Statistic-based approach for highest precision numerical differentiation

Andrej Liptaj*

Abstract

If several independent algorithms for a computer-calculated quantity exist, then one can expect their results (which differ because of numerical errors) to follow approximately Gaussian distribution. The mean of this distribution, interpreted as the value of the quantity of interest, can be determined with much better precision than what is the precision provided by a single algorithm. Many practical algorithms introduce a bias using a parameter, e.g. a small but finite number to compute a limit or a large but finite number (cutoff) to approximate infinity. One may vary such parameter of a single algorithm, interpret the resulting numbers as generated by several algorithms and compute the average. A numerical evidence for the validity of this approach is, in the context of a fixed machine epsilon, shown for differentiation: the method greatly improves the precision and leads, presumably, to the most precise numerical differentiation nowadays known.

Keywords: numerical differentiation; precision; accuracy; averaging;

1 Introduction

Numerical differentiation (ND) can be addressed in different contexts. The context is usually linked to the reason why a symbolic approach is not used. One might want to differentiate a function f numerically because

1. one has only a (measured) set of data points $\{x_i, f(x_i)\}_{i=1}^{N}$,
2. f is highly-composed with large number of nested functions,
3. it is unknown how values of f are computed.

*Institute of Physics, Bratislava, Slovak Academy of Sciences, andrej.liptaj@savba.sk
I am willing to publish any of my ideas presented here in a journal, if someone (an editor) judges them interesting enough. Journals in the "*Current Contents*" database are strongly preferred.

The first scenario is not the one which is addressed in this text. The numerical precision of (measured) data points is usually much smaller then the precision provided by computer variables. Thus, the problem of differentiation reduces to search for an interpolation with appropriate properties (smoothness, data-noise filtering, etc..).

The second situation is also (usually) outside the scope of this text. Knowing the function formula, one can increase the precision of ND by using arbitrary-precision software or automatic differentiation [1]. Sometimes, however, the here-presented ideas may be applicable. This mostly refers to the situation where the above-mentioned approaches require a lot of work related to re-programming a function which is already coded in some standard (double precision) framework. The precision increase allowed by methods explained here may, under some circumstances, make the benefits of re-programming not worth the work.

The last scenario represents the framework which is relevant for this work - it is the most usual context in which the precision of the ND is addressed and discussed. The accuracy of the ND is here seen as an optimization task with contradictory aims: the size of discretization parameter should take neither large values (bias error), nor very small ones (rounding errors). A compromise leading to the best overall precision is to be found. One may refer to this situation using terms "fixed machine epsilon" or "black-box function". Most commonly such situation happens when one uses compiled external numerical libraries with fixed number format. It may also happen in the client-server environment, where the client asks for function values but has no access to the process in which server computes them. Practically it may also occur very often when one uses, for numerical purposes, his favorite programming language (with fixed number formats) and is unwilling to learn and implement new specific framework to increase the precision of the ND.

I would also like to state the context of this work with respect to the properties of the functions to be differentiated: In this text I focus on the (ill-conditioned) ND of a general differentiable function. I will therefore ignore special recipes suited for special situations, e.g. the set of analytic functions and well-conditioned differentiation based on the Cauchy theorem.

Once the context is clear, one can focus on the aim: reaching the ultimate accuracy in the procedure of ND. Different prescriptions for increasing precision proposed up to now (e.g. [2, 3, 4, 5, 6, 7, 8, 9]) are based on the evaluation of the function value at small number of points near to the point of differentiation and some smart ideas on how to reduce the numerical imprecision. However, all of them neglect the information which can be extracted from statistical considerations.

Here I present a novel statistics-based approach which allows for important error reduction. It uses averaging and may be applied to more situations then just to the ND.

In following section I present some general statistical arguments to support the presented method. Then the text fully focuses on the ND. To honestly study the subject I will use several prescriptions for ND. The corresponding issues will be reviewed in Sec. 3. Next, in Sec. 4, I will explain the heuristic testing method and present its results. In the last sections I will discuss different result-related observations and make summary and conclusions.

2 Statistical arguments

Average understood as summation (divided by a constant) is under very general assumptions subject to the central limit theorem. This can be used in numerical computations for precision increase. Indeed, if several independent algorithms for computing a quantity of interest exist, each of them having certain numerical imprecision, one may average the results and get a smaller error. Depending on circumstances, this procedure may be regarded as repeated unbiased independent measurements with random errors and, for this scenario, the expected shrinking of the error is

$$\sigma = \frac{\sqrt{\sum_{i=1}^{N} \sigma_i^2}}{N} \approx \frac{\sigma^{typical}}{\sqrt{N}}, \tag{1}$$

where one expects the numerical errors of the various methods not to be very different (all close to a typical value $\sigma^{typical}$).

The question of algorithm independence arises. Clearly, if each algorithm from a given set leads to the same result then the algorithms are, in the mathematical sense, fully correlated. However, for what concerns numerical errors, a different optics can be adopted: otherwise result-equivalent algorithms may differ a lot in the functions they use (and corresponding register operations) which de-correlates their numerical uncertainties. It is reasonable to assume that the numerical errors arise from technical details of the computer processing and do not actually depend a lot on the global "idea" of a given algorithm. Therefore one expects that algorithms which differ in "technical" sense provide practically uncorrelated numerical errors of their results.

Unfortunately, in practice, one usually does not have many independent methods to compute a given quantity. What is however often the case is a biased parameter-dependent algorithm. The parameters allow to approximate an ideal situation which is inaccessible via computers: a small (but finite) h can be used as a step in ND or integration, a large (but finite) Λ lambda can be used as a cutoff (approximating infinity). A natural idea arises: one may use different (but reasonable) values for these parameters, compute for each value the result and average the results[1]. Two issues can be addressed here: bias and error correlations.

Obviously, any ND (as an example) with nonzero step h is biased and even if infinite-precision computers were available the result would not be fully correct. The overall "wrongness" thus has two components: numerical errors (called also truncation or round-off errors) and bias induced by discretization error. Here

[1] In other words, one considers two (otherwise identical) methods which differ in parameter value as different methods

an expectation can be made: if the averaging helps to shrink the numerical errors then one should tend to use, in the averaging approach, an algorithm (its parameter values) with smaller bias[2]. In other words: the averaging cannot remove the bias but does remove numerical effects and so one expects to get the most precise results for less biased algorithm[3] compared to the bias leading to the most precise results for a single (i.e. non-averaged) algorithm.

