Computer Simulation in Low-Temperature Plasma Physics: Future Challenges

Miles M. Turner

School of Physical Sciences and National Centre for Plasma Science and Technology, Dublin City University, Dublin 9, Ireland

E-mail: miles.turner@dcu.ie

Abstract. Computer simulations can be carried out with various aims. Perhaps the most challenging is prediction under conditions where experiments are difficult or inaccessible, especially when failure to predict adequately may have unhappy consequences. There is, probably, not much confidence at present in the capability of low-temperature plasma physics simulations in such a context. Other fields have attempted to meet this challenge using a collection of techniques collectively known as "Verification and Validation," or "V&V." These are methods for enhancing confidence in the correctness and fidelity of computer simulations. This paper surveys these techniques and discusses their application to improvements of simulation capability in low-temperature plasma physics.

PACS numbers:

1. Introduction

The potential of numerical calculations as a technique for scientific investigation was appreciated early in the development of modern science. Application of such methods, however, was limited by the laborious manual effort involved, until modern programmable computers became widely available during the nineteen-fifties. Since then, the importance of numerical approaches to technical calculations has increased, and the ambition of such calculations has also increased, in proportion to the computational resources that have become available. At present, so-called exascale computers are expected to be operational in the foreseeable future. These machines will deliver in excess of 10^{18} floating point computations per second. Even a fraction of these resources allows simulations to address physical systems so complex that analytic insight into their behaviour is extremely difficult. Consequently, one may be in the position of reaching important conclusions or making important decisions informed primarily by the results of computer simulations [29]. Calculations relating to the safety of nuclear devices might be an extreme example. Under these conditions, the fidelity of the computer simulations is of obvious and vital importance. Here, the essential opacity of the computational process is a challenge. A strictly mathematical argument can be fully published, and consequently checked by any interested party. In computational work, questions arise not only about the correctness of the computer program (which may contain many thousands of lines of code): Other factors than the correctness of the code influence the results. These include the mathematical model that the computer program is designed to solve, and the physical and numerical parameters of the simulation, which may be numerous. A widespread and customary assumption is that these concerns are addressed as a matter of course by diligent practioners, and need not be discussed systematically and explicitly whenever computational results are reported. Over the last decade or two, however, this position has been undermined by evidence that the diligence of even expert practitioners may be insufficient protection against faulty computations [32, 10, 11, 27, 29]. This impression is reinforced by a number of well-documented failures, attributed to computation, *e.g.* [13, 19, 3]. This has led (from 1986 onwards) to both calls for more detailed reports of computational work [36, 6, 44], and careful consideration of the kind of evidence that should be advanced to sustain the assertion that a computational result is "correct" (or at least "fit for purpose") [29]. Clearly, if we cannot make a convincing argument that computation leads to "correct" results, the ambitions of computational science will be seriously limited.

There has been some divergence between scientific and engineering communities [29] concerning these matters. In scientific communities, the primary purpose of simulation is often seen as developing insight that will lead towards, for example, analytic understanding. In this context, the validity of the analytic understanding need not depend on whether or not the simulations were strictly correct: Major results in the physical sciences are rarely if ever established solely on the basis of computation, for example. If the system under investigation defies analytic understanding, or if expert

judgement cannot otherwise distinguish between a correct and a faulty result, then the simulations must stand alone. In this case, the correctness of the simulations becomes a crucial question. This often the case in an engineering context, in which simulation is used to establish a number (such as a breaking strain), rather than to explore qualitative behaviour. Serious engineering failures with grave legal consequences have been attributed to faulty computations [19]. Clearly, great dangers await if simulations developed under the rather relaxed standards of scientific communities are applied to problems in this second category.

The experience of the computational fluid dynamics community is highly relevant. This is a computational specialisation where engineering prediction is a central goal. This is also the community that experienced the so-called "numerical wind tunnel debacle," an episode during which hubristic claims were made about the imminent replacement of physical wind tunnels by simulations [4]. The computational tools and practices in use at the time, however, proved embarrassingly insufficient. This realisation was an important motivation for the development of the techniques known today as "Verification and Validation," or "V&V." These are unfortunate terms, as similar looking words whose meanings in ordinary usage are indistinct. Indeed, they often treated as interchangeable in the scientific literature. However, in recent usage "verification" refers to the process of testing a computer program, while "validation" is concerned with the testing of mathematical models. In this context, we assume that the purpose of the computer program is to solve some mathematical model, in which case verification is the process of demonstrating that the program executes this solution correctly. Validation is the process of establishing that the solutions that are found are consistent with experimental observations. Clearly, a computer program must be verified before any meaningful validation can occur, since otherwise no distinction can be drawn between errors in the computer program, and deficiencies in the mathematical model.

A simple metric for the "correctness" of a computer program is the number of defects per line of code. Best practice leads to defect densities of around one per ten thousand lines of code [11]. Scientific and engineering code has been shown to exhibit defect densities that are typically ten to a hundred times larger than this [10, 11]. In extreme cases, defect densities as high as one per ten lines have been observed[10]. These metrics refer to professionally maintained code, employed as an aid to making economically consequential decisions. Moreover, the presence of these defects demonstrably affects the results of calculations, to a degree that materially influences decisions based on these calculations [10]. There are, therefore, grounds for serious concern.

Of course, the mathematical structure of the underlying model and correct computer programming are not the only factors contributing to effective computer simulation. The mathematical model will usually contain physical parameters that are not known with certainty, and this uncertainty may limit the predictive capability of the simulation as much or more than any other consideration. When the number

of uncertain parameters is large, special techniques may be needed to investigate how uncertainty infects the computed results, and to determine which parameters are causing the uncertainty. An obvious example of this kind of problem arises when complex chemistry models are employed [48, 49], which may feature hundreds of uncertain parameters, in the form of rate constants and other related data. Investigations of these matters use techniques of "Uncertainty Quantification," or "UQ." In essence, the aim of UQ is to associate error bars with the computed results. Evidently, UQ is required to establish a meaningful concept of "agreement" of computer simulations and experiments in the context of validation. Fields such as combustion science, where complex chemistry models are common, routinely use UQ methods in combination with validation and optimisation of such models to develop "reaction mechanisms" with demonstrated predictive power.

These considerations suggest that there is large scope for improved practice in computer simulation as employed by the low-temperature plasma physics community. Indeed, one might argue that changes in custom and practice are urgently needed in light of the developments in other fields outlined above. The aim of this paper is to survey the main ideas associated with "V&V" and point to examples of their use in low-temperature plasma physics and nearby communities. An important point that emerges from this discussion is that V&V is expensive. Consequently, the benefits must be at least proportionate to the expense. Although this paper is concerned primarily with V&V, this is not the only area where improved practice may be desirable. Improved practice in code development is also likely to have a place [51], but this is beyond the scope of the present paper.

In what follows, we present a brief survey of some of the main ideas that "V&V" has thrown up. Whenever possible, we refer to examples drawn from low-temperature plasma physics, but the present state of development, such examples are rare. In section 2, we discuss techniques for code verification, and we note some special problems that arise in the context of low-temperature plasma physics. We then go on to discuss uncertainty quantification, in section 3, sensitivity analysis, in section 4 and validation issues in section 5. Some implications of these ideas are elaborated in section 6, and concluding remarks are in section 7.

This paper can only glance at the main concepts. There are at least three book length treatments of V&V techniques [16, 35, 29], and hundreds to thousands of pages of writing in journals. Sensitivity analysis is a field with similar scope. A paper of the present length cannot offer a comprehensive overview of all this material, and some important ideas have necessarily been omitted, or treated superficially. The intention of the paper is not be critical of past work in the field of low-temperature plasma physics that was carried out according to the prevailing view of good practice. Rather, the aim is to show that other fields have developed new ideas, the adoption or adaption of which may lead to more powerful computational approaches.

