NPL Report CMSC 39/04

Report to the National Measurement System Directorate, Department of Trade and Industry

Visualisation, Uncertainties, and Continuous Modelling

L Wright

February 2004

NPL Report CMSC 39/04

Visualisation, Uncertainties, and Continuous Modelling

L Wright
Centre for Mathematics and Scientific Computing

February 2004

ABSTRACT

This report addresses issues connected with uncertainties and visualisation in continuous modelling. Many continuous modelling packages, particularly finite element (FE) and continuous fluid dynamics (CFD) packages, produce impressive graphical outputs. These outputs can easily lull the viewer into a false sense of security, and lead the viewer to forget that every measurement should have an associated uncertainty.

The need for uncertainties to be associated with model results and visualisations of results leads to two questions that need to be asked for each visualisation:

- What is the uncertainty associated with the visualisation process?
- How can uncertainties derived from continuous models be visualised most effectively?

This report aims to answer these questions. Some of the points made are illustrated in a case study derived from a thermal metrology problem.

ISSN 1471-0005

National Physical Laboratory
Queens Road, Teddington, Middlesex, TW11 0LW

Approved on behalf of the Managing Director, NPL
by Dave Rayner, Head of the Centre for Mathematics and Scientific Computing

Contents

1 Introduction

This report addresses issues connected with uncertainties and visualisation in continuous modelling. Many continuous modelling packages, particularly finite element (FE) and continuous fluid dynamics (CFD) packages, produce impressive graphical outputs. These outputs can easily lull the viewer into a false sense of security, and lead the viewer to forget that every measurement should have an associated uncertainty.

The need for uncertainties to be associated with model results and visualisations of results leads to two questions that need to be asked for each visualisation:

- What is the uncertainty associated with the visualisation process?
- How can uncertainties derived from continuous models be visualised most effectively?

This report aims to answer these questions. Some of the points made are illustrated in a case study derived from a thermal metrology problem.

The report does not give advice on evaluating the uncertainties associated with continuous models. Advice on this topic can be found in two SS*f*M reports on uncertainties in continuous modelling [1] and model validation in continuous modelling [2]. These reports explain different techniques for evaluating uncertainties in continuous models, and error estimation techniques that can help determine the details of an uncertainty evaluation. The only method used for uncertainty evaluation in this report is Monte Carlo simulation (MCS), which has already been described fully [1].

Users of this report are strongly recommended to consult a recent SS*f*M Good Practice Guide (GPG) on Data Visualisation [8] for further advice on the visualisation of uncertainties. The majority of the points made in the GPG regarding the visualisation of measurement data can equally be applied to the visualisation of uncertainties. In particular, the advice regarding analysing the requirements and designing the visualisation is relevant and helpful. Note, however, that the next edition of the GPG will specifically include material on visualisation of uncertainties.

Visualisation of results is hardly ever used to generate numerical values for which associated uncertainties are required. Visualisation is commonly used for the investigation of more general questions about results, such as

- Do the results “look right”?
- Where are the maxima and minima?
- How do the results from different models compare?
- Over what range of values do the results vary?
- What are the broad trends of the results?

Most of these uses are not necessarily concerned with details of numerical values, and so detailed uncertainty evaluations are not necessarily appropriate. If numerical values are important, it is more usual to take the values of interest directly from the model results rather than extract them from a visualisation. Instead of an uncertainty as it is usually defined, often the viewer needs assurance that the conclusions being drawn from the visualisation are correct. This assurance consists of two components:

- an understanding of the uncertainties associated with the model results, so that

the user is aware of how the criteria on which their conclusions are based may be varying, and

- confidence that the results are being displayed correctly, so that the conclusions drawn from the visualised results are not erroneous.

The distinction between these two components has led to the usage throughout this report of "uncertainty" to mean the numerical value of the uncertainty associated with the model results, and "confidence" to mean the user's level of certainty that the visualisation represents the model results correctly.

Viewers can gain an improved understanding of the uncertainties associated with their model results by using visualisation techniques to explore the uncertainties. This report explains how many of the visualisation techniques that are commonly used to explore the results of continuous models can equally well be used to examine the uncertainties associated with those results.

It is difficult to express the confidence that the visualisation is consistent with the model results in any meaningful numerical way, since a viewer's opinion of, and confidence in, a visualisation are necessarily subjective. With this in mind, the report aims to address these issues by offering advice on improving the user's confidence in a visualisation rather than assigning a level to that confidence.

1.1 Structure of the report.

The report consists of four main sections. Section 2 looks at the main factors influencing the accuracy of visualisations, and offers advice on how to improve users' confidence in their visualisations. It describes ways in which the display of results can be checked and sometimes improved, but does not offer a way to express the confidence, as no such method was identified during the course of this project.

Section 3 describes techniques for the visualisation of uncertainty in continuous models, so that users can have a better understanding of the uncertainties associated with model results. Techniques are illustrated with examples, or by reference to visualisations in other sections of the report.

A case study drawn from a thermal metrology problem forms section 4 of the report. The case study illustrates many of the points raised in sections 2 and 3 of the report, and demonstrates some of the results of the visualisation techniques mentioned in section 3.

Finally, the key recommendations made in the earlier sections are summarised in section 5.

2 Confidence in visualisations

Ideally, the visualisation of results should not add any further uncertainty to that inherent in the modelling process. At the visualisation stage the model results are available and should not be altered by the visualisation. If the model results can be displayed sufficiently accurately and clearly, the viewer would gain a good understanding of them and be in a position to draw reliable conclusions. However, in reality this does not always happen, as other authors have shown [12]. There are three main reasons why the results of the visualisation process can be misleading:

- Extrapolation, interpolation, and smoothing by the visualisation algorithm
- Visualisation algorithm error
- Visualisation parameter choices

These sources of potential confusion are discussed further in the remainder of this section. As mentioned in the introduction, following the guidance provided in this section should result in improved confidence in visualisation, but will not lead to a formal numerical expression of that confidence.

2.1 Extrapolation, interpolation, and smoothing

The majority of continuous modelling algorithms solve a discretised version of a continuous problem. The discretised form of a problem permits the calculation of results at a number of points within the region of interest (known as mesh points). Often it is then also necessary to calculate results at locations other than the mesh points. This issue is particularly important when the results are visualised as a contour plot because such display methods visualise a discrete set of results (the values at the mesh points) as a continuous entity. Additionally, many contour plot visualisations use smoothing or local averaging to produce a more aesthetically pleasing display. Smoothing ensures continuity of results across element boundaries, but can disguise discontinuities and erroneous values.

The concerns raised by these issues are similar to those raised when creating a line plot of discrete data: a scatter plot of discrete points imposes no assumption about the behaviour of the data, but can be difficult to interpret, joining neighbouring points with straight lines makes understanding easier but may be misleading, and a higher-order polynomial curve can give a better understanding of behaviour but imposes strong assumptions.

2.1.1 Interpolation and extrapolation

Some techniques, such as finite element methods and boundary element methods, are derived using an interpolation process to approximate the local behaviour of the solution. This feature makes it straightforward to calculate results at locations that are not mesh points. When a finite element model is developed, some functional approximation is made to describe the local behaviour of the solution. It is reasonable to assume that the solution can be visualised using the same functional approximation. Derivative quantities such as gradients, stresses, and strains can be visualised using the derivative of the approximation.

For example, if a stress analysis model is developed using linear elements, then the displacements should be visualised as linear within each element, and the stresses and strains should be constant within each element. This choice of visualisation

approximation will lead to discontinuity in the stresses and strains, but it will mean that the visualisation is consistent with the model results and does not impose any extra assumptions. Most proprietary FE packages carry out this type of visualisation automatically.

In multi-dimensional cases, the nature of the multi-dimensional interpolation functions will need careful consideration if the functions are anything other than constant or linear. Quadratic functions may require an xy term as well as terms in x^2 and y^2.

Other techniques, such as finite difference methods and finite volume methods, do not involve interpolation in their derivation, and so more careful consideration needs to be given to the display of their results.

Finite volume methods rely on the assumption of constant variable values within a small volume surrounding each mesh point, but the method is frequently further complicated by the use of different meshes for different variables. For instance, the application of a finite volume method to the Maxwell equations often uses one mesh for the magnetic field and a different, offset, mesh for the electric field. The use of two meshes means that careful interpretation is needed, particularly if some derived quantity that is a function of both the magnetic field and the electric field is of interest.

