Title: Beyond Moore's Law Uncertainty

Author(s): Foster, Robert Christian
Gattiker, James R.
Weaver, Brian Phillip
Picard, Richard Roy

Intended for: SIAM/ASA Journal on Uncertainty Quantification

Issued: 2018-09-10 (Draft)

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Beyond Moore's Law Uncertainty

Robert C. Foster, James Gattiker, Rick Picard, Brian Weaver

Los Alamos National Laboratory

Abstract

The future of computing will involve machines and algorithms with uncertainty built into the computational process in order to reach the desired level of performance. Though many may feel uncomfortable with this added uncertainty, this uncertainty often either derives from or mirrors sources of uncertainty in traditional computation that are commonly ignored or remain unaccounted for in the analysis. In this paper, several traditional sources of uncertainty are compared to random noise built into the computation process. Two specific examples - the chaotic Duffing equation and the shallow water equations - are analyzed for path-following uncertainty quantification and aggregate quantity uncertainty quantification. Comparisons are drawn between the traditional sources of uncertainty and random noise that could be introduced by an approximate computation process.

1 Beyond Moore's Law and Approximate Computing

In 1965, Gordon Moore predicted that the technological increases would enable exponentially packing more components onto integrated circuits (Moore (1965)). This would lead to increases in power that have been estimated as doubling roughly every 18-24 months. Moore's law has consistently defied predictions of its demise, as technological breakthroughs have allowed for increasingly dense circuits that were previously unavailable given current technology (Cavin et al. (2012)). However, new advances in technology only delay, not endlessly defer, the end of Moore's law, as problems of power consumption or heat dissipation and cooling that arise as a consequence of the increasingly dense circuits persist.

Though Moore's law will end, demand for increased computational power will certainly not. As such, new methods must be either created or adapted from current technology and software to allow for increased computational power in the absence of denser chips. It is difficult to predict the future of technology; however, a number of new techniques using current software, hardware, and algorithms, are currently being investigated under the umbrella term of approximate computation. These techniques are based on observing that the allowance of small errors in computation may allow for disproportionately large savings in power consumption - allowing a 5% classification loss in a K-means clustering algorithm, for example may allow up to fifty times the energy savings (Mittal (2016)).

Though the use of approximate computation may produce unease due to the introduction of the errors, there are well-known sources of errors in "deterministic" algorithms used today. Because these algorithms give the same output for runs with identical input, the output is often treated as precise; however, parameters are often known only to a certain level of precision, and there may be errors introduced in the computer code itself that give the output itself a level of uncertainty. The output is, in essence, already random as a function of the uncertainty on the inputs, with the replicable nature of the code hiding this fact. Within the statistical community, these sources of uncertainty have been accounted for within statistical calibration models, such as in Kennedy and O'Hagan (2001) and Higdon et al. (2004), and more recent work such as Chkrebtii et al. (2016). These models allow for predictions which incorporate uncertainty explicitly. Known sources of uncertainty which have not generally been a focus statistically have commonly been studied by the mathematical community. For example, the truncation error is commonly used at the local stage to adjust the step size of the procedure or the mesh size of calculation, as in the commonly used method of Fehlberg (1969) or the adaptive meshing methods of Berger and Oliger (1984). The global truncation errors such have traditionally been treated with boundaries, such as in Ralston (1962), that focus on a maximum accumulated local error, assuming perfect knowledge of the true solution.

However, the errors which have commonly been treated mathematically are not fundamentally different than the sources of uncertainty considered probabilistically by the statistical community. The magnitude of these errors has been small enough as to be sufficiently ignored in all but the most potentially catastrophic cases; however, the effect of approximate computing is to magnify or perform techniques that mimic these errors. For example, the well-known floating point truncation error is magnified by the use of precision modification as in Yeh et al. (2007), while the use of linear approximations for calculations of nonlinear functions in Zhang et al. (2014) mirrors traditional solver error. A more complete list of approximate computation techniques may be found in Mittal (2016).

The goal of this paper is to show that noise from an approximate computation process may be treated as both probabilistic and equivalent, in a comparative framework and magnitude sense, to noise that is introduced due to other commonly know sources of uncertainty. Knowledge of this equivalence will be helpful in future allocations of resources when performing modeling using approximate computation methods. This paper will use two examples in order to address these sources of uncertainty: the Duffing equations, a simple one-dimensional chaotic model that will be analyzed on the trajectory-following domain, and the shallow water equations, a commonly used wave model that will be analyzed on the aggregate domain. These are examples of systems that are solved by integrating forward in discrete time steps.

Section 2 will discuss the various sources of uncertainty studied in this paper, along with a brief discussion of the effects of approximate computation and chaos on uncertainty. Sections 3 and 4 will show how the effect of introducing noise into the computational process compares to effects of more traditional inaccuracies, using two specific examples in the trajectory-following and aggregate settings. Finally, section 5 gives a general call to further understand and incorporate these sources of error into uncertainty quantification methods and the computational process.