Question of correlations should be understood in the context where methods with different parameters are seen as different estimators. It is a complex question: such estimators have different discretization error (and corresponding bias) and some behavior with respect to numerical errors. If one assumes that computations are done in regime where numerical errors are dominant, one can ask whether the estimators behave in a correlated way (for different points of differentiation and different functions in case of differentiation). If yes, the average of their estimates does not need to tend to zero (or to a reduced error at all), they also might be biased. The correlation of uncertainties of results from the parameter-changing approach is something one can examine empirically. The case of ND studied in this text shows that their mutual independence is large enough to provide substantial error reduction.

3 Numerical differentiation prescriptions

To make sure that the error shrinking by averaging is not limited to some specific prescription, I propose to test it on three different differentiation prescriptions (with an appropriately chosen h):

- Averaged finite difference (AFD)

$$f'_{AFD}(x,h) = \frac{1}{2}\left[\frac{f(x+h)-f(x)}{h} + \frac{f(x)-f(x-h)}{h}\right], \\ = \frac{f(x+h)-f(x-h)}{2h}.$$

- "Five-point rule" based on the Richardson extrapolation (RE, [2, 3])

$$f'_{RE}(x,h) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}.$$

  The implementation of the ND is in many common mathematical computer packages based on the Richardson extrapolation.

- Lanczos differentiation by integration (LDI, [4])

$$f'_{LDI}(x,h) = \frac{3}{2h^3}\int_{x-h}^{x+h} (x-t)\, f(t)\, dt.$$

[2] A typical value of a parameter is meant here. Of course, when averaging, parameter is changed in each evaluation (by construction of the method).

[3] A set of less biased algorithms.

To evaluate the integral I use, in my programs, the composite Boole's rule [10] with 16 equidistant points $\{x_i\}_{i=1}^{i=16}$, $x_1 = x - h$, $x_{16} = x + h$, $x_{i+1} - x_i = \triangle x$

$$\int_{x-h}^{x+h} f(t)\,dt \approx 2\,\triangle x\,\left(I_1 + I_2 + I_3 + I_4\right)/45,$$

$$I_1 = 7\left[f(x_1) + f(x_{16})\right],$$

$$I_2 = 32\sum_{i=1}^{i=7} f(x_{2i+1}),$$

$$I_3 = 14\sum_{i=1}^{i=3} f(x_{4i}),$$

$$I_4 = 12\sum_{i=0}^{i=3} f(x_{4i+2}).$$

Let me index these prescriptions by the letter k, $k\varepsilon\,\{\text{AFD, RE, LDI}\}$. I implement the averaging procedure in the straightforward way

$$f_k^{\prime AV}(x) = \frac{1}{N}\sum_{i=1}^{N} f_k^{\prime}(x, h_i),\ h_i\epsilon H,$$

where the set H is chosen in function of the h used in the single algorithm computation as follows:

- For AFD $H = [0.5h,\ 1.5h]$ where two options are investigated
  - h_i is generated as a random number with uniform distribution from the interval H (noted AFD_{MC}^{AV}).
  - successive values of h_i are generated such as to be equidistant with $h_1 = 0.5h$ and $h_N = 1.5h$ (noted AFD_{ED}^{AV}).
- For RE and LDI $H = [0.5h,\ 1.5h]$, where h_i is generated as a random number with uniform distribution from this interval (only this option is investigated).

Use of random numbers seems to be a safer option if aiming uncorrelated errors, yet regular division of the interval is tested also. For testing purposes I use a program[4] written in the *JAVA* programming language and double precision variables.

[4]The program can be, at least temporarily, downloaded from `http://www.dthph.sav.sk/fileadmin/user_upload/liptaj/differentiationAveraging.zip` or requested from the author. I also greatly profited from the WxMaxima software.

Case number	Function	$x=$	$\|f'(x)\| \approx$	$\|f''(x)\| \approx$
1	$L_7(x)$	9.683	19.88	0.0011
2	$L_7(x)$	11.2345	0.0031	28.57
3	$L_7(x)$	15.83	265.1	0.1534
4	$L_7(x)$	17.65	1.443	358.1
5	$L_7(x)$	15.8285	265.1	0.0026
6	$L_7(x)$	17.64595	0.0048	356.8
7	$\exp(x)$	-6.9	0.0010	0.0010
8	$\ln(x)$	10	0.1	0.01
9	$\arctan(x)$	6.245	0.0249	0.0078
10	$\cos(x)$	1.47	0.9949	0.1006
11	$\cos(x)$	0.1	0.0998	0.9950
12	$\cos(x)$	0.0025	0.0024	0.9999
13	$\arctan(x)$	0.002	0.9999	0.0039
14	$\ln(x)$	0.03	33.33	1111.1
15	$\exp(x)$	6.9	992.2	992.2
16	$\ln(x)$	1.0	1.0	1.0
17	$L_7(x)$	9.67477	19.88	0.1000
18	$L_7(x)$	11.2311	0.1001	28.49
19	$\exp(x)$	4.25	70.10	70.10

Table 1: Cases (points and functions) for which the averaging procedure was tested.

4 Testing and results

To study the behavior of the averaging method in more details I make an effort to examine it depending on the first and second derivatives of the function and on the step size[5] h. The first quantity directly correlates with what is being approximated (f'), the two others (f'', h) are often related to the expected precision of the approximation. I do the analysis by scanning 6 orders of magnitude for each "dependence" (its absolute value). For that purpose I choose 19 points in the $|f'|$, $|f''|$ plane, trying, in the logarithmic scale, to map it more or less uniformly. To avoid any fine-tuning suspicions I choose to use the basic elementary functions: $\cos(x)$, $\exp(x)$, $\ln(x)$ and $\arctan(x)$. However, with this choice, it is impossible to "uniformly" cover the $10^{-3} \lesssim |f'|, |f''| \lesssim 10^3$ region. Aiming this purpose, I add a suitable polynomial: the Laguerre polynomial $L_7(x)$. Situation is summarized in Tab 1 and in Fig. 1. From now on I will use the word "case" to refer to any of the 19 settings, each of them characterized by a function f, its argument x and the absolute value of its first and second derivatives at x. I will stick to the numbering presented in Tab. 1.