Finite difference methods use Taylor expansions to approximate derivatives of the variable of interest and hence produce a system of equations linking the variable values at the mesh points. This method assumes local continuity of the variable and some of its derivatives, but does not impose other restrictions on the values between the mesh points.

The lack of assumptions about variable behaviour made by these methods means that the best way of visualising their results initially is to regard each mesh point as being representative of some element[1] and to assume that the result is constant within that element. An example of the results of this technique is given in the case study, where the results of a finite difference model are visualised in this way.

The main potential difficulty with assuming quantities to be constant within some element is how to identify the value the quantity should take at points on the interface between two such elements. There are two cases to consider, that of continuous quantities and that of discontinuous quantities. If the quantity in question is expected to be continuous and the model is adequate, then the values within the two elements should be sufficiently close that an average of the two element values can be used. Moreover, the differences between the two element values and (hence) the average should be small. If the quantity is expected to be discontinuous at the interface between two elements, then the value on the interface is meaningless. If results on the boundary are required, both values should be quoted with an indication of which result lies on which side of the boundary.

2.1.2 Smoothing

Smoothing is usually applied to quantities that have been calculated at internal points, and involves extrapolating these quantities to the mesh points using an appropriate approximation, and then taking an appropriate local average to produce a value and local continuity.

The main disadvantage with smoothing is that it disguises discontinuities, which can

[1] "Element" here means a volume/area used for visualisation purposes, not necessarily a finite element.

lead to errors in the results being hidden by the visualisation. Sometimes a quantity, such as stress, is expected to be continuous in reality but has not been constrained to be continuous by the model. A visual check of the smoothness of the results for such a quantity can be used as a model validation method [2]. If smoothing is applied when visualising such a quantity, the results may appear to be continuous when they are not.

An example of this problem is shown in figure 1. The contour plot shown in figure 1a was plotted following local smoothing, and the results vary in a plausible manner. The contour plot in figure 1b shows the same results without contour smoothing. This plot shows the sharp discontinuity between results in adjacent elements, which enabled the user to identify and rectify an error in the model.

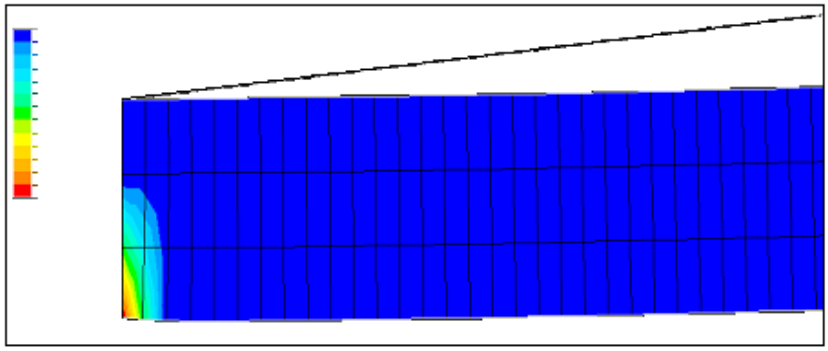

Figure 1a: Contour smoothing activated

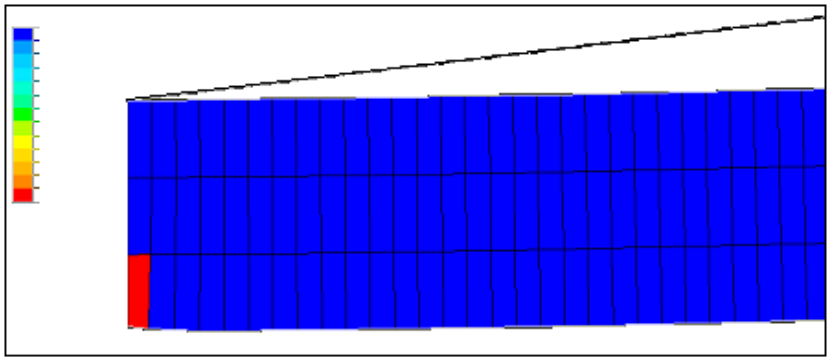

Figure 1b: Contour smoothing deactivated

Figure 1: The same results plotted using smoothed (figure 1a) and unsmoothed (figure 1b) contours (figures not plotted to same colour scale).

There are some models where discontinuities are fully expected to occur. For instance, in a stress analysis of a structure containing two materials with different elastic moduli, it is expected that the stresses will be discontinuous across any joins between the two materials. If the displayed results are to include these discontinuities, the smoothing must be deactivated.

Some packages offer a “quilt” contour plot, where a single result value is displayed for each element. This display is not necessarily identical to that that would be obtained by deactivating the smoothing: the two are only identical if the result is a derivative quantity and the original model used linear elements. The plotted value is usually an averaged value calculated from results at the integration points, and is not generally as useful as the unsmoothed contour. A quilt contour can give the impression that the results are discontinuous when they are not, and remove useful detail that an unsmoothed contour plot provides.

For the majority of well-behaved models that do not contain discontinuities in material properties or boundary conditions, the smoothed and unsmoothed results should look broadly the same since in many cases physical quantities and their gradients are

expected to be continuous. This continuity means that a visual inspection of the differences between smoothed and unsmoothed results can provide insight into the validity of the model results.

2.1.3 Recommendations

- If the derivation of the discretised problem involved approximation functions such as polynomials, visualise the results using the same type of functions as those used for interpolation.
- Use the derivatives of the functions used for the interpolation of the solution for visualisation of derivative quantities such as stresses or strains.
- If the derivation of the discretised version of the problem did not involve interpolation, assume that the results are constant within an appropriate visualisation element.
- Wherever possible, deactivate smoothing effects to check that the consequences of smoothing are not disguising unexpected discontinuities in the results.
- Compare visualisations using smoothed and unsmoothed results to increase confidence in the model results. Visualisations of a well-validated model with continuous results with and without smoothing should look broadly the same.

2.2 Algorithm error

Algorithm error can be difficult for users to identify unequivocally, particularly if a proprietary package is being used for the visualisation. Errors do not occur in a predictable way, or they would be removed before the software was released. Since visualisation packages have usually been tested prior to release using model results typical of those on which the package would normally be used, errors are most likely to occur during visualisation of models that are not standard everyday problems, i.e., including unusual or challenging features. Unfortunately, such models are often less well understood than more straightforward models, and frequently users have very little idea of how the results should look, making it difficult to detect any errors that do occur.

A probable example of algorithm error is shown in figure 2. The deformation of a cylinder and piston assembly under stress has been modelled using FE analysis. The radial variation of the piston as a function of axial distance is shown at the bottom of the plot in both cases, and that for the cylinder at the top. The image in figure 2a was produced using the post-processing package of the FE software. It purports to show how the piston and cylinder have been displaced, and it indicates that the deformed piston and cylinder overlap.

An alternative visualisation of the deformed assembly can be created by obtaining the displaced coordinates of the mesh points on the outer surfaces of the piston and cylinder directly from the FE software and plotting these coordinates with a different software package. This visualisation is shown in figure 2b. At the top of the gap the clearance between the two components is 51.5 nm, indicating that the two components do not overlap.

The experienced user who created these plots is thought to have checked all the visualisation parameters before plotting, and the plot shows a close-up view of the main region of deformation. These factors make it unlikely that this misleading display is

cased by a poor choice of scaling factor. It is believed that one of the software's visualisation algorithms is at fault. Errors are most likely to occur when plotting very small displacements of larger objects. In this case, the radius of the piston was 1.25 mm, a typical displacement was 1 µm, and the minimum separation was 51.5 nm, meaning that the various dimensions spanned eight orders of magnitude. This range of dimensions may be the source of the problem.