2 Sources of Uncertainty

In any run of a computer code there are multiple sources of uncertainty - Kennedy and O'Hagan (2001) lists parameter uncertainty, model inadequacy, residual variability, parametric variability, observation error, and code uncertainty. Though all these sources may be important in the analysis, it is not necessary to include all of these sources in order to understand the effects of approximate computation. In particular, this paper will not consider discrepancy, rather treating the model as adequate in the sense that the underlying set of equations the code perfectly describe the physical system considered. Furthermore, the codes considered will be what is traditionally defined as deterministic - apart from the approximate computation - and as such will always produce identical output given identical starting conditions, and so residual variability will not be incorporated. Lastly, no observations are used for the chosen codes, and hence observation error is of no concern. This simplifies the problem to a small set of familiar uncertainties.

2.1 Parameter and Starting Value Uncertainty

Parameters and starting values which control various properties of the system will be necessary in implementation of the code. These parameters may not be known with precision, and uncertainty in the parameters will propagate through into uncertainty in the resulting solution. Similar to parameters, starting values are also both required and commonly unknown with precision, and uncertainty in their specification will also propagate through to add uncertainty to the resulting solution.

2.2 Solver Uncertainty due to Discrete Step Size

There are, of course, multiple numerical techniques which can be used to solve a system, but generally a numerical solver takes discrete steps forward in time, either with a constant or adaptive step size. At each step, a small amount of error is introduced, with larger errors generally being introduced for larger time steps. Though this source of error is often acknowledged, it is also commonly ignored, as it may be reasonable to assume that this error is small enough not to affect the resulting output when the step size is sufficiently small and the code is run for a sufficiently short amount of time. As such, enlarging the error by increasing the step size error may provide a reasonable analogue to other sources of approximate computation uncertainty.

2.3 Uncertainty from Approximate Computation

The type of uncertainty introduced through approximate computation does not fit well into the previous categories. Rather than initial uncertainty added to the initial parameters or conditions and propagated through, the nature of approximate computation introduces a small amount of error at certain steps or at each step of the computational process. This error may not be the same for each run of the algorithm and varies from a small amount at added at each step to larger amounts added at random steps. The exact form of the uncertainty depends on how the approximate computation is performed. However it is considered, an additional source of uncertainty is introduced into the resulting numerical solution, and must be accounted for.

It is important to note that many of the sources of error introduced by approximate computation are already present, though their magnitude is small enough as to be dwarfed by the more traditional sources of error. The implication of approximate computation is that such sources of error may be brought to a level commensurate with the more traditional sources of error. For example, it is widely known that floating point computation introduces errors which may accumulate over long calculations. If a program is run in double precision, this error may be small - but if, as in approximate computation, the precision is lowered, the error will be exponentially larger, and may affect the final computation in ways that raises issues of practicality. A further example is given by truncation error, common in scientific applications which approximate solutions to systems with truncated sums. Truncation of fewer terms of an infinite series gives faster approximation, but at the cost of computational accuracy. If approximate computation is used, linear approximations to functions may be analogous to linear approximations given by truncation error.

2.4 Effects of Chaos on Uncertainty

Chaotic systems amplify the effects of all other types of uncertainty. The general incorporation of chaos into an uncertainty quantification framework will require much work and careful thought which is beyond the scope of this paper (an attempt at deriving initial conditions of a simple chaotic system may be found in Du and Smith (2017), showing the difficulty of the problem); yet, a brief discussion is warranted, as one example in this paper uses chaotic divergence as a measurement itself.

Formally, chaotic systems have a positive Lyapunov exponent, indicating an exponential divergence in trajectory (Strogatz (2014)). Informally chaotic systems display sensitive dependence on initial conditions – any changes in input will eventually cause a change in the resulting output, no matter how small, and the exponential nature of divergence means that once any practical divergence begins the effects of chaotic uncertainty will overwhelm all others and prediction will only be possible on the probabilistic scale.

Though chaotic systems are constantly diverging, in a sense, prior to divergence of practical significance it is natural to track the trajectory of the system in phase space, and if possible compare to that of a "true" solution. Unfortunately, chaotic systems are generally unsolvable beyond simple and one-dimensional maps, so the "true" solution is unavailable. After divergence, the uncertainty may be analyzed on the aggregate domain by taking measurements on an ensemble of systems that differ in the initial conditions. In this method, the probabilistic scale of the output is itself observed in order to detect changes. In climatology, for example, the chaotic nature of the models themselves necessarily leads to differing output based on as little as the computer system the code is being run on - validation of code in this scenario has been accomplished by creating an initial condition ensemble, commonly used to measure uncertainty in climate models (Stensrud et al. (2000)), on a machine whose accuracy is unknown, which post-divergence can then be compared to an ensemble from a trusted machine to determine accuracy (Baker et al. (2015) and Milroy et al. (2016)).

The effects of divergence are not always apparent - indeed, the process will still be running with no apparent errors, just in an incorrect area of the phase space as compared to a system with no introduced errors. This effect on uncertainty is unique to chaotic systems, and as such, modeling techniques that incorporate all other sources of uncertainty will only be valid prior to divergence. It is for this reason that in the example of the chaotic system in this paper, estimated divergence time will be a response variable.