2. Verification

Verification is concerned with testing of computer programs. The general approach involves demonstrating that a particular program can reproduce an exact solution of some suitable test problem. In embarking on such an exercise, we assume that the solution procedure implemented by the program is conceptually valid, so that the only question is whether or not the implementation is correct. When the model equations are solved explicitly, as in the example discussed below, this is almost self-evident. But some approaches do not explicitly solve the model equations. This is the case, for instance, with Monte Carlo solutions of the Boltzmann equation, in which the Boltzmann equation is not explicitly expressed in the computer program. In some such cases, the existness of a convergence proof provides assurances that a correct implementation will lead to the expected solution [50, 23]. Consequently, verification and formal convergence proof are complementary but distinct activities.

Probably everyone involved in computational work recognizes that a complex computer program cannot be *assumed* to be correct. Testing is always required. An obvious procedure is to compare the results of a computation with a solution that is indisputably valid, such as an exact analytic calculation. The natural method is to plot a graph, and see whether the two solutions coincide. There are many examples of this approach in the low-temperature plasma physics literature. However, the computed solution cannot be expected to agree "exactly" with the analytic result, because there will be effects of both numerical parameters and finite precision arithmetic (at least). Whether the two solutions agree acceptably, or not, then appears to be a matter of judgement and possible disagreement. There is a stronger method available, however. A well designed numerical procedure will include some analysis of the accuracy of the scheme. For instance, the Laplacian operator may be expressed

$$\frac{d^2\Phi}{dx^2} = \frac{\Phi_{i+1} - 2\Phi_i + \Phi_{i-1}}{\Delta x^2} + \mathrm{O}\left(\Delta x^2\right). \tag{1}$$

A measure of the distance between a numerically derived solution defined at spatial locations $x_i = i\Delta x$ and an exact solution, $\Phi(x_i)$, is the L^2 norm:

$$L^2 = \sum_i \sqrt{\left[\Phi_i - \Phi(x_i)\right]^2 / \sum_i \Phi(x_i)^2}. \tag{2}$$

Equations 1 and 2 imply that

$$L^2 \propto \Delta x^2, \tag{3}$$

and this relation holds if and only if the numerical solution converges to the exact solution at the expected rate. In practice, this scaling is observed only for a finite range of Δx. For Δx too large, higher order terms in the error polynomial are important. For Δx too small, finite precision effects will become influential. Nevertheless, observing the relationship of equation 3 constitutes a strong test for correct implementation of a numerical solver, amounting, in the view of some [34], to a correctness proof. Whether this be accepted or not, this is a far more rigorous and convincing approach than

inspecting a graph comparing two solutions. This idea is the basis of modern concepts of verification.

An indicative application of this technique is shown in figure 1. This example refers to the calculation of single particle orbits in given electric and magnetic fields, in this case a fixed magnetic field making a certain angle with an electric field that oscillates in time. The particle trajectory is computed using a second order accurate leapfrog procedure [1], and the L^2 norm is computed relative to an exact solution for this case. As can be seen, the relationship between the L^2 norm and the integration time step follows the expected relationship, which shows that the numerical solver is correctly implemented.

An obvious difficulty with this technique is that we must know of an exact solution. Since we often mean to solve, for instance, coupled systems of nonlinear partial differential equations, this seems a serious challenge. However, the exact solution need not be physically significant, and this insight has led to the idea of the "method of manufactured solutions" as the basis for correctness tests [34]. As a simple example, suppose we desire to solve

$$\frac{\partial n}{\partial t} = D\frac{\partial^2 n}{\partial x^2} - kn^2, \tag{4}$$

in some region $0 \leq x \leq L$, with $n(t, x = 0) = n(t, x = L) = 0$. This transport equation might model a recombining afterglow. For a manufactured solution, we may assume any function satisfying the boundary conditions, although a combination of transcendental functions is desirable, because no difference scheme of finite order can reproduce the solution exactly. For instance,

$$n_{\text{manu}}(x, t) = n_0 \exp(-D\pi^2 t/L^2) \sin(\pi x/L). \tag{5}$$

This proposed solution of course does not satisfy equation 4, but it will do so if we add on the right hand side the artificial source term:

$$S(x, t) = k n_{\text{manu}}(x)^2. \tag{6}$$

Then equation 5 is an exact solution of the modified equation, and we can use the technique discussed above to verify that any proposed numerical solver behaves in the expected way. Once we have performed this step, the artificial source term can be set to zero. This example is trivial, but the procedure can be applied to arbitrarily complex systems of partial differential equations [38, 24, 14, 5], and so the verification challenge appears to be met for problems in this category. Advantage may even be found in the unphysical character of these manufactured solutions. For instance, the physical problem may involve equations with terms that differ significantly in magnitude. A manufactured solution can be constructed so that all the terms are of similar magnitude, such that all are exercised in the same test.

The foregoing remarks apply to the problem of verifying a computer program, which is referred to as "verification of code." Even if this procedure produces certifiably correct code, this does not guarantee that every calculation carried out with the code

gives a result that is fit for purpose. This is because each such calculation (presumably) has distinct physical parameters, and in general the numerical parameters must be chosen appropriately for each case. So some method is needed to demonstrate that the numerical parameters have been appropriately selected. A simple and robust procedure is to calculate solutions for three set of related numerical parameters. In the example discussed above, for instance, we might choose $\Delta x = H, 2H, 4H$, for some H that we deem suitable. Presumably the finest grid gives the most accurate solution, and we can estimate the error in that solution by comparison with the solution on a coarser grid. The motivation for computing three solutions (and not two) is to check that the rate of convergence is as expected. This step is known as "verification of calculations." Many journals nowadays mandate that this integrity check be demonstrated as an absolutely minimal requirement for computational results to be accepted as publishable.

For mathematical problems consisting of coupled sets of partial differential equations, the verification problem appears to be substantially solved by the methods outlined above. This does not mean that the problem has been made easy. For instance, an obvious difficulty is that codes may have complicated options, particularly when boundary conditions for multi-dimensional cases must be specified. In principle, each logically distinct configuration of the code requires a separate verification test case. Designing a code that can solve a usefully wide range of problems without generating an unreasonable number of test cases may be difficult. Hatton [10], for example, discovered a program with 500 000 000 paths, each in principle requiring a verification test case. Clearly, such a program is practically unverifiable.

These remarks apply to programs designed to solve coupled systems of partial differential equations. However, not all problems are in this category, including several of great importance in low-temperature plasma physics, as we now explain.

2.1. Hybrid Models

Perhaps the single most important type of simulation used in low-temperature plasma physics is the so-called hybrid model [18]. This is, in effect, a technique for addressing simulation problems that encompass widely varying scales, for instance in length or time or both. The typical structure of a hybrid model includes several loosely linked modules, each of which is more-or-less autonomous. A module finds convergence on its own length and time scale, and exchanges data with other modules that may operate on either larger or smaller scales. A common example is a plasma simulation in which macroscopic parameters, such as species densities and fields, are found by solving moment equations, while microscopic information, such as particle energy distribution functions, are obtained at the kinetic level by solving the Boltzmann equation. These modules may be coupled to still more disparate scales, such as microelectronic features (on nanometre length scales), or gas dynamics (on second time scales). Most likely, one can devise a verification strategy for each module using the principles outlined above, but the problem of verifying the simulation as a whole appears challenging. There are two

difficulties. The simulation as a whole is generally not based on a coherent mathematical model. For instance, the procedures connecting different modules are often informally specified, and indeed establishing that different modules do not make contradictory assumptions might be difficult. Consequently, manufacturing an exact solution for verification purposes appears impossible. Even if the mathematical foundations of the simulation could be clarified, the diversity of solution procedures, with different and perhaps incompatible numerical parameters, will make a coherent concept of convergence difficult to establish. Finally, a hybrid model appears particularly prone to the problem of unreasonable multiplication of verification test cases that was discussed above. The diversity of modules that might be included in a hybrid model allow many different configurations of the simulation, each of which will require a separate verification test case. For these reasons, verifying a hybrid model using established approaches [29] appears difficult, and there would be advantage in research into the mathematical foundations of hybrid models.