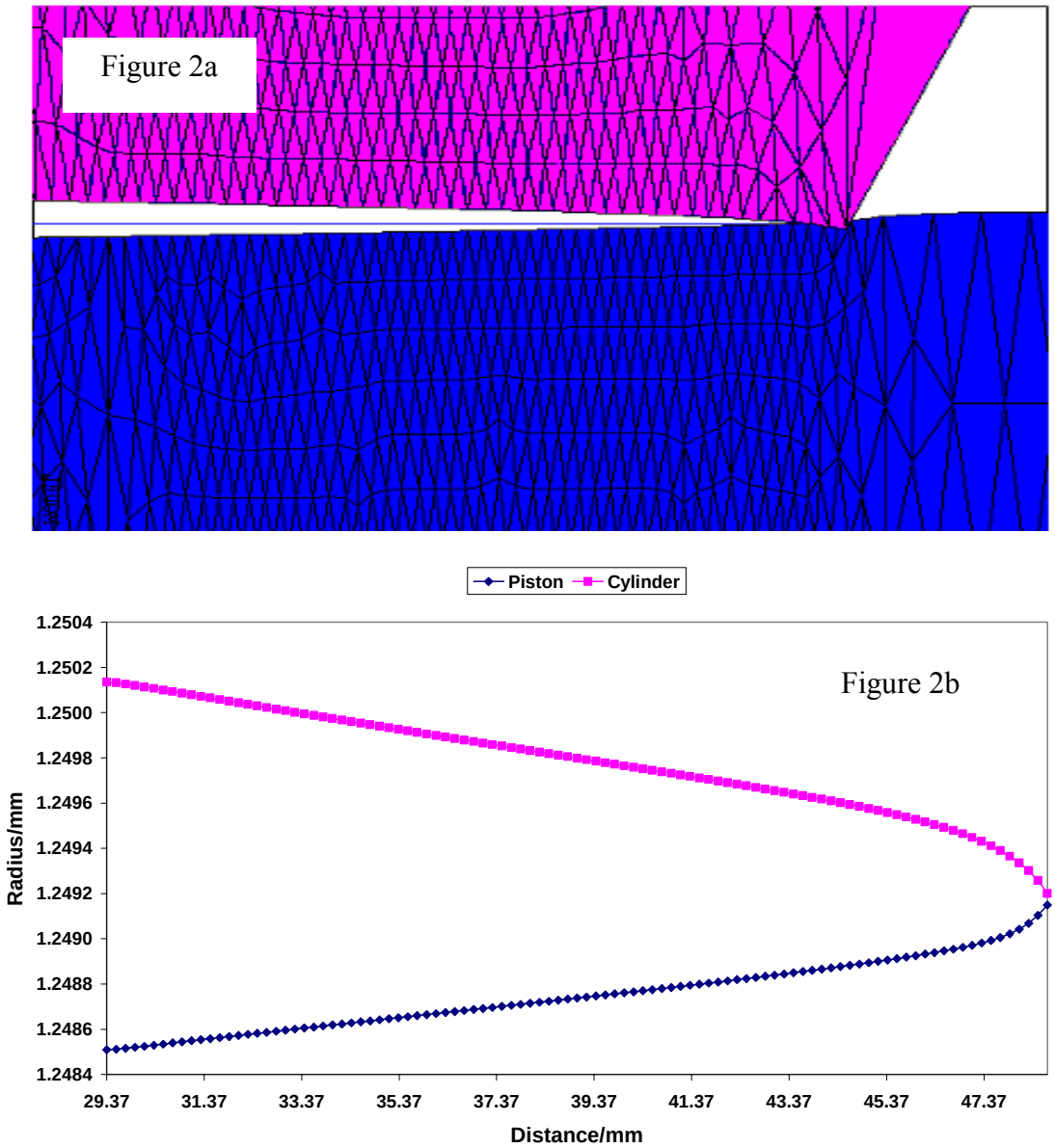

Figure 2: Deformation of a piston and cylinder visualised using two different techniques. Figure 2a shows the results of an FE model in the form of the radial variation as a function of axial distance visualised using a post-processing package, and figure 2b the same results from the same model plotted in a different package. In both cases, the radial variation for the cylinder is the upper curve, shown in pink, and that for the piston is the lower curve, shown in blue.

Many software manufacturers have online helpdesks and user forums. These facilities are often used to alert software users of newly discovered bugs and to offer patches or alternative solutions for these problems. It is useful to access these resources on a regular basis in order to keep up to date with developments. Additionally, new releases

and updates of packages often contain a list of any bugs that have been fixed, which may make re-examination of previous visualisations necessary. Conversely, if a bug is found during use of a proprietary package, it is very important to report it so that other users will be able to adjust their work accordingly.

A key step in identifying and tracing algorithm errors is definition of the algorithm that is being used. If the algorithm has been defined, potential errors are easier to identify and their likely effects on the visualisation are more fully understood. In addition, a full definition of the algorithm will enable users to ensure that the algorithm is suitable for their data.

In some cases it may not be possible to identify the visualisation algorithm, because the documentation of some visualisation packages does not include details of the algorithms. If the algorithm cannot be identified, the user may be able to carry out empirical testing or comparison with other packages. Comparison with other packages requires the existence of alternative packages that perform the same visualisation using a known algorithm. If such packages are not available, the technique may not be useful.

Empirical testing of an algorithm involves visualising many sets of data, usually with well-understood properties, to make deductions about the algorithm. For example, if an algorithm visualises data with up to n^{th} order polynomial behaviour well, it is likely that the algorithm uses n^{th} order polynomial interpolation. This technique may be unreliable because it is not exhaustive and if some feature of the algorithm only affects one type of data, the feature will only be identified if that type of data is used during testing.

The best ways to ensure that algorithm errors are absent are thorough testing before use, cross-checking during use and code review for packages in those cases where the source code is available. These techniques are commonly used in testing and validation of modelling software, and can equally well be applied to visualisation processes and software. The techniques are explained further below.

2.2.1 Testing visualisation software

The package being used for visualisation should be tested like any other piece of software used for modelling. Previous work [3] has developed a methodology for software testing, and a slightly altered version of this methodology can be applied to visualisation software. The methodology identifies reference input data and corresponding reference results, generates test results by using the reference input data in the test software, and compares the test results with the reference results using some metric. Use of this methodology therefore requires reference input data and corresponding reference results, a definition of the results to be tested, and a metric for comparison.

Reference data and results for visualisation software could take several different forms. Model results with known properties such as a linear variation of some variable could be created, with the test being “Is the known property visualised adequately?” If a different visualisation method that is known to be reliable is available a reference visualisation of some well-understood model could be generated using the alternative method and the test would be “Do the two visualisations look the same?” In the most general sense, any set of results that are available as numerical values could be regarded as reference data with the test being “Does the visualisation display the known results correctly?”

The testing of visualisation software is complicated by the fact that the interpretation of the test results is necessarily subjective, which makes it difficult to assign a metric to

the results. It is suggested that one solution to this problem is for several independent people to inspect the results and decide on whether or not the visualisation "looks right". The definition of "looks right" should be stated as part of the test documentation, and may include some or all of the following:

- Are the results varying in the manner expected?
- Are the results continuous where they should be?
- Are any discontinuities or singularities displayed correctly?
- Are the maxima and minima in the places expected?
- Do the maximum and minimum values look reasonable?

As well as being used during formal testing, users should constantly bear these points in mind as they carry out visualisation tasks as a form of ongoing informal testing.

The number n of inspections and the number m of these inspections that consider the software to have passed can be incorporated into a metric. Assuming that the better the software, the greater the metric, the metric should

- be directly proportional to m/n, since it should be proportional to the "pass rate" of the software, and
- increase to some limit as n increases and m/n is held fixed, since the metric should represent the pass rate as n increases.

There are many possible functions that will fulfil these conditions. One example is

$$f(m/n, n) = (m/n)e^{-1/n} \quad 0 \le m/n \le 1, 0 < n, \tag{1}$$

which has the above properties and generates values between 0 and 1. This function is plotted in figure 3.