3 Chaotic Duffing Equations

A simple and well-known system that displays chaotic tendencies is the forced Duffing equation, given by

$$x''(t) + dx'(t) - x(t) + x^3(t) = \gamma \cos(\omega t) \tag{1}$$

Here, $x(t)$ represents the horizontal displacement at time t of a "an inverted, elastic, metal pendulum that is attracted between two magnets." (Berliner (1991)). In these equations, the parameter d controls the amount of damping, γ controls the amplitude of the driving force, and ω controls the period of the driving force. Starting values for the position $x(t)$ and velocity $x'(t)$ must also be chosen.

As a chaotic system, an exact analytical solution is not available, so a fourth order Runge-Kutta method with a step size of $h = 5 \times 10^{-8}$ was used to obtain a close approximation, within the observed time period of the system, of an analytical solution to the Duffing equations with parameters $d = 0.2, \gamma = 0.3, \omega = 1$ and starting values of 0 for both position and velocity. These parameters and starting values were chosen to lead directly to a chaotic system, with no other possible non-chaotic cycles revealed in extensive testing in disrupting the system away from its "true" trajectory. The system was run to a maximum time of 100 time units (a total of two billion steps).

Define the numerically solved position and velocity from the run with step size $h = 5 \times 10^{-8}$ as the "true" position and velocity $u_0(t)$ and $v_0(t)$. Furthermore, define the divergence time of a run with numerically solved position $u(t)$ as

$$div(u(t), u_0(t)) = \min_t |u(t) - u_0(t)| > 0.01 \tag{2}$$

This is the first time at which the absolute difference in magnitude between the "true" and altered systems ($u_0(t)$ and $u(t)$, respectively) is larger than 0.01. This quantity of interest (QoI) of the first point that specified difference in the state is reached is a reasonable expression the impact in practice where the goal is predicting the state.

For the parameters d and ω in equation (1) and both of the starting values for position and velocity, the divergence time, as defined in equation (2), was calculated for a run with a parameter or starting value that was larger than the true value, holding all other parameters and starting values constant. Only positive differences were considered due to symmetry of the resulting divergence plots.

This one-at-a-time manipulation of the parameters is known to have several weaknesses - interactions between parameters will not be detected, and the results can be misleading in the presence of collinearity - yet the simple design is sufficient for the purpose of directly comparing the effects of added uncertainty at various sources.

The divergence time plotted as a function of the difference in true value is shown below in Figure 1.

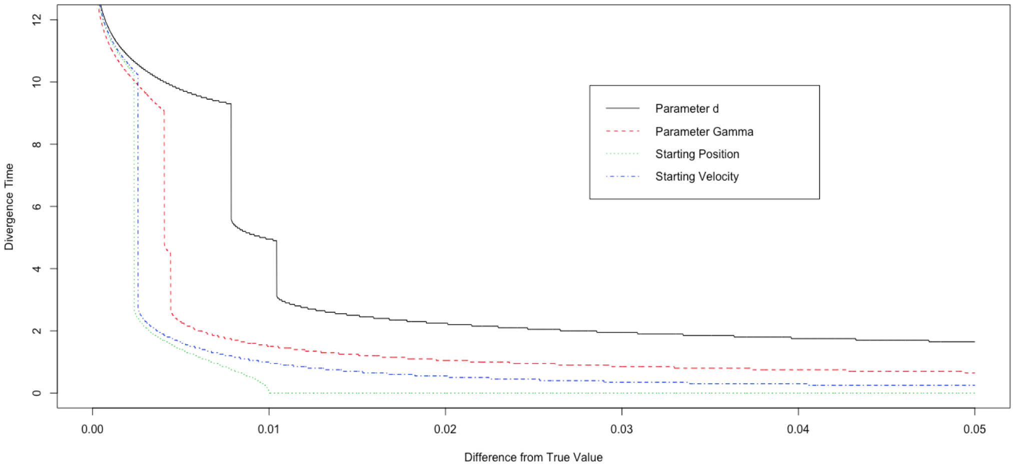

Figure 1: As the difference from the true parameter value or initial condition increases, the time until a specified difference in position from the true system - approximated by a run with specified parameters and step size $h = 5 \times 10^{-8}$ - decreases, with different times for different system inputs. The system is guaranteed to diverge given any uncertainty from the true system due to the chaotic nature of the Duffing equations for the specified parameters. The horizontal axis shows the difference between the true parameter value or initial condition and the one used in a given run of the Duffing equations, while the vertical axis gives the first time a specified difference in position from the true system was observed. Only positive differences are shown, as the time until a specified divergence from the true system exhibits symmetry with negative differences. The discontinuity is caused by the chaotic nature of the Duffing equations.

In general, changes to the parameters and initial conditions produced measurable changes in the divergence. These changes in divergence can be compared directly to, for example, show how different magnitudes of error in different quantities may lead to the same divergence time.