2.2. Monte Carlo Codes

There is relatively little writing on the problem of verifying Monte Carlo codes. The challenges are different from those discussed in connection with hybrid models. Monte Carlo codes typically solve a well-defined mathematical model, and therefore a process resembling the method of manufactured solutions could be used. However, in some cases, the process of convergence is complicated by some mixture of statistical and numerical effects. For example, many Monte Carlo codes integrate the trajectories of test particles in time using a numerical procedure involving a finite time step, with the desired outcome a distribution function expressed as a sum over a large number of such trajectories. The distribution obtained in this way is influenced both by statistical effects, and by effects of the finite time step, in ways that are generally not analytically well understood. In some Monte Carlo codes, such as particle-in-cell simulations, the particle trajectories are not independent. The means of demonstrating that such a simulation converges to an expected solution are not straightforward [31]. Moreover, the method of manufactured solutions is not so easy to apply either. This is because implementing an arbitrary source term is not simple. In particular, a source term that is negative in some region of phase space may be problematic. Avoiding such problems will place challenging constraints on the kind of manufactured solutions that may be used. These are not obviously insurmountable difficulties, but at the present time no Monte Carlo code seems to have been verified in this way. In this context, we note that direct solutions of the Boltzmann equation without Monte Carlo elements (such as the convective scheme [12]) do not appear to encounter these difficulties and could presumably be verified using the method of manufactured solutions.

One may well discover exact solutions that exercise some subset of the functions of a Monte Carlo code, and this may permit the assembly of a suit of verification test problems that collectively exercise all the functions of a code. For instance, the nuclear

physics community has developed a large suite of such test problems [42] which exercise many important functions of a Monte Carlo transport code. This approach can certainly increase confidence in the correctness of a code. However, a tempting recourse in the absence of a comprehensive exact test problem is benchmarking, which has some special problems that we will now discuss.

2.3. Benchmarking as a verification technique

Benchmarking is a comparison of codes applied to some test problem. There are difficulties with this natural procedure, relative to verification against an exact solution, which have made this approach unpopular with writers on verification techniques [46]. Perhaps the greatest of these difficulties arises in the case where none of the codes is known to be correct in advance, but the results on the test problem are found to be divergent. This is a distressingly common occurence in benchmarking exercises, and if, as also often happens, the differences cannot be resolved, the result cannot be regarded as verifying anything [43, 10, 37]. For this reason, verification against an exact solution is preferable to benchmarking, when the former is an option. However, there does appear to be a case for benchmarking, when an exact solution is not available, and if there is a clear concept of agreement. A minimum requirement in this case seems to be that all the benchmarked codes should implement the same mathematical model, so that an idea of agreement within a margin of numerical uncertainty defined by a "verification of calculation" approach can employed. In general, agreement at this level appears difficult to find, but even when such agreement is achieved, some caution is warranted. For instance, there is evidence that errors in independently written computer codes tend to cluster (perhaps because some algorithmic elements are psychologically hard to implement accurately) so that simultaneous error in independent codes, although not likely, is not as improbable as one might hope [15].

An example of a benchmark calculation involving particle in cell simulations with Monte Carlo collisions [47] is shown in figure 2. In this case, all of the benchmarked codes implement nominally the same algorithm using the same numerical and physical parameters, and, consequently, if all the codes are correct then the results should be different only for statistical reasons. One can therefore ask whether the results are different by applying statistical tests. In this case, a X^2 test was used, with the conclusion that the codes under test give statistically indistinguishable results. For an earlier (and arguably less rigorous) attempt to develop a benchmark, see Lawler and Kortshagen [20].

Benchmarking appears a tempting technique for hybrid models when the method of manufactured solutions cannot be used. A serious challenge in this context is that while there are many hybrid models in present use, probably no two of them implement the same model in detail. A concept of agreement between such models thus appears difficult to establish.

3. Uncertainty Quantification

A simulation of any complexity depends on a large number of physical parameters. Important examples include chemistry models, which may contain hundreds or thousands of parameters in the form of rate constants and similar data [39, 22, 33, 48, 49]. All these parameters are more or less uncertain, and this uncertainty necessarily propagates into the results of simulations incorporating such models. Clearly, it would be unsafe to draw any conclusions from an uncertain simulation result without having at least characterized the uncertainty. Both quantitative and qualitative features of a simulation may be uncertain, and if the scope of such uncertainty is not known, there is a risk of drawing invalid conclusions. For these reasons, uncertainty quantification, or "UQ," is a usual practice in disciplines that attempt to used complex simulations for predictive purposes. Combustion science and environmental science are examples. Raw quantification of uncertainty is useful, but further analysis allows the uncertainty in simulation results to be associated with the simulation parameters. That is, we can discover which simulation parameters are causing uncertainty in particular simulation results. This procedure is known as "ranking," where the parameters are being ranked in order of their influence on the simulation results [49].

Figure 3 shows an example drawn from Turner [48]. These data are obtained used a chemistry model for mixtures of helium and oxygen. The figure shows a selection of trajectories for helium metastable densities as a function of time. In these calculations, each rate constant in the chemistry model has an associated uncertainty, constructed, for example, using an experimental error bar. These uncertainties are used to associate a probability distribution with each rate constant, and each trajectory in figure 3 is computed with a different set of rate constants drawn from these distributions. Each trajectory is therefore consistent with the available information about the rate constants. Clearly, any attempt to compare this model with an experiment should account for the variety of possible trajectories. A summary view of the variety of possible trajectories appears in figure 4. In this representation, trajectories show in a figure 3 are ordered at each time coordinate, and the shaded area represents the two middle quartiles, *i.e.* at each time time coordinate, 50 % of the trajectories fall in the shaded region. Of course, this means that the other 50 % of the trajectories are outside this region. The width of the shaded region is therefore a representative error bar for the simulation results. As has been shown elsewhere [48], the margin of uncertainty in this kind of calculation varies with species and simulation conditions in ways that do not appear easy to predict. However, the relative uncertainty can exceed a factor of ten in some circumstances, which is likely to be important for almost any computational purpose.

A broader view of uncertainty can be taken. For example, most computer simulations omit some features of the corresponding physical system, for instance by reducing the number of dimensions. These and similar simplifications can also be regarded as sources of predictive uncertainty, but they are not susceptible to quantification by the techniques discussed above [45], and this interesting problem will

not be further discussed in the present paper.

4. Sensitivity Analysis

The data shown in figures 3 and 4 indicate the magnitude of the uncertainty in the model due to the uncertainty in the rate constant data, but they offer no insight into the processes that cause the uncertainty. The procedure for associating the uncertainty in model predictions with particular model parameters is known as "screening." A screening algorithm computes various metrics that quantify the contribution of each model parameter (or "factor") to the uncertainty in each predicted quantity (or "model output"). The basic method is essentially a Monte Carlo procedure in which, at each step, a single randomly selected factor is changed, while all the others are held constant. In general, changing one factor will change all the model outputs. The changing of one factor in this fashion is known as an "elementary effect." To fully explore the influence of each factor, elementary effects needs to be evaluated at several places in the parameter space, *i.e.* for a selection of values of each of the factors that are held constant while the elementary effect is evaluated. One could do this by randomly selecting the constant factors, but in practice there are more efficient ways to proceed, which will not be discussed here [28, 39]. The leading result of such a process is a coefficient $\mu_{i,j}$, which characterizes the effect of the ith factor on the jth model output. This coefficient may be positive or negative, and it is basically the rate of change of the output with respect to the factor, averaged over all possible values of the other factors. That is, μ is a global, and not a local, characteristic. This procedure leads to a "ranking" of the factors in terms of their influence on the model outputs. An example of ranking applied to the example discussed above is shown in figure 5. These data will not be discussed in detail, but inspection will show that the results are not intuitively obvious. This kind of ranking is, of course, distinct from principal pathway analysis [21, 26]. For instance, elastic collisions of electrons with helium appear in the ranking of figure 5. This reaction clearly cannot appear in any relevant reaction pathway: But it affects the uncertainty by influence on the electron energy balance.