The step size h is changed from $h = 10^{-3}$ to $h = 10^{-8}$ in geometrical

[5]For averaging method the average step value is meant here, individuals step values go from $0.5h$ to $1.5h$.

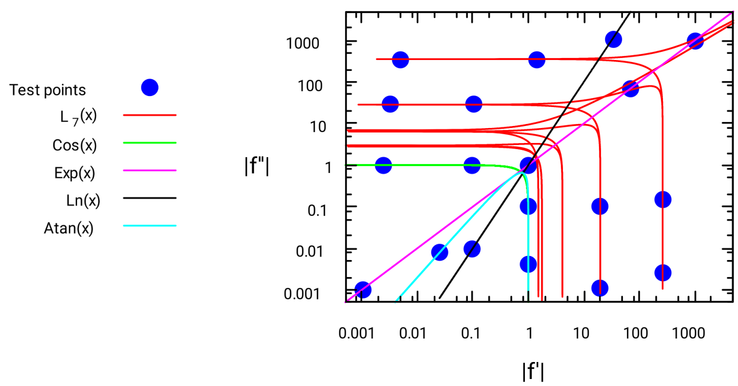

Figure 1: Studied cases depicted in $|f'|,|f''|$the plane.

progression with factor 10. One needs also to define the size of the statistical sample. To profit most from the averaging method a big number is suitable; I fix it to $N = 10^6$. This choice is driven also by practical considerations, i.e. the wish to keep the computer processing time in reasonable limits ($\sim$ minutes). The error is shown as absolute error

$$\left| f'_{\mathrm{approximated}} - f'_{\mathrm{true}} \right|, \tag{2}$$

where for f'_{true} the numerical value of the corresponding (known) derivative function is taken.

To prevent long listings within the main text, I put the tables with detailed results in Attachment. Each table corresponds to a single case and is differential in the step size and used prescription/method. Here, I average these tables (i.e. I average each cell over 19 cases), which might be somewhat artificial but has more message-conveying power.

Case-averaged results

h	10^{-3}	10^{-4}	10^{-5}
AFD	2.0×10^{-3}	2.0×10^{-5}	$\mathbf{1.8 \times 10^{-7}}$
AFD^{AV}_{MC}	2.2×10^{-3}	2.2×10^{-5}	2.2×10^{-7}
AFD^{AV}_{ED}	1.2×10^{-3}	1.2×10^{-5}	1.2×10^{-7}
RE	$\mathbf{6.9 \times 10^{-9}}$	7.6×10^{-9}	7.1×10^{-8}
RE^{AV}	1.1×10^{-8}	$\mathbf{6.0 \times 10^{-12}}$	4.2×10^{-11}
LDI	1.2×10^{-3}	$\mathbf{1.4 \times 10^{-5}}$	7.2×10^{-4}
LDI^{AV}	1.3×10^{-3}	1.3×10^{-5}	$\mathbf{4.4 \times 10^{-7}}$

Case-averaged results

h	10^{-6}	10^{-7}	10^{-8}
AFD	3.0×10^{-7}	1.8×10^{-6}	6.3×10^{-2}
AFD^{AV}_{MC}	$\mathbf{2.3 \times 10^{-9}}$	3.1×10^{-9}	5.1×10^{-5}
AFD^{AV}_{ED}	1.7×10^{-9}	$\mathbf{1.6 \times 10^{-9}}$	1.0×10^{-6}
RE	4.3×10^{-7}	2.4×10^{-6}	9.4×10^{-2}
RE^{AV}	7.1×10^{-10}	1.0×10^{-9}	4.0×10^{-5}
LDI	7.2×10^{-2}	5.5×10^{0}	1.2×10^{7}
LDI^{AV}	9.3×10^{-5}	5.1×10^{-3}	8.0×10^{3}

5 Discussion

Results confirm that the averaging method is very efficient in providing precise numerical derivative and reducing related errors. The overall error reduction (in absolute error) typically corresponds to two or three orders of magnitude[6] (when comparing the most precise results). Besides the obvious fact of reducing the error by increasing statistics[7], the assumptions concerning method functioning are further confirmed by the behavior with respect to h: as predicted earlier (Sec. 1) the most precise results of the averaging method typically happen for smaller step h than is h which corresponds to the most precise result of the same, but non-averaged prescription. Rather numerous are situations where h remains the same, rare are exceptions where the behavior is opposite (RE in case 12 and AFD^{AV}_{MC} in case 16). Rather amazing are results for RE^{AV} in cases 13 and 14 where, within the computer precision, exact results are reconstructed.

When comparing AFD^{AV}_{MC} and AFD^{AV}_{ED} approaches, one observes that their performances are rather equivalent. Yet, the "equidistant" method performs somewhat better which is little bit surprising: one can imagine that a regular division could introduce some correlation into the numbers to be averaged and thus slightly spoil the results. One can speculate that this behavior could be related to what is know from quasi-Monte Carlo methods: random numbers are often distributed quite unevenly, i.e. the "low-discrepancy" of the equidistant method may be the reason for it to win. It might certainly be a good idea for further studies to use, within the averaging method, low-discrepancy sequences.

The results also show that the averaging method can be combined with any of the three proposed prescriptions, which points once more to the general statistical aspects of the method. One may notice that the three prescriptions differ quite not only in the definition but also in the optimal step size h. The most precise of them is clearly the one based on the Richardson extrapolation.

[6]One may notice that this error reduction roughly corresponds to what is predicted by the formula (1). It gives additional hint in favor of the expected "modus operandi" of the averaging method.

[7]Non-averaged results can be seen as averaged results with statistics equal to one.

Finally, one needs to remark that for the *LDI* prescription in cases 14 and 16 the averaging method fails. I cannot think about a solid explanation, it might by a random accident or it might be somehow related to the *LDI* itself. At least in the case 14 the non-averaged result it atypically precise for this prescription, which might be interpreted as a "luck". In case 16 the difference between results is small, making the averaging failure not to be so "pronounced". In any case I want to stress that, despite these two observations, the averaging method works in general very well also for the *LDI* algorithm.