The plot appears to have several "jumps" - the reason for this that while the "true" and theoretical solutions may reach the divergence criterion of an absolute difference larger than 0.01 at one time point, the absolute difference may narrow again to within the tolerance for some period of time before separating again later, and so if divergence does not occur in time period, the solution may not reach another region of phase space with a strong potential for divergence until later on.

A natural question arises as to when divergence times introduced by misspecification of the parameters or initial conditions is roughly equivalent to divergence times introduced by step size errors, which are ignored. In fact, plotting the divergence time as a function of the step size (in Figure 2 below) yields a plot that is not dissimilar to Figure 1 above.

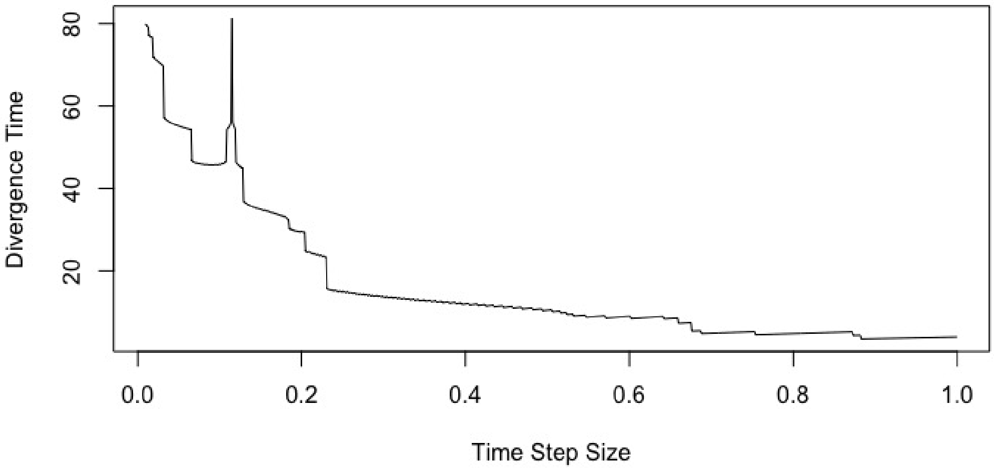

Figure 2: Increasing the step size has a similar effect as introducing an error into the initial parameters or conditions, and gives a plot similar in form to the one in Figure 1. The horizontal axis gives the step size used in a given run of the Duffing equations with specified parameters and initial conditions, while the vertical axis gives the first time a specified difference in position from the true system was observed. As the error introduced at each step increases, the general trend is for the time required until a minimum separation from a reference solution to increase. Again, the chaotic nature of the Duffing equations causes unusual features to the graph - in this case, the spike at around $h = 0.1$

At roughly a step size of $h = 0.25$, the divergence time is equivalent to divergence times caused by even small misspecification of the initial values and parameters. Once again, the nature of there being regions of phase space which are more likely to cause divergence, causing both

In order to simulate the effects of approximate added computation, a small amount of $N(0, \sigma^2)$ noise was added to the position and velocity at each step of the Runge-Kutta process and the divergence time from the reference $(u_0(t), v_0(t))$ solution was calculated. This was chosen to represent generic noise - in truth, the noise standard deviation could vary depending on the location of the process in phase space (a key concept behind adaptive step sizing and meshing), or could be correlated between steps (as in floating point error). However, the lack of knowledge about the specific features of the noise gives pause to assigning such properties, which could cause an underestimation of the effects of the noise on the process. Instead, a general independent random noise is preferred.

A time step of $h = 0.01$ was used - as shown in Figure 2, the time step error for this process will typically not cause a divergence until approximately 80 units of time, though it must be acknowledged that this error will likely interact with other errors when taken in compound. The distribution of divergence times is not unimodal and symmetric, as the nature of the Duffing equations may allow. Nevertheless, statistics of the divergence time, which is likely of interest to the researcher, may be easily calculated. The median divergence time from 2500 simulations for various noise levels is shown below in Figure 3, superimposed on the graph of divergence times from usual sources of uncertainty in Figure 1.