5. Validation

Validation aims to test the physical fidelity of the mathematical model that is solved by a computer program. This commonly means that the model predictions are to be compared with experimental measurements. Clearly, for this to be a meaningful activity, the computer program that solves the mathematical model must have been verified (section 2), the particular calculations executed for comparison purposes must have been verified (section 2), and an uncertainty quantification must be available (section 3). When these conditions are satisfied, we are unambiguously testing the model, and not either the computer program, or the choice of numerical parameters. Moreover, since the uncertainty quantification associates error bars with the model

predictions, and the experimental measurements also (presumably) have error bars, there is a clear criterion for agreement between the model and the experiments. Most writers on the subject argue that validation is most effective as a collaborative procedure between experimentalists and simulationists [29]. Among other reasons, this is because few experimental reports in the literature are so detailed that all the parameters required by a simulation are clearly specified. If the simulationist is required to estimate (guess) some parameters, the value of the exercise is degraded. Furthermore, an effective collaboration between experimentalists and simulationists should lead to mutual criticism of a constructive kind, that may improve both the simulation and the experiments. Examples of the difficulties that may reduce the value of experiments as validation targets are discussed by Golda et al. [8].

A special problem of validation occurs when the question of optimising a complex model arises. When a model contains uncertain parameters, varying such parameters within their margin of uncertainty to improve the fit of a model to experiments is a legitimate procedure. Uncertainty quantification and sensitivity analysis can be used to guide such a process. The combustion community tends to use the term "reaction mechanism" to describe a model that has passed through this process of validation and optimisation. A well-known and important example is "GRI Mech 3.0," a model for natural gas combustion containing some 325 reactions [41]. GRI Mech 3.0 is optimised using an automated procedure [7] against approximately 100 "targets." A target is essentially a particular validation experiment. Clearly, the development of a reaction mechanism in this sense entails a commitment to a careful and detailed experimental study of the system of interest, in order to establish such targets. There are few examples of sufficiently detailed studies in low-temperature plasma physics. Studies of the chlorine system represent a recent initiative in this direction [2, 40, 25], but remain a work in progress.

The low-temperature plasma physics community probably does not have any model that counts as a "reaction mechanism" in the sense discussed above. One can find examples of careful collation of rate constants [17, 48], uncertainty quantification [48], sensitivity analysis [49], validation against experimental targets [9], and limited optimisation [30], but none of these works combines all the features discussed above in a systematic fashion. But there are chemistries of probably sufficient importance to justify the exertion required, such as the mixtures of rare and atmospheric gases that have become important in biomedical applications of plasmas, and certain chemistries of interest for microelectronic manufacturing.

An important sub-field of low-temperature plasma physics in which validation has been pursued vigorously is swarm physics. The practice of testing suites of cross sections by computing transport coefficients for comparison with experimental data has been established since the nineteen-sixties. Even in this area, however, there might be advantage to applying some of the techniques discussed above.

6. Discussion

The preceding sections show that the development of a simulation using best practice, as understood by, for example, the computational fluid dynamics community, follows this procedure:

(i) Formulation of a mathematical model
(ii) Development of a code to compute solutions of the model
(iii) Verification of the code, against exact (including manufactured) solutions of the mathematical model
(iv) Specification of a protocol for verification of calculations using the code
(v) Uncertainty quantification with respect to the physical parameters of the mathematical model
(vi) Validation of the mathematical model by reference to experiments
(vii) Optimisation of the physical parameters with respect to "target" experiments

These steps aim towards a mature predictive capability. In practice, as we have seen, in the context of low-temperature plasma physics, there are difficulties at almost every point. For instance, the structure of hybrid models, with no formal mathematical basis, and diverse (possibly incommensurate) numerical parameters, makes verification of either code or calculations difficult (perhaps infeasible) according to these principles. Other important categories of code in wide use in low-temperature plasma physics are also difficult to verify using the techniques recommended in other fields. Clearly, there are problems here requiring research. Uncertainty quantification, sensitivity analysis, and optimisation are all possible, but have not been widely used. It is doubtful whether sufficient data of appropriate quality exist to either validate or optimize any particular chemistry, certainly not to the degree seen in GRI-Mech 3.0, for example. None of these obstacles is obviously insurmountable, but collectively they are a formidable challenge.

In this context, the resources required for verification, validation, and uncertainty quantification are a relevant consideration. There is no doubt that the procedures discussed above greatly increase the resources required to carry out computer simulations. This is true at several levels. Verification requires both the identification of suitable test solutions, and demonstration that the computer code under test converges to these solutions in the expected way. A code convergences to the expected solution at the expected rate only if the implementation is accurate even in subtle details to a degree that is difficult to achieve. For instance, in a finite difference procedure, an error at a single grid point is enough to reduce the order of convergence, and such mistakes are easy to make but difficult to find. The verification procedure will show that there is an error in the code, but not where it is to be found. So verification greatly increases the human resource required to develop a code. Verification of calculations is likely to be easier, because this step can be automated, and usually will be done by coarsening the grid (or other numerical parameters), which need not greatly increase the computational

expense. Uncertainty quantification, however, even with an efficient procedure, is likely to require the examination of around ten cases per uncertain parameter, which is a serious increase in computational effort, especially when hundreds or thousands of parameters are involved. Similarly, validation and optimisation requires some mixture of critical evaluation of the existing literature and (preferably) design and execution of fresh experiments to build a set of validation "targets." From these considerations, it will be clear that the expense (in any of several senses) of pursuing predictive simulation through verification and validation is likely to be large. Indeed the factor involved may be two or three orders of magnitude beyond the effort required to produce a simulation at the minimal level with a code that does not (usually) crash, and with results that appear reasonable to an expert judge (the traditional validity criterion for physicists).

We assume that simulations are always carried out in order to substantiate certain conclusions. The important point is then to demonstrate that the conclusions are not in doubt because of error or uncertainty in the simulations. How much of the programme outlined above must be delivered to support this demonstration will vary, depending on the context. If the simulations were done as a conceptual crutch for an analytic model, which will eventually stand by itself, then possibly none of the programme needs to be delivered. If the simulations alone will inform engineering design decisions with important human or economic consequences, then it may be imperative to deliver the whole programme. This is a matter of risk management. Well-known epistemological considerations show that no categorical guarantee can ever be given against predictive failure. But the risk of such a failure can be minimized by best practice.

7. Concluding Remarks

The low-temperature plasma physics community has achieved major advances in simulation capability in recent decades. These advances have, in general, been towards greater realism in simulation. We now have simulations with detailed representations of complex geometry combined with rich physical and chemical models. The drift of this work is from scientific exploration and towards engineering prediction. Against this background, and in view of the discussion above, it may be timely to consider the methodological changes that have occurred in other fields, and their potential application in low-temperature plasma science. As we have seen, other fields have made important advances that have contributed substantially to the acceptance of computer simulation as a predictive tool. These techniques are undoubtedly expensive to apply, which suggests that individual investigators should consider carefully the consequences for their research goals of adopting, or not adopting, such methods. The experience of other fields shows that it is reasonable to question the significance of computational results that ignore issues of verification and validation.

The community might consider collective actions. Possibly the development of an elaborate code with full verification is beyond the resources of any single research group. Similarly, developing "reaction mechanisms" for important chemistries may be

a goal more appropriately pursued collectively than individually. By these means we may achieve a significantly more powerful computer simulation capability than we enjoy today.

References

[1] Charles K. Birdsall and A. Bruce Langdon. *Plasma Physics via Computer Simulation*. Adam Hilger, 1991. ISBN 978-0-7503-0117-6.

[2] J. P. Booth, Y. Azamoum, N. Sirse, and P. Chabert. Absolute atomic chlorine densities in a Cl 2 inductively coupled plasma determined by two-photon laser-induced fluorescence with a new calibration method. *J. Phys. D: Appl. Phys.*, 45 (19):195201, 2012. ISSN 0022-3727.

[3] Geoffrey Chang, Christopher B. Roth, Christopher L. Reyes, Owen Pornillos, Yen-Ju Chen, and Andy P. Chen. Retraction. *Science*, 314(5807):1875b–1875b, December 2006. ISSN 0036-8075, 1095-9203.

[4] D. R. Mark Chapman. Computers vs. wind tunnels for aerodynamic flow simulations. *Astronautics and Aeronautics*, April 1975.

[5] B. D. Dudson, J. Madsen, J. Omotani, P. Hill, L. Easy, and M. Løiten. Verification of BOUT++ by the method of manufactured solutions. *Physics of Plasmas (1994-present)*, 23(6):062303, June 2016. ISSN 1070-664X, 1089-7674.