6 Summary and conclusion

In this text I made a numerical study of the averaging method applied to the ND. A rigorous approach to the whole idea would require a rigorous treatment of the floating-point arithmetic in computer registers. If possible, such an approach would certainly be very tedious with many assumptions and special cases. I believe the presented numerical evidence is strong enough to make claims about the method and its mechanism. The method is efficient and provides an important precision increase. It is very general and robust because of its statistical character. It should be used in situations where precision is the priority, its main drawback, slowness, makes it not suitable for quick computations. With lack of any more accurate approach to be found in literature, one can claim that the averaging method, when combined with a "standard" high precision prescription, is, in the context of fixed machine epsilon, the most precise numerical differentiation method at the market today.

References

[1] A. Griewank and A. Walther, "Introduction to automatic differentiation," *PAMM*, vol. 2, no. 1, pp. 45–49, 2003.

[2] L. F. Richardson, "The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 210, no. 459-470, pp. 307–357, 1911.

[3] L. F. Richardson and J. A. Gaunt, "The deferred approach to the limit. part i. single lattice. part ii. interpenetrating lattices," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 226, no. 636-646, pp. 299–361, 1927.

[4] C. Lanczos, *Applied Analysis.* Upper Saddle River, NJ 07458, USA: Prentice-Hall, 1956.

[5] J. N. Lyness and C. B. Moler, "Van der monde systems and numerical differentiation," *Numerische Mathematik*, vol. 8, pp. 458–464, Aug 1966.

[6] T. Ström and J. N. Lyness, "On numerical differentiation," *BIT Numerical Mathematics*, vol. 15, pp. 314–322, Sep 1975.

[7] B. Fornberg, "Generation of finite difference formulas on arbitrarily spaced grids," *Mathematics of computation*, vol. 51, no. 184, pp. 699–706, 1988.

[8] F. Feuillebois, "Precise calculation of numerical derivatives using lagrange and hermite interpolation," *Computers & Mathematics with Applications*, vol. 19, no. 5, pp. 1 – 11, 1990.

[9] E. Diekema and T. H. Koornwinder, "Differentiation by integration using orthogonal polynomials, a survey," *Journal of Approximation Theory*, vol. 164, no. 5, pp. 637 – 667, 2012.

[10] M. Abramowitz, *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables,*. Dover Publications, Incorporated, 1974.

Appendix

The following tables give detailed results for cases mentioned in Tab 1. In each table the step h is varied from 10^{-3} to 10^{-8} in columns, in rows different prescriptions/methods are presented (notation from Sec. 3 is used). Individual cells contain absolute error (formula 2), the most precise of them is, for each method, shown in bold characters.

Case number: 1

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	2.1×10^{-4}	2.1×10^{-6}	$\mathbf{4.9\times10^{-9}}$	2.1×10^{-7}	1.2×10^{-6}	9.5×10^{-5}
AFD_{MC}^{AV}	2.2×10^{-4}	2.2×10^{-6}	2.2×10^{-8}	$\mathbf{2.3\times10^{-10}}$	1.8×10^{-9}	6.4×10^{-8}
AFD_{ED}^{AV}	1.2×10^{-4}	1.2×10^{-6}	1.2×10^{-8}	$\mathbf{3.9\times10^{-10}}$	5.2×10^{-10}	2.1×10^{-9}
RE	$\mathbf{1.5\times10^{-10}}$	4.4×10^{-10}	2.3×10^{-8}	2.9×10^{-7}	1.9×10^{-6}	1.5×10^{-4}
RE^{AV}	5.0×10^{-11}	$\mathbf{1.7\times10^{-12}}$	1.2×10^{-11}	2.9×10^{-10}	7.8×10^{-11}	9.4×10^{-8}
LDI	1.2×10^{-4}	$\mathbf{8.4\times10^{-7}}$	2.6×10^{-5}	2.6×10^{-3}	1.3×10^{-1}	2.6×10^{4}
LDI^{AV}	1.3×10^{-4}	1.3×10^{-6}	$\mathbf{2.8\times10^{-8}}$	8.4×10^{-6}	5.5×10^{-4}	1.1×10^{1}

Case number: 2

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	3.9×10^{-4}	3.9×10^{-6}	$\mathbf{1.7\times10^{-8}}$	3.4×10^{-7}	1.8×10^{-6}	4.8×10^{-4}
AFD_{MC}^{AV}	4.2×10^{-4}	4.2×10^{-6}	4.2×10^{-8}	$\mathbf{4.5\times10^{-10}}$	2.1×10^{-9}	4.7×10^{-7}
AFD_{ED}^{AV}	2.3×10^{-4}	2.3×10^{-6}	2.3×10^{-8}	$\mathbf{4.0\times10^{-10}}$	6.0×10^{-10}	1.8×10^{-9}
RE	2.2×10^{-9}	$\mathbf{4.4\times10^{-10}}$	3.0×10^{-8}	4.1×10^{-7}	2.2×10^{-6}	7.2×10^{-4}
RE^{AV}	2.8×10^{-9}	$\mathbf{7.0\times10^{-12}}$	9.5×10^{-12}	4.8×10^{-10}	2.8×10^{-9}	2.7×10^{-7}
LDI	2.3×10^{-4}	$\mathbf{1.6\times10^{-6}}$	4.9×10^{-5}	4.9×10^{-3}	2.4×10^{-1}	1.2×10^{5}
LDI^{AV}	2.5×10^{-4}	2.5×10^{-6}	$\mathbf{1.4\times10^{-7}}$	5.5×10^{-6}	4.9×10^{-4}	2.6×10^{0}