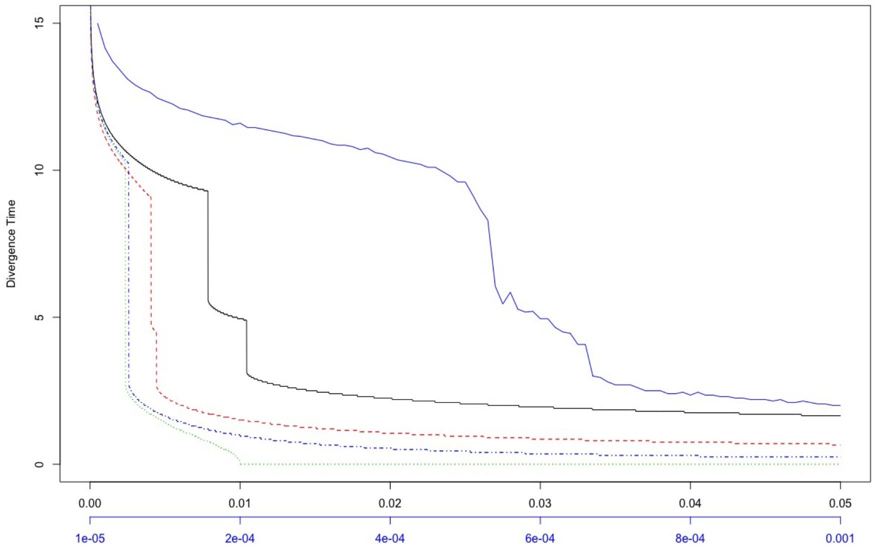

Figure 3: Divergence times from random introduced noise can be directly compared to divergence times from traditional sources of uncertainty. This plot shows Figure 1 above, along with the median divergence time from added noise in 2500 runs of the Duffing equation given the true specified parameters and intial conditions. The top horizontal axis is the same as in Figure 1, while the bottom horizontal axis gives the standard deviation of normal $N(0, \sigma^2)$ noise added at each step in the Duffing equations. The vertical axis once again gives the first time a specified difference in position from the true system was observed.

Distributionally, for the smallest noise added $\sigma = 0.0001$ the median divergence time was approximately 12.4 units of time, with a central 95% quantile range of 11.00 to 24.05 units of time. This range of possible divergence times due to the small added noise is roughly equivalent to the range of divergence times cause by an absolute difference from the true value of the parameter d of between 0.00002 to 0.0018, or an absolute difference from the true value of the parameter γ of between 0.00002 to 0.00155, or an absolute difference from the true value of the starting position of between 0.00002 to 0.0014, or an absolute difference from the true value of the starting velocity of between 0.00002 to 0.0015, or using a step size of between 0.325 to 0.786. If a researcher is comfortable with these small differences, then the researcher should be comfortable with approximate computing of the noise amount described in this paper.

4 Shallow Water Equations

The shallow water equations are a set of partial differential equations derived from the Navier-Stokes equations representing water flow when the wave length is much larger water depth (as in tsunamis). The set of partial differential equations define the shallow water equations are given by:

$$\begin{aligned} \frac{\partial(dH)}{\partial t} + \frac{\partial(dHU)}{\partial x} + \frac{\partial(dHV)}{\partial y} &= 0 \\ \frac{\partial(dHU)}{\partial t} + \frac{\partial(dHU^2 + \frac{1}{2}gdH^2)}{\partial x} + \frac{\partial(dHUV)}{\partial y} &= 0 \\ \frac{\partial(dHU)}{\partial t} + \frac{\partial(dHUV)}{\partial x} + \frac{\partial(dHU^2 + \frac{1}{2}gdH^2)}{\partial y} &= 0 \end{aligned} \tag{3}$$

where H is fluid height, U is fluid velocity in the x direction, V is fluid velocity in the y direction, d is the density of the liquid (which is $d = 1$ for the density of water), and g is gravity (which is $g = 9.8m/s^2$ for the gravity of earth).

These equations were solved numerically using the Lax-Wendroff method, which is second order accurate in time and operates by dividing a space into a grid consisting of N cells, taking half-steps in X and Y direction at each cell, and then using those to take take full steps. This method does not require the calculation of a Jacobian to implement.

Since maximum computational efficiency of the boundary zone was a concern of the demonstration, a simple dissipating boundary zone was implemented using the method of Martinsen and Engedahl (1987), first by calculating the updated height and velocities in each direction within the boundary zone and then by relaxing the velocities towards zero by a fixed percentage, essentially allowing the wave to dissipate by adding a friction coefficient within the dissipating boundary zone.

A wave generation method was implemented by creating a zone in which energy was added in both directions and allowed to build to a wave before reaching the observed area. The result of this method is an $n \times n$ grid, with three zones - an outer L-shaped border in which waves are created and allowed to propagate, a central square zone that will be used for measurement of the system, and an outer L-shaped border opposite the wave generation zone where the energy of waves is dissipated. All energy added and absorbed was only included at time periods that were multiples of 0.05 units of time.

In this way, the energy added and removed from the system is done so in a controlled and consistent manner that will not increase as the step size h is changed between runs.

The solver was run with a Gaussian initial perturbation added to the water to induce initial movement in the water in order to show the dissipation of initial energy added to the system. The effect of the absorbing boundaries and the wave generating boundaries was to create a "wave pool" steady state after a period of time. With U_j and V_j as the full-step velocities at each interior (non-absorbing) cell j, an analogue measure of the kinetic energy in the system is given by the average magnitude of the discrete fast Fourier transform (DFT) at each point on the grid. After the initial disturbance dissipates, the system should converge to a steady state consisting of only the energy added, creating a "wave pool" effect . This is shown in a plot of the average DFT versus time for various initial drop sizes (differing heights and radii) in Figure 4 below.