[6] Christopher J. Freitas. Editorial. *Journal of Fluids Engineering*, 115(3):339–340, September 1993. ISSN 0098-2202.

[7] Michael Frenklach, Hai Wang, and Martin J. Rabinowitz. Optimization and analysis of large chemical kinetic mechanisms using the solution mapping method—combustion of methane. *Progress in Energy and Combustion Science*, 18(1):47–73, January 1992. ISSN 0360-1285.

[8] J. Golda, J. Held, B. Redeker, M. Konkowski, P. Beijer, A. Sobota, G. Kroesen, N. St J. Braithwaite, S. Reuter, M. M. Turner, T. Gans, D. O'Connell, and V. Schulz-von der Gathen. Concepts and characteristics of the 'COST Reference Microplasma Jet'. *J. Phys. D: Appl. Phys.*, 49(8):084003, 2016. ISSN 0022-3727.

[9] B.F. Gordiets, C.M. Ferreira, V.L. Guerra, J.M.A.H. Loureiro, J. Nahorny, D. Pagnon, M. Touzeau, and M. Vialle. Kinetic model of a low-pressure N2-O2 flowing glow discharge. *IEEE Transactions on Plasma Science*, 23(4):750–768, August 1995. ISSN 0093-3813.

[10] L. Hatton. The T experiments: errors in scientific software. *IEEE Computational Science and Engineering*, 4(2):27–38, April 1997. ISSN 1070-9924.

[11] L. Hatton. The Chimera of Software Quality. *Computer*, 40(8):104–103, August 2007. ISSN 0018-9162.

[12] W. N. G. Hitchon, D. J. Koch, and J. B. Adams. An efficient scheme for convection-dominated transport. *J. Comp. Phys.*, 83:79–95, 1989.

[13] Bernt Jakobsen. The Sleipner accident and its causes. *Engineering Failure Analysis*, 1(3):193–199, October 1994. ISSN 1350-6307.

[14] Temesgen Kindo. Verify Simulations with the Method of Manufactured Solutions. `http://www.comsol.com/blogs/verify-simulations-with-the-method-of-manufactured-so` 2015-07-27T04:56:57+00:00.

[15] J. C. Knight and N. G. Leveson. An experimental evaluation of the assumption of independence in multiversion programming. *IEEE Transactions on Software Engineering*, SE-12(1):96–109, January 1986. ISSN 0098-5589.

[16] Patrick Knupp and Kambiz Salari. *Verification of Computer Codes in Computational Science and Engineering*. Chapman and Hall/CRC, Boca Raton, Fla.; London, 1 edition edition, October 2002. ISBN 978-1-58488-264-0.

[17] I. A. Kossyi, A. Yu Kostinsky, A. A. Matveyev, and V. P. Silakov. Kinetic scheme of the non-equilibrium discharge in nitrogen-oxygen mixtures. *Plasma Sources Sci. Technol.*, 1(3):207, 1992. ISSN 0963-0252.

[18] Mark J. Kushner. Hybrid modelling of low temperature plasmas for fundamental investigations and equipment design. *J. Phys. D: Appl. Phys.*, 42(19):194013, 2009. ISSN 0022-3727.

[19] Ty D. Laurie and Mark R. Becker. Responsible Use of Computer Technology by Engineers. *Journal of Professional Issues in Engineering Education & Practice*, 130(2):128–133, April 2004. ISSN 10523928.

[20] J. E. Lawler and U. Kortshagen. Self-consistent monte carlo simulations of the positive column of gas discharges. *J. Phys. D: Appl. Phys.*, 32(24):3188, 1999.

[21] Ralph Lehmann. An Algorithm for the Determination of All Significant Pathways in Chemical Reaction Systems. *Journal of Atmospheric Chemistry*, 47(1):45–78, January 2004. ISSN 0167-7764, 1573-0662.

[22] G. Lin and G. E. Karniadakis. Sensitivity analysis and stochastic simulations of non-equilibrium plasma flow. *Int. J. Numer. Meth. Engng.*, 80(6-7):738–766, November 2009. ISSN 1097-0207.

[23] S Longo. Direct derivation of Skullerud's Monte Carlo method for charged particle transport from the linear Boltzmann equation. *Physica A: Statistical Mechanics and its Applications*, 313(3–4):389–396, October 2002. ISSN 0378-4371.

[24] Shankar Mahadevan. Verification strategies for fluid-based plasma simulation models. In *65th Annual Gaseous Electronics Conference*, page AM3.002, Austin, Texas, U. S. A., October 2012.

[25] Daniil Marinov, Mickaël Foucher, Ewen Campbell, Mark Brouard, Pascal Chabert, and Jean-Paul Booth. High sensitivity ultra-broad-band absorption spectroscopy of inductively coupled chlorine plasma. *Plasma Sources Sci. Technol.*, 25(3):035019, 2016. ISSN 0963-0252.

[26] A. H. Markosyan, A. Luque, F. J. Gordillo-Vázquez, and U. Ebert. Pumpkin: A

Tool to Find Principal Pathways in Plasma Chemical Models. *Computer Physics Communications*, 185(10):2697–2702, October 2014. ISSN 0010-4655.

[27] Zeeya Merali. Computational science: ...Error. *Nature News*, 467(7317):775–777, October 2010. ISSN 0028-0836.

[28] Max D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.

[29] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, October 2010. ISBN 978-1-139-49176-1.

[30] J. M. Pouvesle, A. Khacef, J. Stevefelt, H. Jahani, V. T. Gylys, and C. B. Collins. Study of two-body and three-body channels for the reaction of metastable helium atoms with selected atomic and molecular species. *Journal of Chemical Physics*, 88(5):3061, 1988/03//3/1/1988. ISSN 00219606.

[31] Gregg A. Radtke, Keith L. Cartwright, and Lawrence C. Musson. Stochastic Richardson Extrapolation Based Numerical Error Estimation for Kinetic Plasma Simulations. Sandia Report SAND2015-8620, Sandia National Laboratories, Albuquerque, New Mexico 87185, U. S. A., October 2015.

[32] Ivan D. Reid. Corrigendum: An investigation of the accuracy of numerical solutions of boltzmann's equation for electron swarms in gases with large inelastic cross sections. *Aust. J. Phys.*, 35(4):473–474, 1982.

[33] D. Reiter, B. Küppers, and R. K. Janev. Hydrocarbons in edge plasmas: A sensitivity analysis. *Phys. Scr.*, 2009(T138):014014, 2009. ISSN 1402-4896.

[34] Patrick J. Roache. Code Verification by the Method of Manufactured Solutions. *J. Fluids Eng*, 124(1):4–10, November 2001. ISSN 0098-2202.

[35] Patrick J. Roache. *Fundamentals of Verification and Validation*. Hermosa Publishers, Socorro, NM, September 2009. ISBN 978-0-913478-12-7.

[36] Patrick J. Roache, Kirti N. Ghia, and Frank M. White. Editorial Policy Statement on the Control of Numerical Accuracy. *J. Fluids Eng*, 108(1):2–2, March 1986. ISSN 0098-2202.

[37] S. J. Rose. The radiative opacity at the Sun centre—a code comparison study. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 71(2–6):635–638, October 2001. ISSN 0022-4073.

[38] Kambiz Salari and Patrick Knupp. Code verification by the method of manufactured solutions. SAND2000-1444, Sandia National Laboratories, Albuquerque, New Mexico, U. S. A., 2000.

[39] Andrea Saltelli, Marco Ratto, Stefano Tarantola, and Francesca Campolongo. Sensitivity Analysis for Chemical Models. *Chem. Rev.*, 105(7):2811–2828, July 2005. ISSN 0009-2665.

[40] N. Sirse, J. P. Booth, P. Chabert, A. Surzhykov, and P. Indelicato. Chlorine atom densities in the $(3p^5)^2p^0_{1/2}$ excited spin–orbit state measured by two-photon

absorption laser-induced fluorescence in a chlorine inductively coupled plasma. *J. Phys. D: Appl. Phys.*, 46(29):295203, 2013. ISSN 0022-3727.