Case number: 3

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	1.7×10^{-3}	1.7×10^{-5}	$\mathbf{9.6 \times 10^{-8}}$	2.5×10^{-7}	1.1×10^{-6}	4.6×10^{-2}
AFD_{MC}^{AV}	1.8×10^{-3}	1.8×10^{-5}	1.8×10^{-7}	$\mathbf{1.3 \times 10^{-9}}$	2.3×10^{-8}	2.2×10^{-5}
AFD_{ED}^{AV}	9.8×10^{-4}	9.8×10^{-6}	9.8×10^{-8}	$\mathbf{5.5 \times 10^{-9}}$	1.4×10^{-8}	7.4×10^{-7}
RE	1.1×10^{-8}	$\mathbf{6.3 \times 10^{-9}}$	1.0×10^{-7}	4.8×10^{-7}	5.1×10^{-7}	6.9×10^{-2}
RE^{AV}	1.8×10^{-8}	$\mathbf{2.0 \times 10^{-11}}$	1.2×10^{-10}	4.1×10^{-9}	1.9×10^{-9}	6.4×10^{-5}
LDI	1.0×10^{-3}	$\mathbf{1.1 \times 10^{-6}}$	6.0×10^{-4}	6.0×10^{-2}	3.0×10^{0}	1.2×10^{7}
LDI^{AV}	1.1×10^{-3}	1.1×10^{-5}	$\mathbf{6.2 \times 10^{-7}}$	4.6×10^{-5}	6.5×10^{-3}	1.0×10^{4}

Case number: 4

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	5.3×10^{-3}	5.3×10^{-5}	$\mathbf{1.1 \times 10^{-7}}$	2.6×10^{-6}	1.9×10^{-5}	5.4×10^{-1}
AFD_{MC}^{AV}	5.7×10^{-3}	5.7×10^{-5}	5.7×10^{-7}	$\mathbf{4.9 \times 10^{-9}}$	1.7×10^{-8}	6.3×10^{-4}
AFD_{ED}^{AV}	3.1×10^{-3}	3.1×10^{-5}	3.1×10^{-7}	$\mathbf{2.3 \times 10^{-10}}$	9.7×10^{-10}	9.0×10^{-6}
RE	$\mathbf{2.1 \times 10^{-8}}$	8.7×10^{-8}	6.2×10^{-7}	3.8×10^{-6}	2.7×10^{-5}	8.0×10^{-1}
RE^{AV}	2.7×10^{-8}	$\mathbf{1.8 \times 10^{-11}}$	3.4×10^{-10}	1.7×10^{-9}	4.0×10^{-9}	5.7×10^{-5}
LDI	3.2×10^{-3}	$\mathbf{2.2 \times 10^{-5}}$	9.6×10^{-4}	9.6×10^{-2}	9.6×10^{0}	9.9×10^{7}
LDI^{AV}	3.4×10^{-3}	3.4×10^{-5}	$\mathbf{1.7 \times 10^{-6}}$	7.1×10^{-4}	1.9×10^{-2}	6.5×10^{4}

Case number: 5

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	1.7×10^{-3}	1.7×10^{-5}	$\mathbf{5.2 \times 10^{-8}}$	2.1×10^{-7}	5.8×10^{-6}	4.7×10^{-2}
AFD_{MC}^{AV}	1.8×10^{-3}	1.8×10^{-5}	1.8×10^{-7}	$\mathbf{2.2 \times 10^{-9}}$	6.7×10^{-9}	6.0×10^{-5}
AFD_{ED}^{AV}	9.8×10^{-4}	9.8×10^{-6}	9.8×10^{-8}	$\mathbf{2.4 \times 10^{-9}}$	5.7×10^{-9}	1.4×10^{-7}
RE	9.2×10^{-9}	$\mathbf{7.9 \times 10^{-9}}$	1.6×10^{-7}	4.5×10^{-7}	5.9×10^{-6}	6.9×10^{-2}
RE^{AV}	1.8×10^{-8}	$\mathbf{1.9 \times 10^{-11}}$	3.3×10^{-11}	1.3×10^{-9}	4.3×10^{-9}	3.1×10^{-5}
LDI	1.0×10^{-3}	$\mathbf{1.1 \times 10^{-6}}$	6.0×10^{-4}	6.0×10^{-2}	3.0×10^{0}	1.2×10^{7}
LDI^{AV}	1.1×10^{-3}	1.1×10^{-5}	$\mathbf{2.0 \times 10^{-7}}$	3.5×10^{-5}	2.2×10^{-3}	1.0×10^{4}

Case number: 6

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	5.3×10^{-3}	5.3×10^{-5}	$\mathbf{7.7 \times 10^{-7}}$	1.5×10^{-6}	1.6×10^{-6}	5.6×10^{-1}
AFD_{MC}^{AV}	5.7×10^{-3}	5.7×10^{-5}	5.7×10^{-7}	7.4×10^{-9}	$\mathbf{2.2 \times 10^{-9}}$	2.7×10^{-4}
AFD_{ED}^{AV}	3.1×10^{-3}	3.1×10^{-5}	3.1×10^{-7}	8.9×10^{-9}	$\mathbf{5.7 \times 10^{-9}}$	8.9×10^{-6}
RE	$\mathbf{1.7 \times 10^{-8}}$	3.3×10^{-8}	3.3×10^{-7}	1.9×10^{-6}	2.0×10^{-6}	8.4×10^{-1}
RE^{AV}	2.7×10^{-8}	$\mathbf{3.0 \times 10^{-11}}$	2.1×10^{-10}	4.8×10^{-9}	1.2×10^{-9}	6.0×10^{-4}
LDI	3.2×10^{-3}	$\mathbf{2.2 \times 10^{-5}}$	9.6×10^{-4}	9.6×10^{-2}	9.6×10^{0}	9.8×10^{7}
LDI^{AV}	3.4×10^{-3}	3.4×10^{-5}	$\mathbf{2.0 \times 10^{-6}}$	2.8×10^{-4}	6.2×10^{-2}	6.7×10^{4}

Case number: 7

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	2.1×10^{-2}	2.1×10^{-4}	2.1×10^{-6}	2.5×10^{-7}	1.1×10^{-7}	$\mathbf{3.7 \times 10^{-12}}$
AFD^{AV}_{MC}	2.3×10^{-2}	2.3×10^{-4}	2.3×10^{-6}	2.2×10^{-8}	6.6×10^{-10}	$\mathbf{5.0 \times 10^{-15}}$
AFD^{AV}_{ED}	1.2×10^{-2}	1.2×10^{-4}	1.2×10^{-6}	1.2×10^{-8}	9.6×10^{-10}	$\mathbf{6.2 \times 10^{-17}}$
RE	3.1×10^{-8}	4.0×10^{-9}	2.4×10^{-8}	4.0×10^{-7}	8.0×10^{-7}	$\mathbf{5.6 \times 10^{-12}}$
RE^{AV}	4.7×10^{-8}	2.7×10^{-12}	1.5×10^{-11}	2.8×10^{-10}	1.8×10^{-9}	$\mathbf{4.2 \times 10^{-15}}$
LDI	1.3×10^{-2}	$\mathbf{2.0 \times 10^{-4}}$	1.0×10^{-2}	1.0×10^{0}	7.8×10^{1}	2.7×10^{-4}
LDI^{AV}	1.4×10^{-2}	1.4×10^{-4}	3.5×10^{-6}	6.5×10^{-4}	4.0×10^{-3}	$\mathbf{3.3 \times 10^{-7}}$