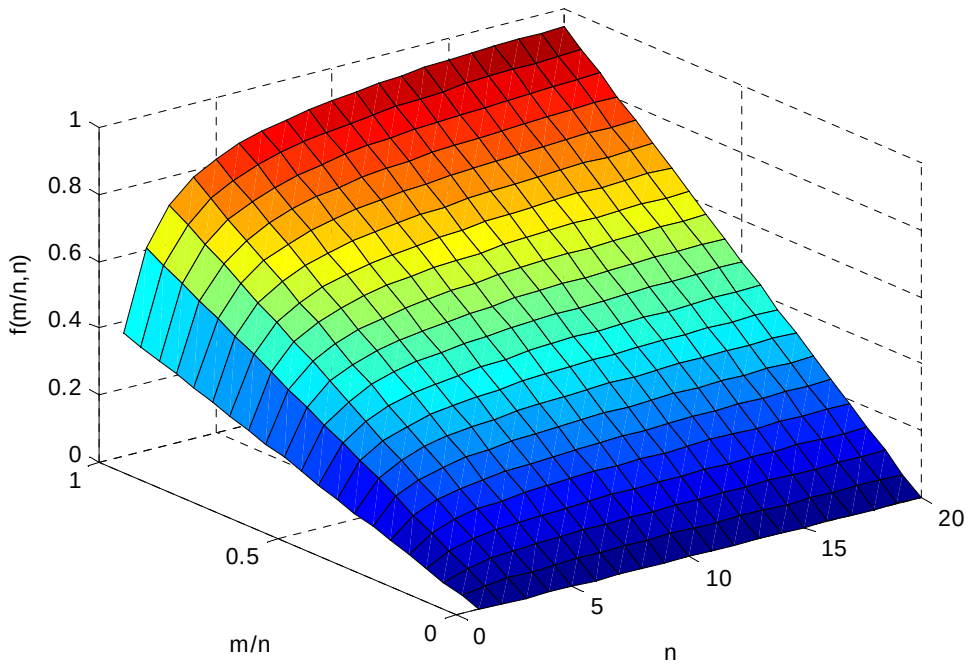

Figure 3: The metric suggested in section (1) plotted as a function of m/n and n.

Another potential difficulty is the retention of test results. It is recommended that screen shots of visualisation tests should be saved if possible, particularly in the cases where the results of the test are not clear-cut. If screen shots are saved as bitmaps, it may be possible to compare them quantitatively. Keeping such records can be useful when revisiting the tests if the software is to be used for a new type of visualisation, as well as being important for traceability purposes.

2.2.2 Cross-checking

Cross-checking of results involves comparing the visualisation, either in whole or in part, to other information such as numerical values of results or an independent visualisation using a different technique. It differs from testing because it is not carried out in a formal and documented manner to improve confidence in the software. Instead it is a continual process that improves confidence in the correctness of individual visualisations.

Cross-checking requires access to the numerical results that the package is being used to visualise. The key features of these results, such as locations and values of maxima and minima, can be extracted and used for comparisons with the visualisation. In many cases it is possible to visualise a part of the results using a different method. For example, the behaviour of the results along a line such as the outer edge of the geometry can usually be plotted as a line graph using an alternative piece of software.

Figure 2 shows a successful example of cross-checking: the erroneous overlap of the piston and the cylinder in figure 2a was identified during the visualisation by re-visualising numerical results using a different method as in figure 2b. In general, cross-checking often requires little extra effort and can help to identify errors that would otherwise be missed.

2.2.3 Code review

If the accuracy of the results generated by a piece of software is important, and the source code of the software is available, then that software should have its code reviewed. This is as true of visualisation software as it is of any other type of software. Most companies that produce their own software regularly have their own set of code review guidelines, so detailed guidance on the subject will not be given here. Broadly, the minimum aims of a code review are

- to ensure that the software will meet its specification, given any valid set of input data,
- to ensure that the code can easily be maintained and contains sufficient comment lines that it is comprehensible to people other than the author.

Generally code reviews also provide suggestions for improvement and performance enhancement, but these need to be followed if accuracy is the key concern. Code reviews should be carried out by someone other than the author of the code, who understands the programming language used, as well as having an understanding of the function of the code. Reviews should be fully documented, including details of any alterations made to the code in the light of the reviewer's comments.

Some packages used for visualisation can accept sequences of commands from a text file so that common visualisation tasks can be automated. If such a file is to be used frequently to generate visualisations it may be best to regard the sequence of commands as code and obtain a review of the file.

2.2.4 Recommendations

- Before using a proprietary package for visualisation, test it with well-understood model results.
- Wherever possible, use challenging test models (since simple cases will have been tested prior to the software's release) that are typical of the full range of intended uses of the package.
- During visualisation, carry out informal checks that the results "look right".
- Use a different package or an alternative method to visualise all or part of the results to cross-check.
- If the source code of the software used for visualisation is available, carry out a code review.
- If the package being used has a website, check any user forums, helpdesk areas, or update section regularly for updates, bug fixes, and advice.
- Report any bugs you find so that developers can improve their products.

2.3 Visualisation parameter choices

The increasing use of proprietary FE packages by people with little or no mathematical background has led to an increase in the ease of use of the post-processing and visualisation parts of such packages. The recent increase in the range of problems that FE packages can solve has led to an increase in the range of options offered by the corresponding visualisation software. These trends have meant that many packages now have a large number of visualisation parameters, many of which are set to default values of which the user may not be aware. Although the default values are generally chosen to be suitable for the majority of models, they should be checked to ensure that the visualisation is displaying what the user expects. This state of affairs is not exclusive to FE packages; it also affects more general visualisation packages.

The following examples of visualisation parameters will be discussed further below:

- choice of viewing surface and clipping planes,
- choice of local or global directions for vector and tensor quantities,
- scaling factors,
- display tolerances.

In general, the documentation for the visualisation software will include descriptions of all the parameters and their default values.

2.3.1 Choice of viewing surface

Some finite elements, for example beams with non-uniform cross sections, shell elements, and elements used for modelling laminates, may have several integration points through their thinnest dimension despite that dimension being theoretically negligible. These points are necessary to describe the through-thickness variation of the results, but since the elements are visualised as one- or two-dimensional objects, the through-thickness variation is not displayed. Instead, the user chooses the results from a single set of integration points for display. The integration points are usually identified with the upper, middle or lower surface of the structure, with the default option often

being the middle surface.

Similarly, if a user wishes to examine results from mesh points in the interior of a three-dimensional object, a clipping plane must be defined. A clipping plane defines a cross-sectional slice through the structure so that the interior of the model can be seen. The software does not display any parts of the model that lie in front of the clipping plane, so that the required results can be seen.

Both of these features enable users to view otherwise inaccessible results, but the default values can be misleading. An example is shown in figure 4. The figure shows a dynamic model constructed of shell elements. The image on the left of figure 4 shows the stresses displayed using the default choice of element surface. The default surface choice is the middle surface, which in this case is a neutral surface of the structure, and so the stress values are much smaller than might be expected and the stress distribution is not what was expected. The image on the right of figure 4 shows results from the upper surface of the same model, and shows stress values approximately 10^9 times larger than those on the left, which is more in keeping with the expected results. Similarly, the distribution of the stresses in the right-hand figure looks correct.