[41] Gregory P. Smith, David M. Golden, Michael Frenklach, Nigel W. Moriarty, Boris Eiteneer, Mikhail Goldenberg, C. Thomas Bowman, Robert K. Hanson, Soonho Song, William C. Gardiner, Jr, Vitali V. Lissianski, and Zhiwei Qin. GRI-Mech 3.0. `http://combustion.berkeley.edu/gri-mech/`, 2000.

[42] Avneet Sood, R. Arthur Forster, and D. Kent Parsons. Analytical benchmark test set for criticality code verification. *Progress in Nuclear Energy*, 42(1):55–106, 2003. ISSN 0149-1970.

[43] M. Surendra. Radiofrequency discharge benchmark model comparison. *Plasma Sources Science and Technology*, 4(1):56, 1995. ISSN 0963-0252.

[44] The Editors. Editorial: Uncertainty Estimates. *Physical Review A*, 83(4), April 2011. ISSN 1050-2947, 1094-1622.

[45] T. G. Trucano, L. P. Swiler, T. Igusa, W. L. Oberkampf, and M. Pilch. Calibration, validation, and sensitivity analysis: What's what. *Reliability Engineering & System Safety*, 91(10–11):1331–1357, October 2006. ISSN 0951-8320.

[46] Timothy Guy Trucano, Martin M. Pilch, and William Louis Oberkampf. On the role of code comparisons in verification and validation. Technical Report SAND2003-2752, 918244, Sandia National Laboratories, August 2003.

[47] M. M. Turner, A. Derzsi, Z. Donkó, D. Eremin, S. J. Kelly, T. Lafleur, and T. Mussenbrock. Simulation benchmarks for low-pressure plasmas: Capacitive discharges. *Physics of Plasmas (1994-present)*, 20(1):013507, January 2013. ISSN 1070-664X, 1089-7674.

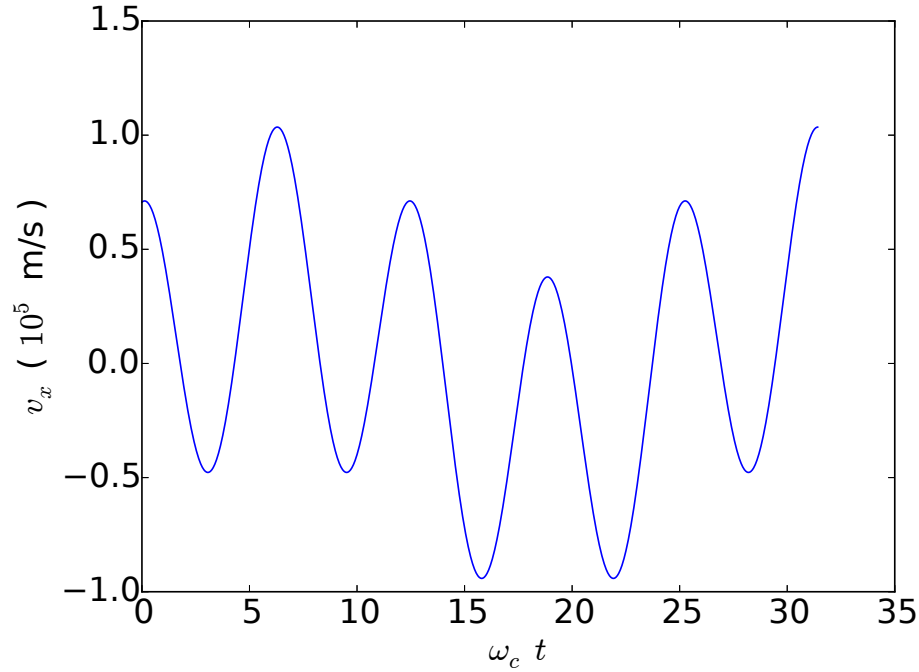
[48] Miles M. Turner. Uncertainty and error in complex plasma chemistry models. *Plasma Sources Sci. Technol.*, 24(3):035027, 2015. ISSN 0963-0252.

[49] Miles M. Turner. Uncertainty and sensitivity analysis in complex plasma chemistry models. *Plasma Sources Sci. Technol.*, 25(1):015003, 2016. ISSN 0963-0252.

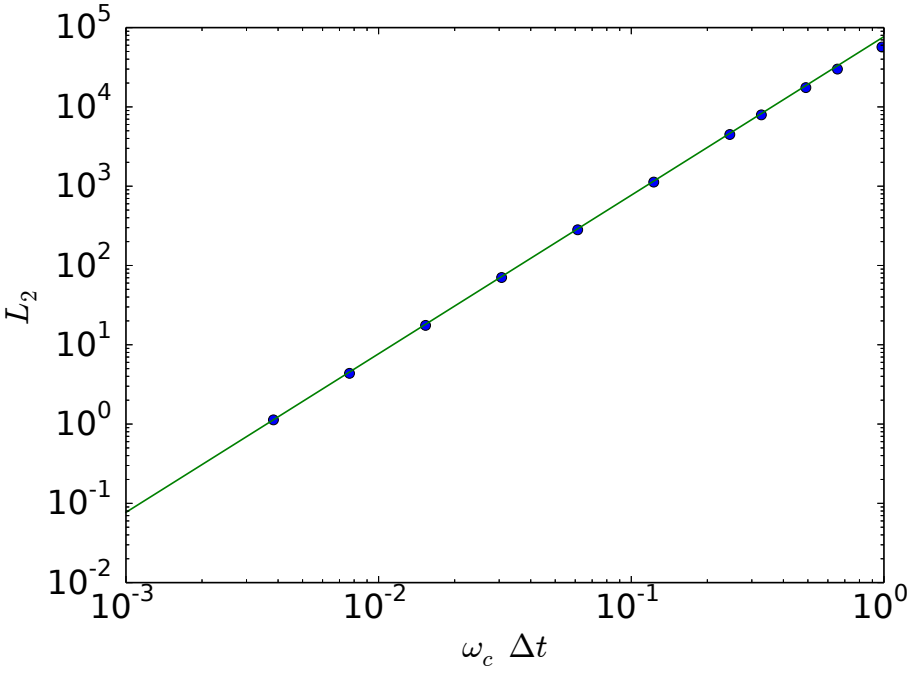
[50] Wolfgang Wagner. A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation. *J Stat Phys*, 66(3-4):1011–1044, 1992. ISSN 0022-4715, 1572-9613.

[51] Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, and Paul Wilson. Best Practices for Scientific Computing. *PLOS Biol*, 12(1):e1001745, January 2014. ISSN 1545-7885.

Appendix A. Table of contents



(a)



(b)

Figure 1. A correctness demonstration showing convergence of the trajectory of a charged particle moving in crossed electric and magnetic fields towards an exact solution. In this example, the magnetic field is stationary, while the electric field oscillates in time with frequency $\omega = \omega_c/4$, where ω_c is the cyclotron frequency. The upper panel shows one component of the exact particle trajectory, while the lower panel shows the L_2 norm relative to this exact solution as a function of the integration step Δt. The solid line shows the expected result $L_2 \propto \Delta t^2$, while the points are computed results.