Case number: 8

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	2.5×10^{-4}	2.5×10^{-6}	$\mathbf{4.8 \times 10^{-8}}$	2.2×10^{-7}	7.6×10^{-8}	1.3×10^{-4}
AFD^{AV}_{MC}	2.7×10^{-4}	2.7×10^{-6}	2.7×10^{-8}	$\mathbf{2.4 \times 10^{-10}}$	1.3×10^{-9}	7.7×10^{-8}
AFD^{AV}_{ED}	1.5×10^{-4}	1.5×10^{-6}	1.5×10^{-8}	$\mathbf{7.2 \times 10^{-11}}$	8.6×10^{-11}	3.7×10^{-10}
RE	$\mathbf{2.7 \times 10^{-10}}$	5.1×10^{-10}	2.7×10^{-8}	3.1×10^{-7}	1.6×10^{-7}	1.9×10^{-4}
RE^{AV}	5.0×10^{-10}	$\mathbf{2.0 \times 10^{-12}}$	1.2×10^{-11}	2.6×10^{-11}	4.2×10^{-10}	1.4×10^{-7}
LDI	1.5×10^{-4}	$\mathbf{1.0 \times 10^{-6}}$	3.3×10^{-5}	3.3×10^{-3}	1.6×10^{-1}	3.5×10^{4}
LDI^{AV}	1.6×10^{-4}	1.6×10^{-6}	$\mathbf{6.9 \times 10^{-8}}$	8.7×10^{-6}	7.1×10^{-4}	1.7×10^{1}

Case number: 9

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	8.6×10^{-5}	8.6×10^{-7}	8.5×10^{-9}	$\mathbf{3.0 \times 10^{-9}}$	2.7×10^{-8}	1.3×10^{-6}
AFD^{AV}_{MC}	9.3×10^{-5}	9.3×10^{-7}	9.3×10^{-9}	$\mathbf{7.0 \times 10^{-11}}$	1.1×10^{-10}	3.8×10^{-10}
AFD^{AV}_{ED}	5.0×10^{-5}	5.0×10^{-7}	5.0×10^{-9}	$\mathbf{3.3 \times 10^{-11}}$	4.3×10^{-11}	1.3×10^{-10}
RE	1.1×10^{-9}	$\mathbf{7.6 \times 10^{-11}}$	2.0×10^{-10}	6.2×10^{-9}	4.2×10^{-8}	1.7×10^{-6}
RE^{AV}	1.6×10^{-9}	1.9×10^{-13}	$\mathbf{1.9 \times 10^{-13}}$	2.2×10^{-11}	1.7×10^{-10}	1.3×10^{-9}
LDI	5.2×10^{-5}	$\mathbf{5.6 \times 10^{-7}}$	5.5×10^{-6}	5.5×10^{-4}	4.1×10^{-2}	1.4×10^{2}
LDI^{AV}	5.6×10^{-5}	5.6×10^{-7}	$\mathbf{7.1 \times 10^{-9}}$	2.8×10^{-7}	1.3×10^{-6}	6.7×10^{-1}

Case number: 10

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	4.5×10^{-5}	4.5×10^{-7}	4.5×10^{-9}	$\mathbf{1.1 \times 10^{-10}}$	6.7×10^{-10}	1.4×10^{-10}
AFD^{AV}_{MC}	4.9×10^{-5}	4.9×10^{-7}	4.9×10^{-9}	4.9×10^{-11}	$\mathbf{2.6 \times 10^{-12}}$	6.1×10^{-12}
AFD^{AV}_{ED}	2.7×10^{-5}	2.7×10^{-7}	2.7×10^{-9}	2.7×10^{-11}	$\mathbf{1.8 \times 10^{-13}}$	2.9×10^{-13}
RE	3.9×10^{-9}	$\mathbf{4.0 \times 10^{-12}}$	4.1×10^{-12}	2.1×10^{-10}	1.4×10^{-9}	6.0×10^{-10}
RE^{AV}	5.9×10^{-9}	5.8×10^{-13}	$\mathbf{1.7 \times 10^{-14}}$	2.5×10^{-13}	2.0×10^{-12}	1.2×10^{-12}
LDI	2.7×10^{-5}	2.7×10^{-7}	$\mathbf{4.1 \times 10^{-8}}$	7.6×10^{-6}	1.1×10^{-3}	8.7×10^{-1}
LDI^{AV}	3.0×10^{-5}	3.0×10^{-7}	$\mathbf{2.7 \times 10^{-9}}$	2.6×10^{-8}	1.7×10^{-6}	1.0×10^{-3}

Case number: 11

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	5.3×10^{-4}	5.3×10^{-6}	5.3×10^{-8}	5.2×10^{-10}	6.6×10^{-11}	$\mathbf{3.9 \times 10^{-11}}$
AFD_{MC}^{AV}	5.7×10^{-4}	5.7×10^{-6}	5.7×10^{-8}	5.7×10^{-10}	5.7×10^{-12}	$\mathbf{3.6 \times 10^{-13}}$
AFD_{ED}^{AV}	3.1×10^{-4}	3.1×10^{-6}	3.1×10^{-8}	3.1×10^{-10}	3.1×10^{-12}	$\mathbf{5.1 \times 10^{-15}}$
RE	6.8×10^{-9}	7.0×10^{-13}	$\mathbf{5.8 \times 10^{-13}}$	2.2×10^{-12}	6.6×10^{-11}	4.1×10^{-10}
RE^{AV}	1.0×10^{-8}	1.0×10^{-12}	$\mathbf{8.9 \times 10^{-16}}$	3.6×10^{-15}	4.5×10^{-14}	1.4×10^{-13}
LDI	3.2×10^{-4}	3.2×10^{-6}	$\mathbf{3.3 \times 10^{-8}}$	5.0×10^{-7}	3.3×10^{-5}	1.8×10^{-2}
LDI^{AV}	3.4×10^{-4}	3.4×10^{-6}	3.4×10^{-8}	$\mathbf{2.2 \times 10^{-10}}$	3.7×10^{-8}	3.6×10^{-6}