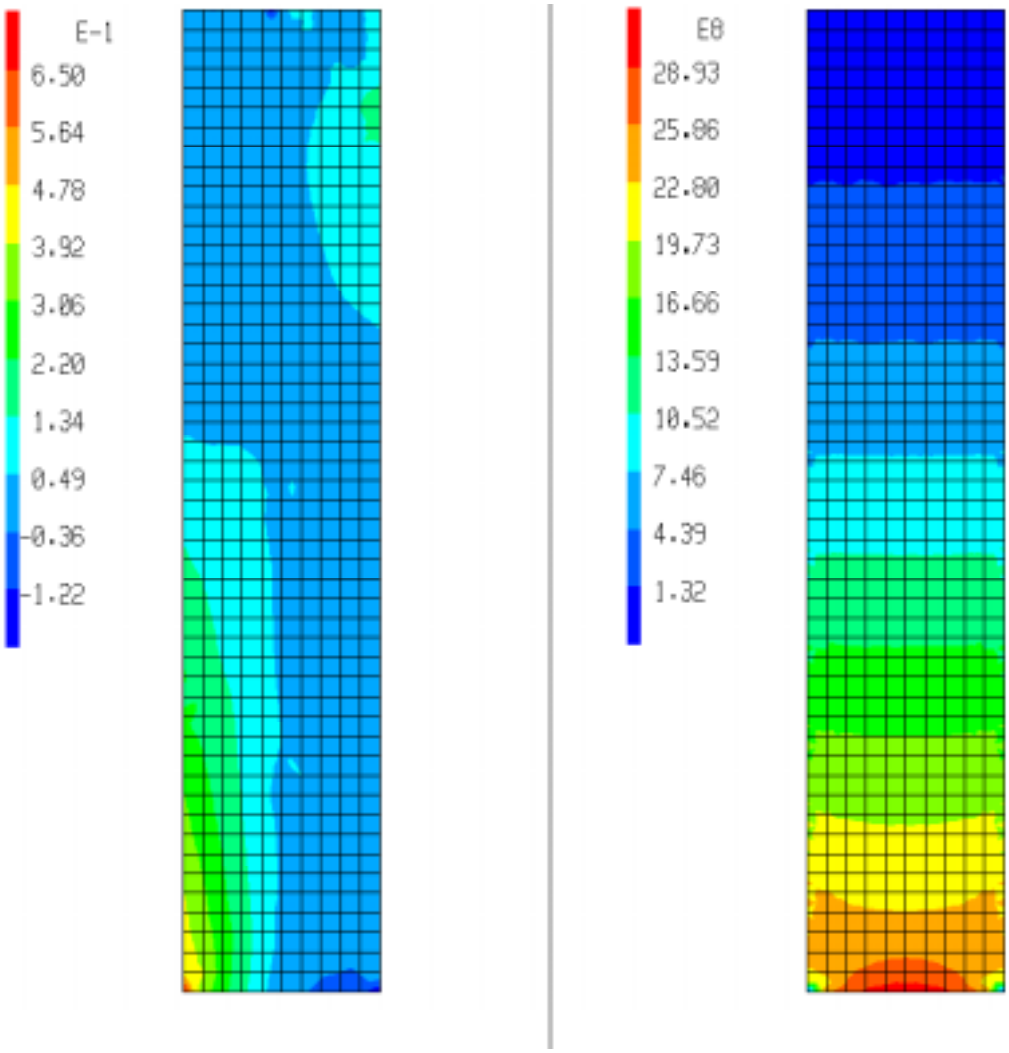

Figure 4: Stresses in a vibrating plane, displayed on the middle surface (left image) and the upper surface (right image).

This example took the user some time to rectify, as the user was unfamiliar with the behaviour of shell elements and assumed that the error was in the model rather than the

visualisation. The problem might have been avoided if the documentation had been read more thoroughly.

2.3.2 Choice of local or global directions

Many quantities calculated using continuous models are vector or tensor quantities, for example, stresses, strains, electro-magnetic fields, and wave velocities. Such quantities are often displayed as contour plots by resolving them into component form relative to some set of axes.

Many types of finite element have local axes defined for them. For example, it is common practice to take the direction lying along the length of a beam to be its local x direction, no matter which way it is oriented relative to the global axes. Additionally, when defining complicated geometries it is often convenient to define local axes, such as local curvilinear axes, in order to construct the geometry more accurately.

The existence of extra sets of axes means that there can easily be confusion as to which set of axes has been chosen for the display of vector and tensor quantities. In some cases the confusion can be cleared by plotting vectors as arrow plots, or by plotting tensors as principal values and a set of principal axes. However, not every package offers these options and they do not always clarify the situation, particularly in cases where principal axis directions change by large amounts between adjacent elements.

2.3.3 Scaling factors

Many continuous models simulate large structures that are displaced by a small amount. For instance, a probe tip may be deflected a few micrometres, but the probe itself may be hundreds of thousands of micrometres in size. This difference in scale means that if the deflection is shown to the same scale as the model geometry, there is no difference between the visualisation of the undeflected structure and the visualisation of the deflected structure.

Under such circumstances, some packages automatically scale the displayed deflections so that the displacements are large enough to be visible. This method can produce misleading images. An example is shown in figure 5. The probe, shown in its undeflected state on the left, has maximum dimensions 9 mm by 9 mm by 9 mm. The deflection of the probe tip under loading is 1.83×10^{-5} mm, and so visualising the displacements to the scale of the model geometry would not produce a sufficiently clear picture of the deflected shape. However, the displacement visualisation on the right makes the probe look seriously deformed and exaggerates the extent of the motion in comparison with the length scale of the probe itself. This effect is due to the poor choice of scaling factor.

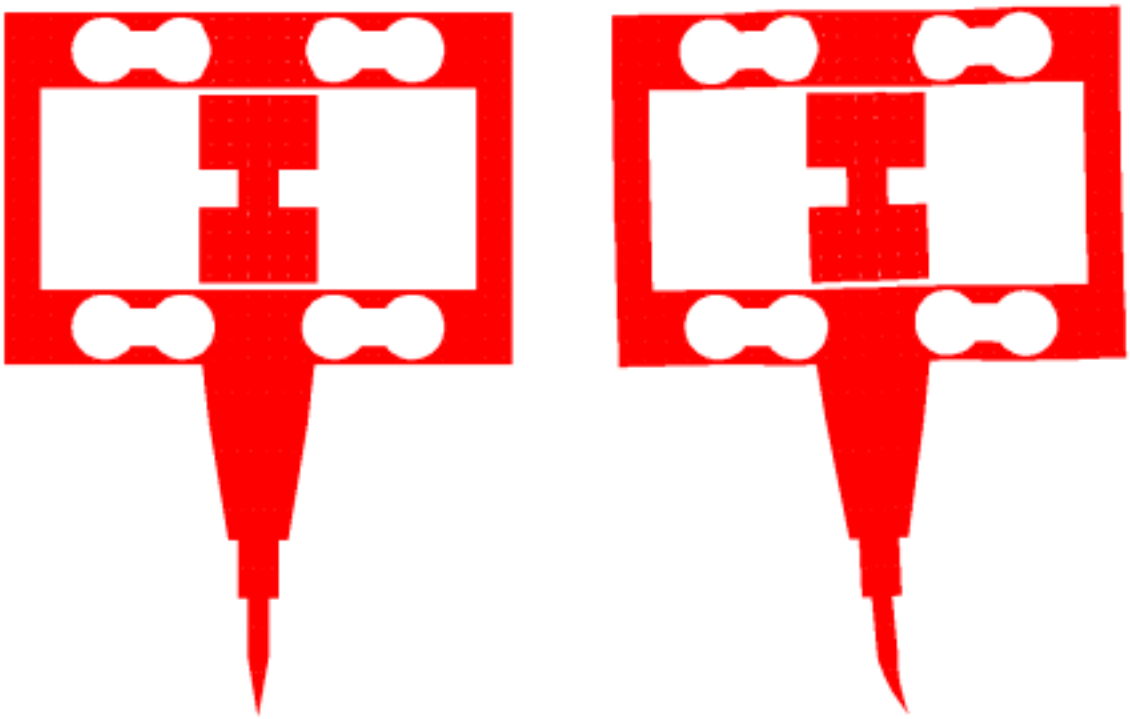

Figure 5: Visualisations of an undeflected probe (left image) and the same probe deflected under load with misleading displacement scaling.

If one particular part of the model is of interest, an alternative way of avoiding misleading scaling is to "zoom in" the view and only visualise a small detail of the model. If the visualised undeflected area measures 1 μm, it is more likely to produce a clear image of a 100 nm deflection than visualisation of the full 1 mm model.

2.3.4 Tolerances

Some continuous models produce more results than a visualisation package can process. Further, models with fine meshes and many mesh points can take a long time to visualise. This is particularly true of three-dimensional models, where it is necessary to identify which mesh points lie on the outer surface of the model before visualisation can proceed. In order to produce usable images as quickly as possible, some packages use routines to produce a close approximation to the mesh that can be visualised more quickly than the full mesh.

These routines usually form a polygonal approximation to the original geometry, preserving important features such as boundaries and holes, by not using internal mesh points to define the geometry. As an example, consider the two figures below. Figure 6(a) defines a five-point mesh of a square. However, node 1 at the centre does not define the geometry of the square, so it can be removed from the geometry definition without any problems. Figure 6(b) shows a plate with a hole in it. Here, if node 1 is removed there is a significant detrimental effect on the visualisation of the geometry due to the deformation of the visualisation of the hole.