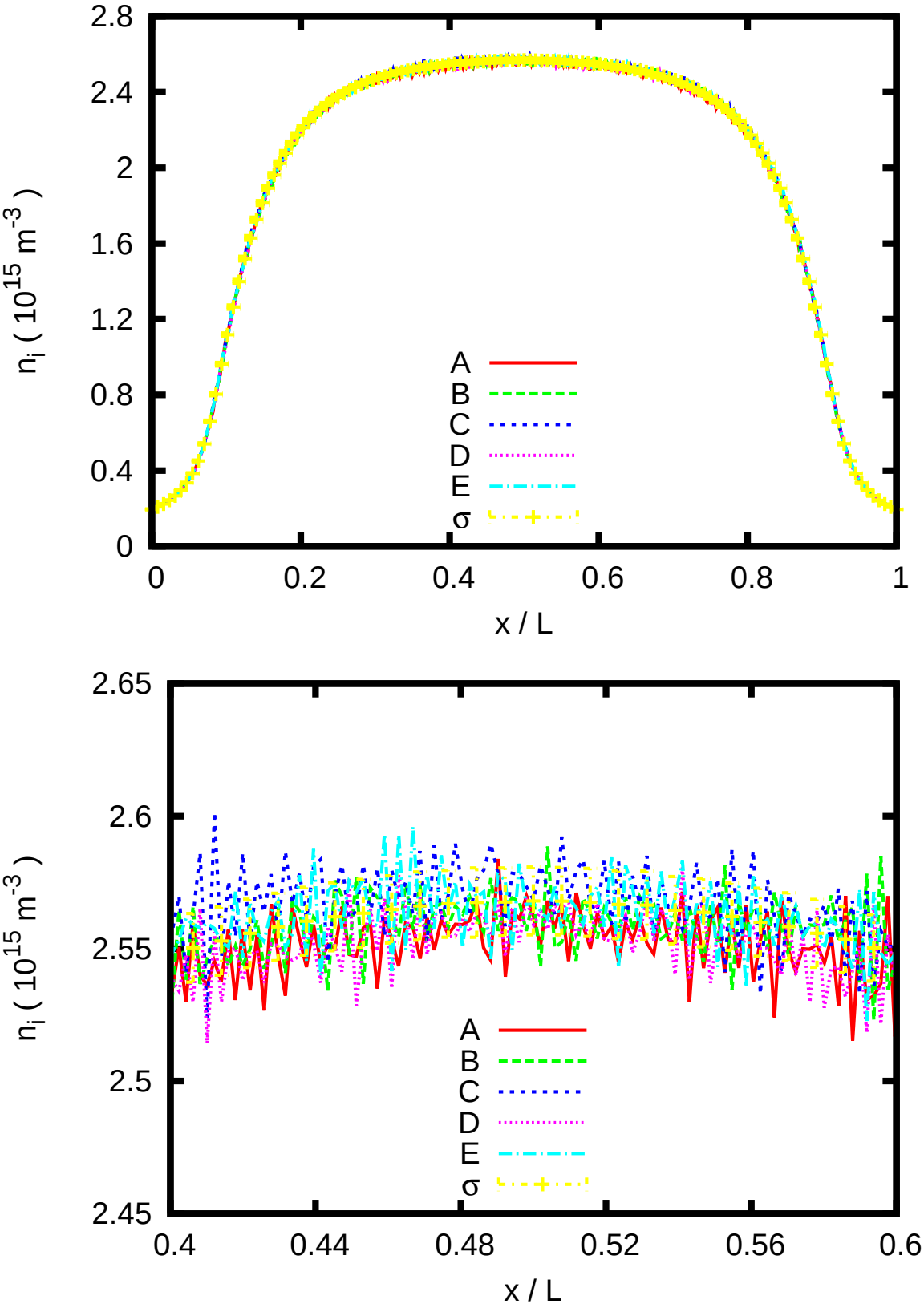


Figure 2. An example of a verification benchmark from Turner et al. [47]. This benchmark refers to a one-dimensional capacitively coupled discharge in helium at a pressure of 300 mTorr. Simulations have been carried out using five independently written particle-in-cell simulation codes for the same physical conditions and numerical parameters. The data shown here are for the ion density averaged with respect to time. The error bars are statistical. The five codes tested here agree within the error bars, and it can be shown that the simulation results are statistically indistinguishable. The lower panel is an expanded view of the central region.

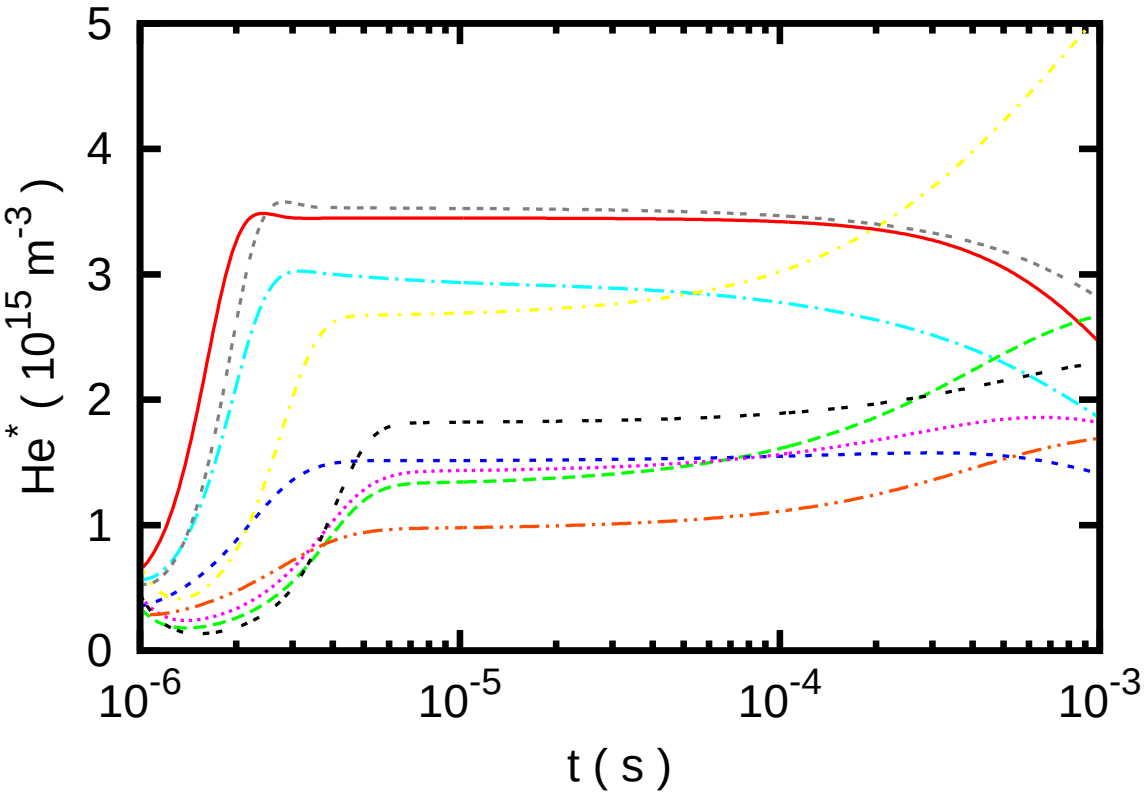


Figure 3. These trajectories of helium metastable density as a function of time were obtained using a Monte Carlo procedure that chooses a random set of rate constants in a manner consistent with the associated uncertainties [48]. Each trajectory thus represents a solution for the same physical conditions, using different randomly chosen rate constants. The set of trajectories therefore show a range of possible outcomes, all consistent with the state of knowledge of the rate constants. Clearly, the trajectories exhibit both quantitative and qualitative differences, to a degree suggesting that comparison with experiments should proceed with great caution.