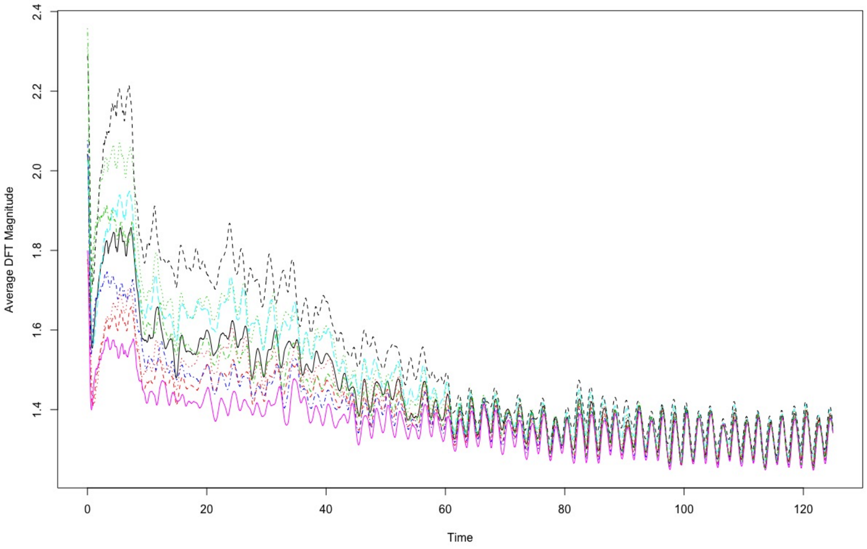

Figure 4: This figure shows the average discrete Fourier transformation magnitude, where the DFT applied to an entire grid, converging to the same steady state regardless of initial disturbances with differing drop peak heights and drop radii. Each point represents a single grid at a single time value. That all converge to the same distribution shows that comparisons between systems with slightly altered conditions are valid, as changes in resulting distributions may be attributed to the different conditions and not simply a secondary steady state. The horizontal axis gives the time, while the vertical axis gives the average discrete Fourier transformation magnitude. Each line represents a different initial perturbation - in terms of drop radius and magnitude - added to the system.

After the effect from the initial drop has dissipated, the system can be analyzed in aggregate by observing the distribution of the average DFT in the steady state. Since the distribution is not necessarily unimodal or symmetric, a boxplot summary is appropriate. This steady-state distribution will change depending on, for example, the parameter values, which there may be uncertainty about. As an example, the gravitational constant for water is the well-known $g = 9.8m/s^2$. Suppose, however, that this constant was unknown, and different gravitational constants were considered instead The aggregate distributions are then shown in Figure 5 below.

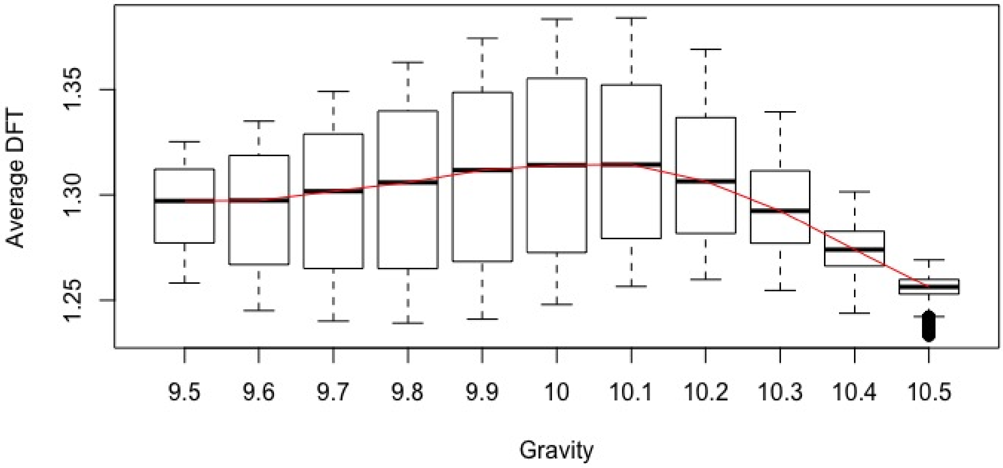

Figure 5: This figure shows a boxplot of the average discrete Fourier transformation magnitude, again applied to an entire grid, after initial effects have dissipated and only the steady-state remains. The horizontal axis gives the gravitational constant used in the run of the shallow water equations, while the vertical axis gives a boxplot of the average discrete Fourier transformation magnitude. In terms of Figure 4, this would be a boxplot of the points on a single curve taken on a large slice of time after it has converged to the consistent sinusoidal pattern. This figure shows that changes to the gravitational constant do affect the distribution of average DFT magnitude, but in a nonlinear and asymmetric manner - increasing the gravitational constant by a specified amount changes the distribution in a different way than decreasing it by a specified amount.

A simple change of the parameter can drastically change both the center and variation of the distribution of the aggregate quantity. As one would expect, changes near the true value of 9.8 m/s^2 appear to have a slight shift on the distribution, for larger values the distribution can change drastically. This can be due to physical processes changing and interacting in unanticipated ways as the parameters change. If the speed at which waves propagate through the system is altered, constructive or destructive interference can occur between the waves currently in the system and the condensed distribution for the highest gravitational constant of 10.5 m/s^2 in Figure 5 above appeared to be caused by changing the speed at which the waves moved through the water so that there was destructive interference with the wave generating process.