Case number: 12

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	5.8×10^{-4}	5.8×10^{-6}	5.8×10^{-8}	5.8×10^{-10}	$\mathbf{7.8 \times 10^{-12}}$	4.3×10^{-10}
AFD_{MC}^{AV}	6.3×10^{-4}	6.3×10^{-6}	6.3×10^{-8}	6.3×10^{-10}	6.3×10^{-12}	$\mathbf{2.8 \times 10^{-13}}$
AFD_{ED}^{AV}	3.4×10^{-4}	3.4×10^{-6}	3.4×10^{-8}	3.4×10^{-10}	3.4×10^{-12}	$\mathbf{8.0 \times 10^{-14}}$
RE	7.0×10^{-9}	7.6×10^{-13}	5.4×10^{-13}	$\mathbf{4.8 \times 10^{-13}}$	4.5×10^{-11}	6.1×10^{-10}
RE^{AV}	1.1×10^{-8}	1.1×10^{-12}	$\mathbf{1.8 \times 10^{-15}}$	5.3×10^{-15}	9.8×10^{-15}	7.0×10^{-13}
LDI	3.5×10^{-4}	3.5×10^{-6}	3.5×10^{-8}	$\mathbf{1.3 \times 10^{-8}}$	2.6×10^{-6}	1.3×10^{-4}
LDI^{AV}	3.8×10^{-4}	3.8×10^{-6}	3.8×10^{-8}	$\mathbf{4.3 \times 10^{-10}}$	4.4×10^{-9}	3.8×10^{-7}

Case number: 13

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	5.8×10^{-4}	5.8×10^{-6}	5.8×10^{-8}	5.8×10^{-10}	8.6×10^{-11}	$\mathbf{3.7 \times 10^{-11}}$
AFD_{MC}^{AV}	6.3×10^{-4}	6.3×10^{-6}	6.3×10^{-8}	6.3×10^{-10}	6.4×10^{-12}	$\mathbf{7.6 \times 10^{-13}}$
AFD_{ED}^{AV}	3.4×10^{-4}	3.4×10^{-6}	3.4×10^{-8}	3.4×10^{-10}	3.4×10^{-12}	$\mathbf{4.3 \times 10^{-14}}$
RE	7.0×10^{-9}	7.1×10^{-13}	$\mathbf{2.7 \times 10^{-13}}$	8.4×10^{-13}	7.7×10^{-11}	2.2×10^{-10}
RE^{AV}	1.1×10^{-8}	1.1×10^{-12}	$\mathbf{0.0 \times 10^{0}}$	1.3×10^{-14}	3.8×10^{-14}	7.7×10^{-14}
LDI	3.5×10^{-4}	3.5×10^{-6}	3.5×10^{-8}	$\mathbf{1.3 \times 10^{-8}}$	2.6×10^{-6}	1.3×10^{-4}
LDI^{AV}	3.8×10^{-4}	3.8×10^{-6}	3.8×10^{-8}	$\mathbf{3.9 \times 10^{-10}}$	2.1×10^{-9}	7.2×10^{-7}

Case number: 14

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	5.7×10^{-4}	5.7×10^{-6}	5.7×10^{-8}	5.6×10^{-10}	$\mathbf{1.9 \times 10^{-11}}$	3.0×10^{-11}
AFD_{MC}^{AV}	6.1×10^{-4}	6.1×10^{-6}	6.1×10^{-8}	6.1×10^{-10}	6.2×10^{-12}	$\mathbf{5.1 \times 10^{-13}}$
AFD_{ED}^{AV}	3.3×10^{-4}	3.3×10^{-6}	3.3×10^{-8}	3.3×10^{-10}	3.3×10^{-12}	$\mathbf{4.9 \times 10^{-14}}$
RE	6.9×10^{-9}	6.3×10^{-13}	$\mathbf{2.5 \times 10^{-14}}$	3.5×10^{-12}	1.0×10^{-11}	3.0×10^{-11}
RE^{AV}	1.0×10^{-8}	1.0×10^{-12}	$\mathbf{0.0 \times 10^{0}}$	2.7×10^{-15}	2.7×10^{-14}	1.2×10^{-12}
LDI	3.4×10^{-4}	3.4×10^{-6}	3.6×10^{-8}	2.5×10^{-7}	$\mathbf{2.0 \times 10^{-11}}$	1.1×10^{-3}
LDI^{AV}	3.7×10^{-4}	3.7×10^{-6}	3.7×10^{-8}	$\mathbf{2.1 \times 10^{-10}}$	6.2×10^{-9}	4.8×10^{-6}

Case number: 15

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	7.7×10^{-5}	7.7×10^{-7}	4.9×10^{-9}	$\mathbf{3.1 \times 10^{-9}}$	1.0×10^{-7}	2.6×10^{-6}
AFD_{MC}^{AV}	8.3×10^{-5}	8.3×10^{-7}	8.3×10^{-9}	$\mathbf{4.7 \times 10^{-11}}$	7.6×10^{-11}	3.6×10^{-9}
AFD_{ED}^{AV}	4.5×10^{-5}	4.5×10^{-7}	4.5×10^{-9}	6.5×10^{-11}	2.1×10^{-10}	$\mathbf{6.5 \times 10^{-11}}$
RE	1.1×10^{-9}	$\mathbf{2.2 \times 10^{-10}}$	4.1×10^{-9}	4.5×10^{-9}	9.9×10^{-8}	3.9×10^{-6}
RE^{AV}	1.8×10^{-9}	$\mathbf{3.8 \times 10^{-13}}$	1.6×10^{-12}	9.6×10^{-12}	4.8×10^{-10}	1.4×10^{-9}
LDI	4.6×10^{-5}	$\mathbf{5.1 \times 10^{-7}}$	6.7×10^{-6}	6.7×10^{-4}	5.1×10^{-2}	2.6×10^{2}
LDI^{AV}	5.0×10^{-5}	5.0×10^{-7}	$\mathbf{8.6 \times 10^{-10}}$	3.7×10^{-7}	2.8×10^{-5}	1.1×10^{0}