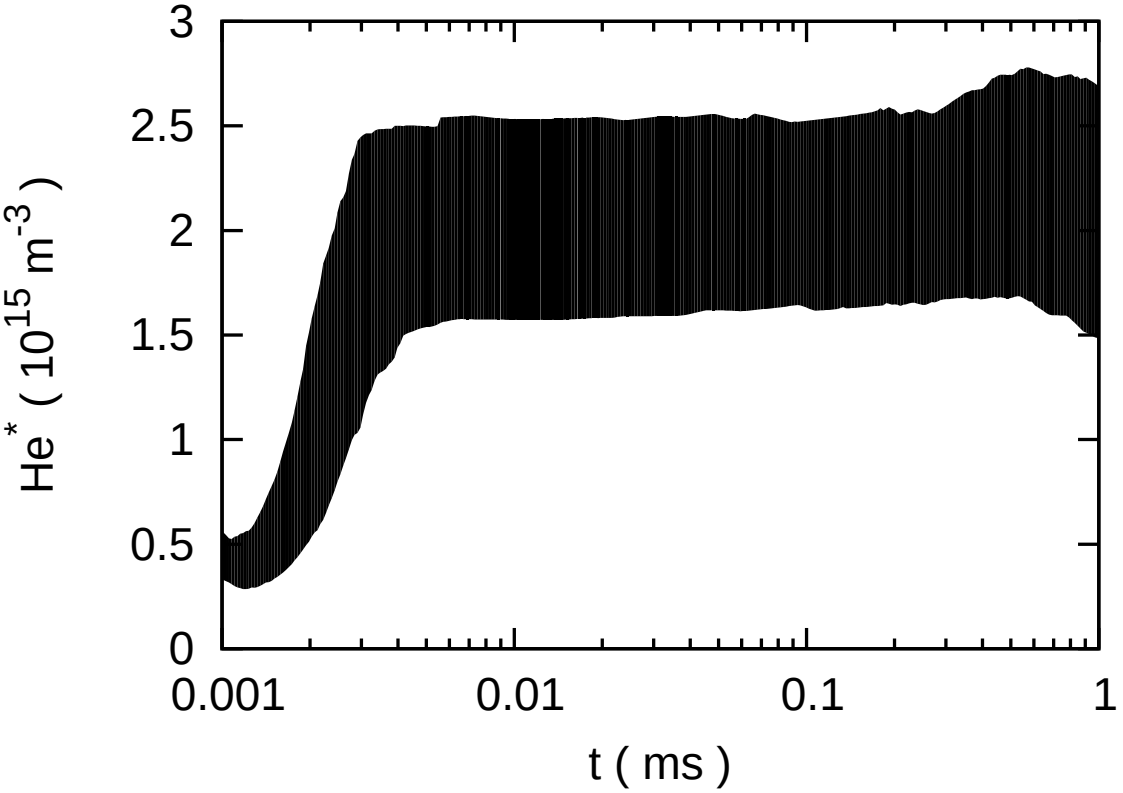


Figure 4. The helium metstable density as a function of time in a model of a helium-oxygen plasma, as in figure 3. In this representation, the shaded region is bounded by the upper and lower quartile values of the distribution of trajectories at each time point. These data therefore indicate the magnitude of the uncertainty in the computed density, due to the uncertainty in the rate constant data included in the chemistry model.

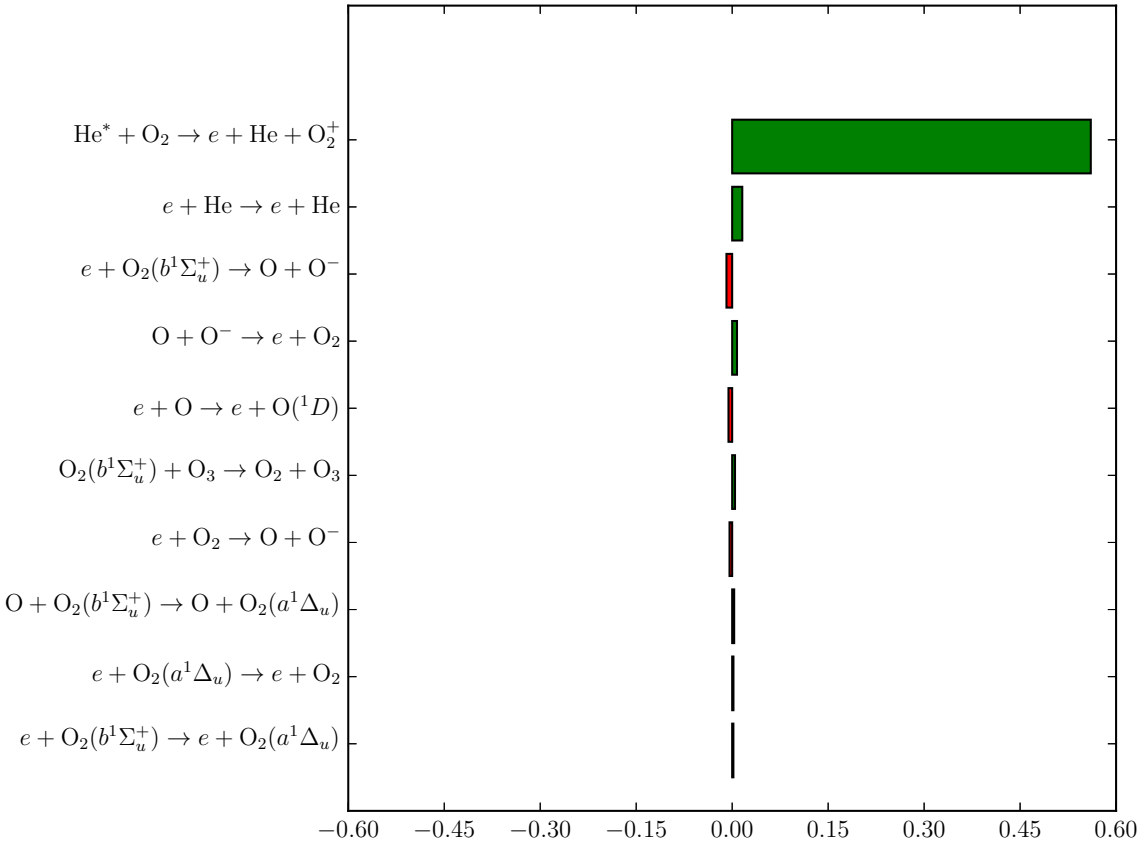
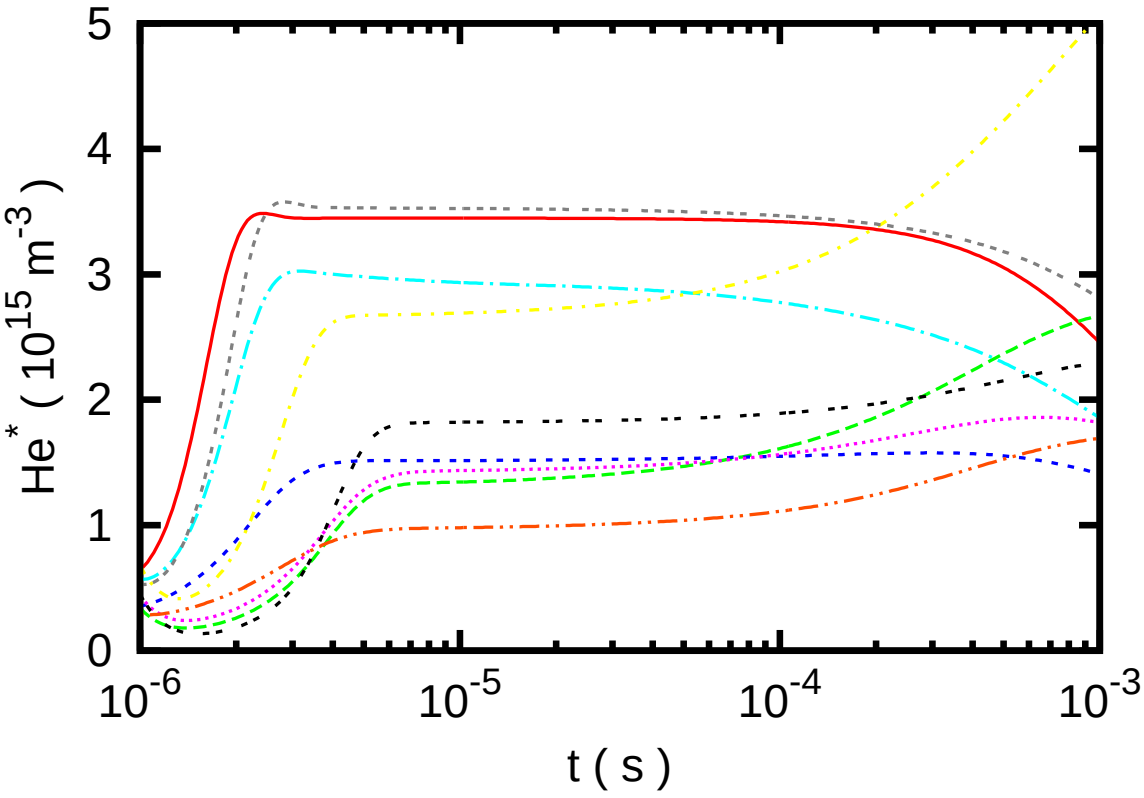


Figure 5. A ranking of rate constants by their influence on the uncertainty in the density of helium metastables, evaluated using the Morris method [28, 49]. The horizontal axis represents the coefficient μ discussed in the text.


5
4
3
2
1
0
He^* (10^{15} m^{-3})
10^{-6}
10^{-5}
10^{-4}
10^{-3}
t (s)