Similarly, suppose that time steps is considered as an uncertain quantity. Then the aggregate quantity is shown below in Figure 6.

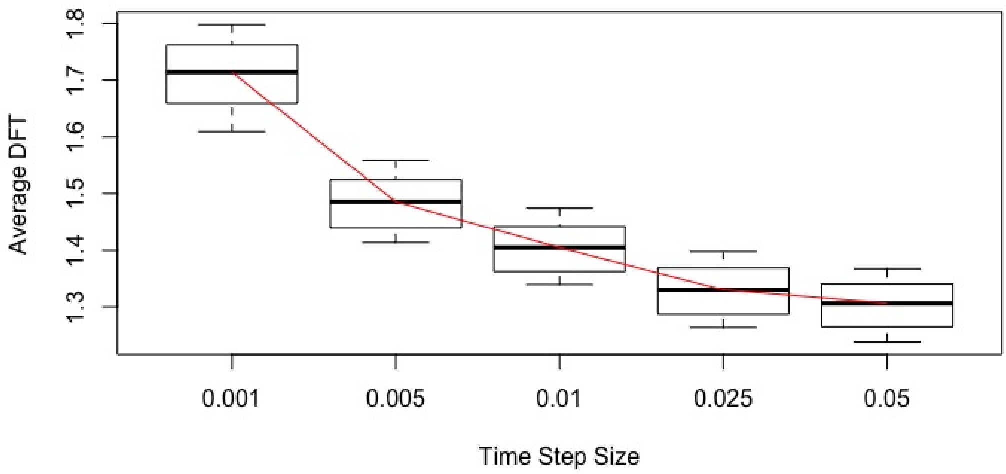

Figure 6: Similarly to Figure 5, this figure also shows a boxplot of the average discrete Fourier transformation magnitude, again applied to an entire grid, after initial effects have dissipated and only the steady-state remains, for multiple time step sizes. The horizontal axis gives the time step used in the Lax-Wendroff method, while the vertical axis gives a boxplot of the average discrete Fourier transformation magnitude. Altering the time step size directly affects the distribution of average DFT magnitude, though this will likely depend on the specific solver used.

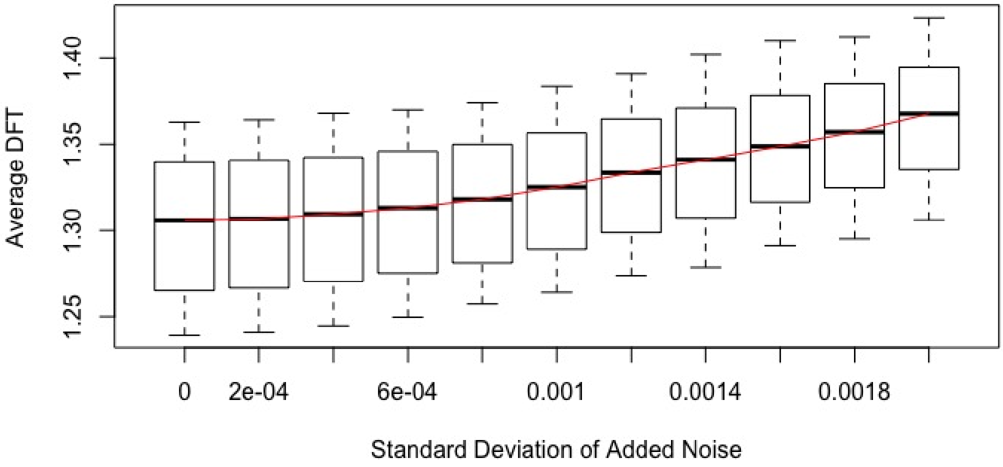

Figure 7: Similarly to Figures 5 and 6, This figure also shows a boxplot of the average discrete Fourier transformation magnitude, again applied to an entire grid, after initial effects have dissipated and only the steady-state remains, for multiple levels of added noise within the computational process. The horizontal axis gives the standard deviation σ noise added at each time step, while the vertical axis gives a boxplot of the average discrete Fourier transformation magnitude. For the noise, a small amount of normal $N(0, \sigma^2)$ error was added to each point on the grids representing the velocities in the X and Y directions at each time step during the computation. The distribution of average DFT magnitude distributions for added noise systems show that noise added does affect this distribution, often in similar ways to other types of uncertainty.

A key conclusion is that, much in the case of the trajectory-following Duffing equations, the introduction of random noise into the computational process induces an uncertainty in the measured quantity of interest which may be directly compared to the usual uncertainty obtained by other common sources of error, such as step size error or parameter uncertainty.

5 Conclusion

Code that is currently thought of as "deterministic" is in no way deterministic. All computer codes are stochastic. The fact that output is reproducible does not mean that it is correct; all codes are subject to random elements, both seen and unseen. Even if parameters were known with complete precision, commonly ignored sources of error such as floating point and solver error render the result as one possible outcome from a fully stochastic process. The sooner this is recognized, the sooner a full analysis of computer codes, incorporating all levels of uncertainty - not just the initial inaccuracies which exists outside of the code, but also those internal to the computational process - can begin. Though currently small relative to error external sources of code inaccuracies, it does not follow that a source of error currently insignificant should be ignored in perpetuity.