Case number: 16

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	1.6×10^{-4}	1.6×10^{-6}	1.6×10^{-8}	1.3×10^{-10}	1.2×10^{-10}	$\mathbf{5.9 \times 10^{-11}}$
AFD_{MC}^{AV}	1.7×10^{-4}	1.7×10^{-6}	1.7×10^{-8}	1.7×10^{-10}	$\mathbf{6.7 \times 10^{-13}}$	1.4×10^{-12}
AFD_{ED}^{AV}	9.1×10^{-5}	9.1×10^{-7}	9.1×10^{-9}	9.1×10^{-11}	1.1×10^{-12}	$\mathbf{2.4 \times 10^{-13}}$
RE	4.8×10^{-9}	$\mathbf{1.1 \times 10^{-12}}$	5.8×10^{-12}	3.8×10^{-11}	3.0×10^{-10}	1.2×10^{-9}
RE^{AV}	7.3×10^{-9}	7.3×10^{-13}	$\mathbf{1.8 \times 10^{-14}}$	1.2×10^{-13}	2.8×10^{-13}	1.1×10^{-13}
LDI	9.4×10^{-5}	9.4×10^{-7}	$\mathbf{6.7 \times 10^{-9}}$	3.4×10^{-7}	2.0×10^{-5}	4.4×10^{-2}
LDI^{AV}	1.0×10^{-4}	1.0×10^{-6}	$\mathbf{9.9 \times 10^{-9}}$	2.3×10^{-8}	2.4×10^{-6}	1.6×10^{-2}

Case number: 17

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	2.0×10^{-4}	2.0×10^{-6}	2.7×10^{-8}	$\mathbf{1.8 \times 10^{-8}}$	2.0×10^{-6}	1.0×10^{-4}
AFD_{MC}^{AV}	2.2×10^{-4}	2.2×10^{-6}	2.2×10^{-8}	$\mathbf{2.4 \times 10^{-10}}$	1.2×10^{-9}	6.9×10^{-10}
AFD_{ED}^{AV}	1.2×10^{-4}	1.2×10^{-6}	1.2×10^{-8}	$\mathbf{1.3 \times 10^{-10}}$	2.5×10^{-10}	5.1×10^{-7}
RE	$\mathbf{1.8 \times 10^{-10}}$	7.8×10^{-10}	1.0×10^{-8}	3.0×10^{-8}	2.2×10^{-6}	1.4×10^{-4}
RE^{AV}	3.9×10^{-11}	$\mathbf{1.3 \times 10^{-12}}$	2.2×10^{-11}	2.8×10^{-10}	2.2×10^{-9}	6.7×10^{-8}
LDI	1.2×10^{-4}	$\mathbf{8.3 \times 10^{-7}}$	2.6×10^{-5}	2.6×10^{-3}	1.3×10^{-1}	2.5×10^{4}
LDI^{AV}	1.3×10^{-4}	1.3×10^{-6}	$\mathbf{1.7 \times 10^{-8}}$	2.7×10^{-7}	1.2×10^{-4}	3.6×10^{1}

Case number: 18

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	3.9×10^{-4}	3.9×10^{-6}	2.9×10^{-8}	$\mathbf{2.6 \times 10^{-8}}$	1.4×10^{-6}	4.8×10^{-4}
AFD_{MC}^{AV}	4.2×10^{-4}	4.2×10^{-6}	4.2×10^{-8}	$\mathbf{8.5 \times 10^{-10}}$	2.5×10^{-9}	7.0×10^{-7}
AFD_{ED}^{AV}	2.3×10^{-4}	2.3×10^{-6}	2.3×10^{-8}	$\mathbf{1.6 \times 10^{-10}}$	2.4×10^{-9}	9.8×10^{-9}
RE	$\mathbf{1.8 \times 10^{-9}}$	3.2×10^{-9}	1.7×10^{-8}	6.3×10^{-8}	1.9×10^{-6}	6.8×10^{-4}
RE^{AV}	2.7×10^{-9}	$\mathbf{7.1 \times 10^{-12}}$	3.4×10^{-11}	1.4×10^{-10}	8.6×10^{-11}	1.1×10^{-6}
LDI	2.3×10^{-4}	$\mathbf{1.6 \times 10^{-6}}$	4.9×10^{-5}	4.9×10^{-3}	2.4×10^{-1}	1.2×10^{5}
LDI^{AV}	2.5×10^{-4}	2.5×10^{-6}	$\mathbf{3.7 \times 10^{-8}}$	1.6×10^{-5}	3.0×10^{-5}	2.1×10^{2}

Case number: 19

h	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
AFD	1.0×10^{-6}	1.0×10^{-8}	$\mathbf{7.6 \times 10^{-10}}$	2.7×10^{-9}	1.6×10^{-8}	1.9×10^{-7}
AFD_{MC}^{AV}	1.1×10^{-6}	1.1×10^{-8}	1.1×10^{-10}	$\mathbf{1.5 \times 10^{-12}}$	5.7×10^{-11}	2.2×10^{-10}
AFD_{ED}^{AV}	5.8×10^{-7}	5.8×10^{-9}	5.8×10^{-11}	3.8×10^{-12}	2.9×10^{-11}	$\mathbf{3.0 \times 10^{-12}}$
RE	9.0×10^{-11}	$\mathbf{6.3 \times 10^{-11}}$	8.2×10^{-10}	3.6×10^{-9}	1.8×10^{-8}	2.8×10^{-7}
RE^{AV}	1.4×10^{-10}	$\mathbf{4.3 \times 10^{-14}}$	8.0×10^{-13}	1.3×10^{-12}	4.2×10^{-11}	2.2×10^{-10}
LDI	6.0×10^{-7}	$\mathbf{1.7 \times 10^{-8}}$	1.5×10^{-6}	1.5×10^{-4}	1.1×10^{-2}	1.9×10^{1}
LDI^{AV}	6.5×10^{-7}	6.5×10^{-9}	$\mathbf{4.1 \times 10^{-10}}$	4.1×10^{-8}	1.3×10^{-5}	8.0×10^{-2}