. *J. Fluids Eng.* 1994, 116, 405–413. [CrossRef]
  Celik, I.B.; Ghia, U.; Roache, P.J.; Freitas, C.J. Procedure for estimation and reporting of uncertainty due to discretization in CFD applications. *J. Fluids Eng.* 2008, 130:078001.
- 13. American Institute of Aeronautics and Astronautics (AIAA). Editorial Policy Statement on Numerical and Experimental Accuracy. *AIAA J.* 2013, 51. [CrossRef]
- 14. Richardson, L.F. The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam. *Phil. Trans. R. Soc. Lond. A* **1911**, *210*, 307–357.
- 15. Richardson, L.F. The Deferred Approach to the Limit. Part I. Single Lattice. Phil. Trans. R. Soc. Lond. A 1927, 226, 299–349.
- 16. Oberkampf, W.L.; Roy, C.J. Verification and Validation in Scientific Computing; Cambridge University Press: Cambridge, UK, 2010.
- 17. Eca, L.; Hoekstra, M. An Evaluation of Verification Procedures for CFD Applications. In Proceedings of the 24th Symposium on Naval Hydrodynamics, Fukuoka, Japan, 8–13 July 2002.
- Eca, L.; Vaz, G.; Hoekstra, M. A contribution for the Assessment of Discretization Error Estimators Based on Grid Refinement Studies. J. Verif. Valid. Uncertain. Quantif. 2018, 3, 021001. [CrossRef]
- Stern, F.; Wilson, R.V.; Coleman, H.W.; Paterson, E.G. Comprehensive Approach to Verification and Validation of CFD Simulations — Part 1: Methodology and Procedures. J. Fluids Eng. 2001, 123, 793–802. [CrossRef]
- 20. Cadafalch, J.; Segarra-Perez, C.D.; Consul, R.; Oliva, A. Verification of Finite Volume Computations on Steady-State Fluid Flow and Heat Transfer. J. Fluids Eng. 2002, 124, 11–21. [CrossRef]
- 21. Wilson, R.V.; Shao, J.; Stern, F. Discussion: Criticisms of the 'Correction Factor' Verification Method. J. Fluids Eng. 2004, 126, 704–706. [CrossRef]
- 22. Xing, T.; Stren., F. Factors of Safety for Richardson Extrapolation. J. Fluids Eng. 2010, 132, 061403. [CrossRef]
- 23. Eca, L.; Hoekstra, M. A procedure for the estimation of the numerical uncertainty of CFD calculations based on grid refinement studies. *J. Fluids Eng.* **2014**, *262*, 104–130.
- 24. Phillips, T.S.; Roy, C.J. A New Extrapolation-Based Uncertainty Estimator for Computational Fluid Dynamics. J. Fluids Eng. 2016, 1, 041006. [CrossRef]
- 25. Phillips, T.S.; Roy, C.J. Evaluation of Extrapolation-Based Discretization Error and Uncertainty Estimators. *AIAA Pap.* **2011**. [CrossRef]
- 26. Celik, I.B.; Parons, D.R., III. Prediction of discretization error using the error transport equation. J. Comput. Phys. 2017, 339, 96–125. [CrossRef]
- 27. Yan, G.K.; Ollivier-Gooch, C. On Efficiently Obtaining Higher Order Accurate Discretization Error Estimates for Unstructured Finite Volume Methods Using the Error Transport Equation. *J. Verif., Valid. Uncertain. Quantif.* **2017**, *2*, 041003. [CrossRef]
- 28. Wang, H.; Tyson, W.C.; Roy, C.J. Discretization Error Estimation for Discontinuous Galerkin Methods using Error Transport Equations. *AIAA Pap.* **2019**. [CrossRef]
- 29. Roache, P.J. Quantification of Uncertainty in Computational Fluid Dynamics. *Annu. Rev. Fluid. Mech.* **1997**, *29*, 123–160. [CrossRef]
- 30. Roache, P.J. Code Verification by the Method of Manufactured Solutions. J. Fluids Eng. 2002, 124, 4–10. [CrossRef]
- Eca, L.; Vas, G.; Toxopeus, S.L.; Hoekstra, M. Numerical Errors in Unsteady Flow Simulations. J. Verif. Valid. Uncertain. Quantif. 2019, 4, 021001. [CrossRef]