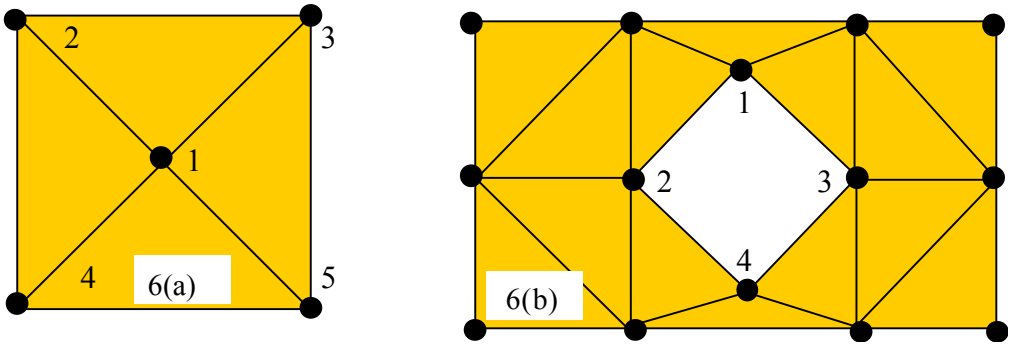

Figure 6: Examples of visualisations where removal of node 1 will (6(b)) and will not (6(a)) affect the visualisation of the overall geometry.

The algorithms used for this procedure require a tolerance in order to decide whether or not a mesh point can be ignored. If the distortion to the geometry visualisation is less than the tolerance, the node is ignored. The user can change this tolerance. It is likely that the default tolerance will be suitable for the majority of applications, but models in which the details of the geometry are crucial may require the tolerance to be altered so that only a very low level of mesh distortion is accepted.

2.3.5 Recommendations

- Read the package documentation to find out which parameters affecting the display have default values.
- Consider which values of these parameters are most suitable for the results being visualised.
- If it is not clear which value is most suitable, adjust the value and rerun the visualisation to obtain an idea of the effect of the parameter.
- If shell or beam elements with multiple through-thickness integration points have been used, check which point has been chosen for the visualisation of results.
- When visualising vector or tensor quantities in component form, make sure that the visualisation makes clear to which set of axes the components refer.
- Consider limiting the visualisation to a small part of the model if displaying the full model would produce a misleading picture of the results.
- If the model has geometric features that need to be visualised accurately, check that any algorithm tolerances are set to a sufficiently small value that the details of interest will not be neglected.

3 Visualisation of uncertainties

As mentioned in the introduction, the visualisation of uncertainties is crucial to assisting users in obtaining confidence in their visualisation of model results. It is important for users to understand the uncertainties of the results prior to making decisions based on visualisations, particularly since visualisations can encourage users to forget that the quantities they are viewing have uncertainties associated with them. Combined use of visualisations of model results and the associated uncertainties will improve the user's understanding of the overall behaviour of the model results.

An SS*f*M report by SIRA [4] on visual modelling and data visualisation outlined a six-step procedure for visualising metrology data. This procedure can equally well be applied to the visualisation of uncertainties. The six steps listed in the SIRA report [4], and their application to uncertainties, can be paraphrased as follows:

1. Assemble the measurement data (in this case, uncertainties).

The main challenge in the assembly of uncertainties for a continuous model is the calculation of the uncertainties themselves. An SS*f*M report on Uncertainty Evaluation in Continuous Modelling [1] describes a range of suitable methods for these calculations. The explanations of the techniques given in the report will not be repeated here.

2. Define the problem that the visualisation is to solve.

The definition of the visualisation problem consists of the answers to two questions. Which quantities are to have their associated uncertainties visualised? What description of the uncertainty is to be visualised? Possible answers to these questions are discussed further in section 3.1.

3. Choose the hardware and software to be used.

The choice of hardware will not be discussed in detail here, since in general it is desirable to use the same hardware for visualisation as was used for the modelling, thus avoiding the transfer of large amounts of data and results. The choice of software for the visualisation of uncertainties is discussed further in section 3.2.

4. Develop the initial solution.

The initial solution will consist of a definition of the problem, the hardware and software, the visualisation method to be used, and any input parameters, commands, or files that the chosen software requires.

5. Investigate how the visualisation can be made easier to interpret.

It is normal for the user to be able to identify improvements to the initial solution that could not be predicted at the design stage. It is useful at this stage to consult any other users of the visualisation to ensure that they interpret the visualisation in the same way as the developer of the solution, and to investigate any suggestions they may have for improvements.

6. Validate the visualisation as far as possible.

The aim of validation of visualisations is to improve confidence that the visualisation is correct. The process to use is described in section 2 of this report. The techniques mentioned there are effectively validation techniques and should be used as such.

3.1 Definition of the visualisation problem

The definition of the visualisation problem answers two questions:

- Which output quantities are to be investigated?
- How is the uncertainty to be represented?

The answers to the two questions are often inter-dependent. For example, representing the uncertainty as a distribution function may limit the range of output quantities that can be considered. Similarly, if the full results of a three-dimensional model are required, it may only be possible to visualise their mean values and the associated standard uncertainties, since typically the results of such models are given at several thousand spatial points and so visualisation of correlations or distribution functions is not feasible. Additionally, the answers to these questions will be restricted by the available hardware and software options.

3.1.1 Choice of output quantity

The most obvious choice of output quantity is the model results in their entirety. However, the majority of continuous models produce results at several thousand points, and so the retention and manipulation of distribution function information relating to all these points can sometimes be prohibitive in terms of computer memory. Additionally, some models generate multiple results at each point, not all of which are necessarily of interest.

In many cases, the user is only interested in a subset of the results. For example, measurements are often only taken at a small number of points, and the user may be particularly interested in the uncertainties associated with the results at those points. Similarly, the user may be most interested in the results on the outer boundaries of the model. If a smaller number of points is of interest, manipulating and visualising the distribution functions becomes less problematic. Reduction of the results of interest to a one- or two-dimensional set can increase the range of visualisation methods that can be used, and often makes the visualisation simpler and easier to interpret.

The user may be interested in some key feature of the results such as the location or value of the maximum and minimum of the results. There may be a single quantity that can be derived from the results that is of particular interest, such as an elastic modulus or a thermal conductivity. In such cases, the visualisation can proceed in the same way as the visualisation of the results of a discrete model, since the origin of the quantities is unimportant.

The output quantity should be chosen to provide a complete definition of the results to be used, including spatial location, time step if the results come from a transient model, and the nature of the results (scalar, vector or tensor).

3.1.2 Choice of representation of the uncertainty

The distribution function for a quantity can be represented in several ways. A plot of the distribution function or probability density function will give the most information, but often it is more convenient to summarise the quantity as a mean value, a standard deviation or standard uncertainty, and a coverage interval. However, if the quantity does not follow one of the standard distributions, a significant amount of information can be lost if a distribution function is not supplied.

The representation of uncertainties associated with two- and three-dimensional results can be problematic. If the distribution functions associated with all the results are

displayed simultaneously, the visualisation will be cluttered and confused. If the results are displayed as mean values and standard uncertainties, the visualisation will be clear but detailed information will be lost. However, the nature of some continuous models can be of assistance in displaying their results as uncertain quantities.

Many of the most common discretisation methods used for continuous models produce large linear systems of equations that link the values of the variables at the mesh points. If the model is time-dependent, the values of the variables at each new time step are usually written as a linear combination of the values at the previous time step. This strong inter-dependency between variable values means that the model results at different mesh points are typically mutually dependent. In some cases, the uncertainties associated with the results at different points will have the same shape of distribution function, albeit with different parameter values. Thus, if the distribution functions are normalised, for example by defining

$$v_i = \frac{u_i - u_{\text{mean}}}{\text{stdev}(u)}, \tag{2}$$

where u_i are the result values, and u_{mean} and stdev(u) are the mean and standard deviation of u respectively, then a single representative plot of v_i against cumulative probability can be displayed to visualise the distribution function for the entire set of model results.

As an example, consider the chemical reaction that produces hydrogen bromide,

$$\mathrm{H_2 + Br_2 \rightarrow 2HBr}.$$

It can be shown that the differential equations governing the progress of the reaction can be written as

$$\frac{d[\mathrm{H_2}]}{dt} = \frac{d[\mathrm{Br_2}]}{dt} = -\frac{1}{2}\frac{d[\mathrm{HBr}]}{dt} = \frac{k_1[\mathrm{H_2}][\mathrm{Br_2}]^{1/2}}{k_2 + [\mathrm{HBr}]/[\mathrm{Br_2}]}, \tag{3}$$

where [A] denotes the concentration of molecule A, in moles per litre, k_1 and k_2 are constants, and it is assumed that the reaction takes place at constant volume. Suppose that k_1 and k_2 are both inexact quantities, with k_1 uniformly distributed between 0.9 and 1.1, and k_2 similarly between 1.9 and 2.1.