The future of computation depends on recognizing this fact. If increased computational power is to be demanded without radical changes to technology, it will depend on understanding and incorporating these errors, which are currently poorly understood, into the computational process. For the majority of the Moore's law era computation has moved ever forward in accuracy; at each technological step acknowledging the inaccuracy of previous computations while simultaneously purporting complete accuracy of the current step. Rapid increases in computational power have promised a fix to current problems of time or accuracy in the near future. The not-so-far future will no longer see these increases. It is time to acknowledge inaccuracy in all computations, and harness this to our advantage.

Thankfully, the particular examples in this paper appear to show these errors influencing the output in much the same manner as those traditionally taken into account in uncertainty quantification analyses. However, there is still much work to be done. The statistical nature of these errors is poorly understood, and there is much fruitful research waiting to be performed in modeling and simulating errors that more accurately mirror those from approximate computation. Particularly, though the errors in this paper have been treated as independent, it is almost certain that they are correlated in application. Further results which generalize the result of introducing intentional errors into common computational methods may also provide a potential avenue for generating new approximate computation methods.

References

Baker, A. H., Hammerling, D. M., Levy, M. N., Xu, H., Dennis, J. M., Enton, B., Edwards, J., Hannay, C., Mickelson, S. A., Neale, R., Nychka, D., Shollenberger, J., Tribbia, J., Vertenstein, M., and Williamson, D. (2015), "A new ensemble-based consistence test for the Community Earth System Model (pyCECT v1.0)," *Geoscientific Model Development*, 8, 2829–2840.

Berger, M. J., and Oliger, J. (1984), "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *Journal of Computational Physics*, 53, 484–512.

Berliner, M. L. (1991), "Likelihood and Bayesian Prediction of Chaotic Systems," *Journal of the American Statistical Association*, 86(416), 938 – 952.

Cavin, R. K., Lugli, P., and Zhirnov, V. V. (2012), "Science and Engineering Beyond Moore's Law," *Proceedings of the IEEE*, 100(Special Centennial Issue), 1720–1749.

Chkrebtii, O. A., Campbell, D. A., Calderhead, B., and Girolami, M. A. (2016), "Bayesian Solution Uncertainty Quantification for Differential Equations," *Bayesian Anal.*, 11(4), 1239–1267.
URL: *https://doi.org/10.1214/16-BA1017*

Du, H., and Smith, L. (2017), "Rising Above Chaotic Likelihoods," *SIAM/ASA Journal on Uncertainty Quantification*, 5(1), 246–258.
URL: *https://doi.org/10.1137/140988784*

Fehlberg, E. (1969), Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems,, Technical Report 315, NASA.

Higdon, D., Kennedy, M., Cavendish, J., Cafeo, J., and Ryne, R. (2004), "Combining Field Data and Computer Simulations for Calibration and Combining Field Data and Computer Simulations for Calibration and Prediction," *SIAM Journal on Scientific Computing*, 26, 448–466.

Kennedy, M., and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society B*, 63(3).

Martinsen, E., and Engedahl, H. (1987), "Implementation and Testing of a Lateral Boundary Scheme as an Open Boundary Condition in a Barotropic Ocean Model," *Coastal Engineering*, 11, 603 – 627.

Milroy, D. J., Baker, A. H., Hammerling, D. M., Dennis, J. M., Mickelson, S. A., and Jessup, E. R. (2016), "Towards characterizing the variability of statistically consistent Community Earth System Model simulations," *Procedia Computer Science*, 80, 1589 – 1600.

Mittal, S. (2016), "A Survey of Techniques for Approximate Computing," *ACM Computing Surveys*, 48(4), 1 – 33.

Moore, G. E. (1965), "Cramming More Components onto Integrated Circuits," *Electronics Magazine*, 8.

Ralston, A. (1962), "Runge-Kutta Methods with Minimum Error Bounds," *Mathematics of Computation*, 16(80), 431–437.

Stensrud, D. J., Bao, J.-W., and Warner, T. T. (2000), "Using Initial Condition and Model Physics Perturbations in Short-Range Ensemble Simulations of Mesoscale Convective Systems," *Monthly Weather Review*, 128(7), 2077–2107.
URL: *https://doi.org/10.1175/1520-0493(2000)128¡2077:UICAMP¿2.0.CO;2*

Strogatz, S. (2014), *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Studies in Nonlinearity, 2 edn Avalon Publishing.

Yeh, T., Faloutsos, P., Ercegovac, M., Patel, S., and Reinman, G. (2007), The Art of Deception: Adaptive Precision Reduction for Area Efficient Physics Acceleration,, in *40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007)*, pp. 394–406.

Zhang, H., Putic, M., and Lach, J. (2014), Low Power GPGPU Computation with Imprecise Hardware,, in *Proceedings of the 51st Annual Design Automation Conference*, DAC '14, ACM, New York, NY, USA, pp. 99:1–99:6.
URL: *http://doi.acm.org/10.1145/2593069.2593156*