The solution to equation (3) can be approximated numerically using a finite difference method. The results of interest are taken to be the concentration of Br_2 at the ten times $t = 0.1, 0.2, 0.3, \ldots, 1.0$, and a Monte Carlo simulation is run. The distribution function for each of the results is constructed from the MCS results. The distribution functions after normalising using equation (2) are shown in figure 7.

These functions are sufficiently similar to regard them as having the same shape, and so one possible method of displaying all of the information about the distribution functions of the results is shown in figure 8. The sub-plots are (left to right) the mean value of each result, the standard deviation of each result, and the distribution function of the results as shown in figure 7.

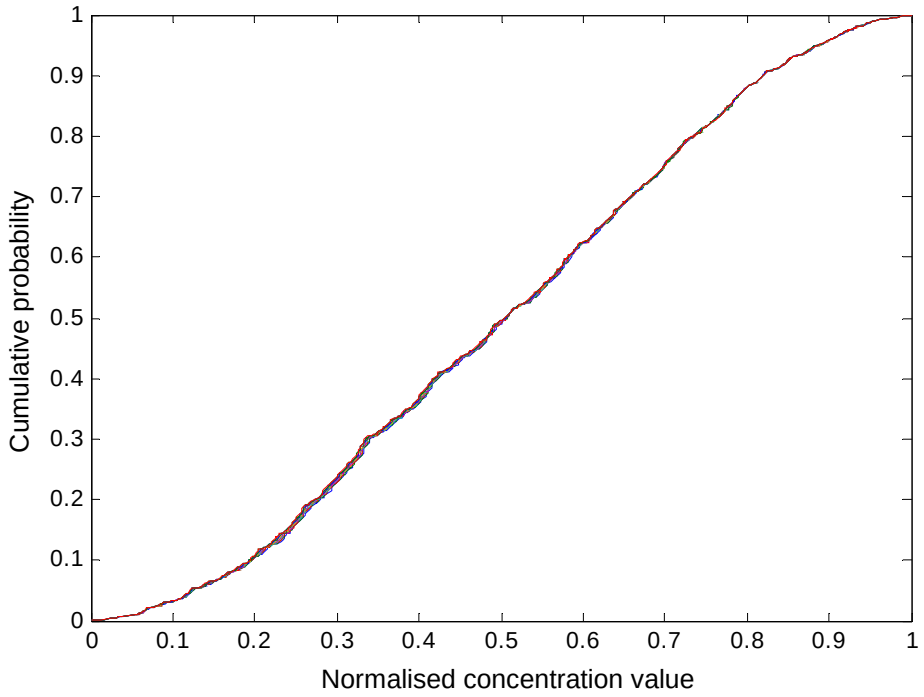

Figure 7: Distribution functions for the ten results calculated from 1 000 MCS trials.

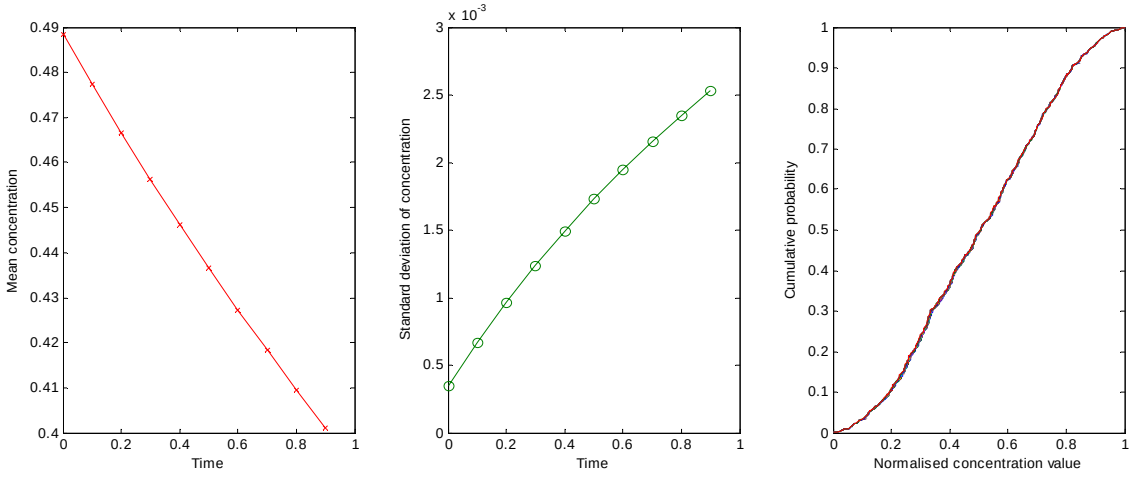

Figure 8: Simultaneous display of mean value, standard deviation, and distribution function of the results.

This feature of the results means that displaying the mean values, the standard uncertainties, and a typical distribution function should convey enough information for the user to be able to understand the behaviour of the results. However, it cannot be assumed that all continuous model results will have this feature. As a counter-example, consider one-dimensional heat transfer through a solid composed of two different materials. Suppose that the temperature distribution is continuous and that it obeys the equations

$$\frac{d}{dx}\left(\lambda_1 \frac{dT}{dx}\right) = 0, \quad 0 \le x < 0.5,$$

$$\frac{d}{dx}\left(\lambda_2 \frac{dT}{dx}\right) = 0, \quad 0.5 < x \le 1,$$

$$\lambda_1 \left.\frac{dT}{dx}\right|_{x=0} = 1, \quad T(1) = 1, \quad \lambda_1 \left.\frac{dT}{dx}\right|_{x=0.5^-} = \lambda_2 \left.\frac{dT}{dx}\right|_{x=0.5^+}.$$

The thermal conductivities of both materials are regarded as uncertain quantities, with λ_1 distributed normally with mean 1 and standard deviation 0.1, and λ_2 distributed uniformly between 1.9 and 2.1. A Monte Carlo simulation is run to simulate the problem, and distribution functions are constructed for all values of x. The results of the calculations do not have the same shape of distribution for all x. The distribution functions (normalised to lie between 0 and 1) for x = 0, x = 0.45, and x = 0.95 calculated from 1 000 MCS trials are shown in figure 9 and are clearly not identical.

An alternative display method for these results is shown in figure 10. In this figure, a surface plot is used to show how the distribution function of the normalised temperature behaves as x varies. This display method can be used to show variations along a line, and may be useful for displaying subsets of results from two- or three-dimensional problems.

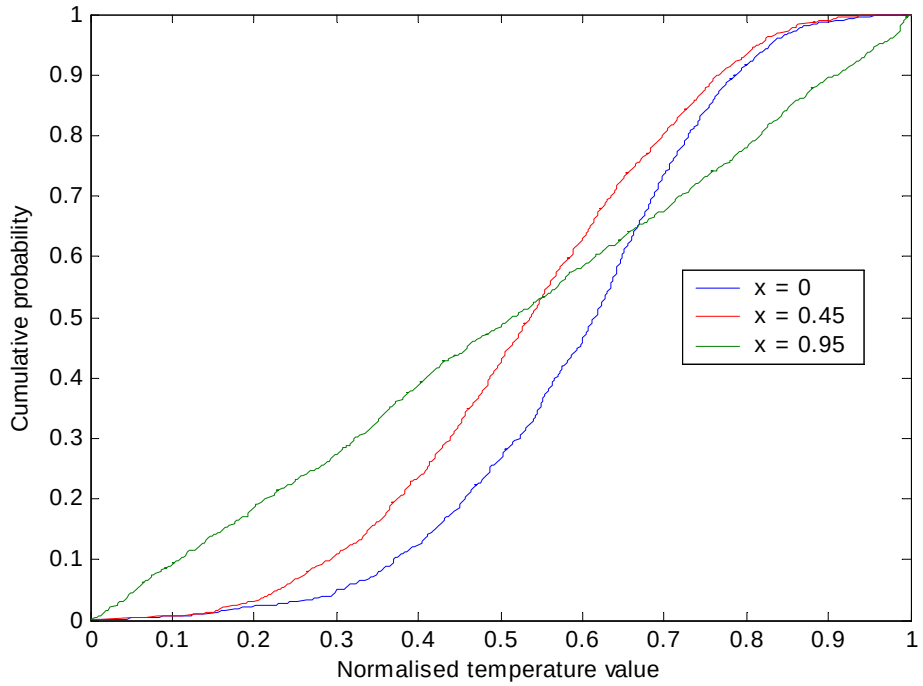

Figure 9: Probability distribution functions for the normalised value of T at x = 0, x = 0.45, and x = 0.95.

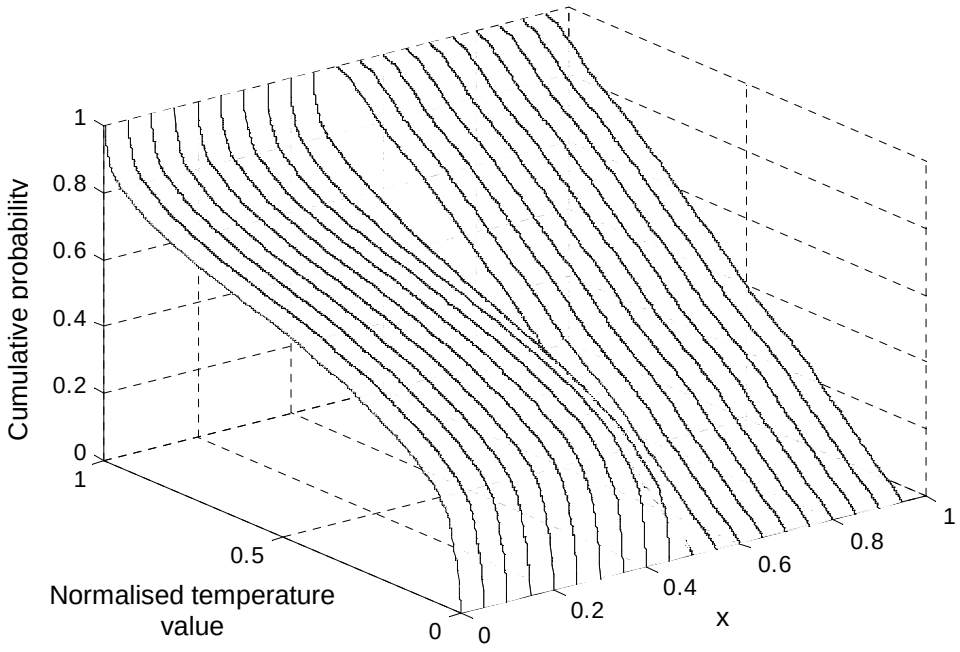

Figure 10: Variation of distribution function with x. Note the abrupt change of shape at $x = 0.5$.

Since the normalisation technique (2) fails for even this relatively simple example, it is clearly not applicable to all models. However, it may still be useful for models with a single inexact input quantity, and may be a sufficiently good approximation for models where the uncertainty associated with one input quantity dominates. In general, the technique should not be assumed to be applicable without thorough checking of the results.

Care needs to be taken when constructing distribution functions for the results of continuous models. If a single result Y is of interest, and the results of Monte Carlo simulation runs y_i, $i = 1,2, \ldots M$ are available, the procedure [9] is as follows:

1. Reorder the results so that $y_1 \leq y_2 \leq \ldots \leq y_M$.
2. Assign a cumulative probability $p_r = (r - ½)/M$ to y_r for $r = 1, 2, \ldots M$.
3. Construct the piecewise-linear function joining the points (y_r, p_r), $r = 1, \ldots, M$, so that $\hat{G}(y) = p_r + \dfrac{y - y_r}{M(y_{r+1} - y_r)}, \quad y_r \leq y \leq y_{r+1}, 1 \leq r \leq M - 1$ is the approximation to the density function $G(y)$.

This method needs careful interpretation when it is applied in this manner to results generated at many mesh points from a single model. The results at each point are likely to require a different reordering in step 1. Whilst this is not a problem in itself, it means that the user must be clear about what is meant by plots such as that in figure 10. The temperature profile obtained by finding the intersection of the surface in figure 10 with a plane of constant cumulative probability $100p$ % is not the result of a single model run: it is the simultaneous presentation of the $100p^{\text{th}}$ percentile results at each value of x.

As has been mentioned in other work on uncertainties of multiple output quantities [9], defining suitable distribution functions and coverage "intervals" for multiple output

quantities can be difficult. If there are more than three output quantities, visualisation of the coverage "interval" becomes impossible since for n output quantities it is represented as an n-dimensional ellipsoid when the distribution is (multivariate) normal, and a possibly more complicated domain in other cases. This difficulty means that the display in figure 10 may be the best visualisation solution available.

3.2 Software for visualisation of uncertainties

There are fewer packages suitable for visualisation of uncertainties associated with continuous models than there are for visualisation of continuous model results. Many continuous modelling software packages consist of a pre-processor for model development, a program for solution of models, and a post-processor for output and visualisation of results. In general, the post-processing parts of these packages will only display the results calculated by the solution program: they are not usually sufficiently flexible to display quantities, such as uncertainties, that are calculated by other means.

If the results of a three-dimensional model have been chosen to be the uncertain quantities, the choice of software packages may be quite restricted, particularly if the model geometry is complicated. Visualisation of such results requires software to be able to read in and interpret nodal positions and element connectivities, and to be able to display the uncertainties on some visualisation of the problem geometry. Some professional visualisation packages such as IRIS Explorer [6] and Amira [7] are able to read and interpret geometric output from common CAD, CFD and finite element packages such as IDEAS, Phoenics, and Hypermesh. Since these packages also accept text file input, it is possible to read a complicated geometry from the CAD or FE package output files, assign the appropriate uncertainties from a text file, and visualise the uncertainties that way.

For simpler geometries, the range of packages able to visualise results may be larger as it will be more straightforward to define the problem geometry within the visualisation package itself. The simplicity of the geometry made it possible for Matlab to be used for the visualisation of the results of the case study (see section 4).

As was noted in section 3.1, continuous model results are frequently used for calculation of a discrete quantity, and so the software used for visualisation of uncertainties in discrete models can be used for such cases. The problem illustrated in the case study in section 4 of this report includes an example of such a quantity.

3.3 Potential initial solutions

As was mentioned in section 1, the majority of techniques that are used for visualisation of continuous model results can also be used for visualisation of uncertainties. Many of these techniques will already be familiar to users and so will not be discussed in detail here. They are discussed comprehensively in the SS*f*M Good Practice Guide to Data Visualisation [8]. In brief, the techniques include:

- Contour plots
    - use colour to display values
    - useful for display of mean values, standard deviations, and standard uncertainties
    - particularly useful for two- and three-dimensional problems
    - potentially misleading for colour-blind users

- Surface plots
    - useful for displaying how the distribution function varies along a line, as in figure 10
    - can be used for displaying distribution functions for one-dimensional problems or for subsets of results of higher-dimensional problems
    - can be used for display of mean values, standard deviations, or standard uncertainties for two-dimensional problems
    - can mislead if used for plots of distribution functions: see comments in section 3.1
- Line plots
    - useful for displaying the distribution functions for a few quantities
    - simplifies the comparison of several curves if results are normalised
    - visualisation can become less clear as the number of curves increases
    - can impose assumptions regarding the interpolation of results
- Scatter plots
    - display values without making assumptions about their underlying behaviour
    - useful for results of MCS trials
    - can be hard to interpret in three dimensions

Some techniques are less easy to use for uncertain quantities. Arrow plots are commonly used for visualising velocities and other vector quantities. These plots use an arrow at each mesh point to represent the magnitude and direction of the vector. The mean and standard deviation vectors can be plotted without difficulty. The standard deviation plot may not be as straightforward to interpret as it is in the case of scalar results, because vector addition can be more difficult to carry out mentally than scalar addition. An alternative visualisation that can give a clearer picture of the vector's behaviour is to consider the coverage area or volume.

A vector in n-dimensional space can be regarded as a set of n interdependent output quantities, and so its coverage area or volume will be an n-dimensional ellipsoid when the underlying distribution is normal. Display of these ellipsoids would result in plots such as those shown in figure 11, which shows a hypothetical two-dimensional example and could represent the velocity of fluid leaving a pipe, for instance. In figure 11, the ends of the stick represent the mean velocity, and the ellipse encloses the area where the tip of the arrow that would be drawn in a normal vector plot could fall. Such "lollipop" plots could be extended to three dimensions in a straightforward way, and can be a convenient way of visualising coverage areas or volumes for vector quantities.

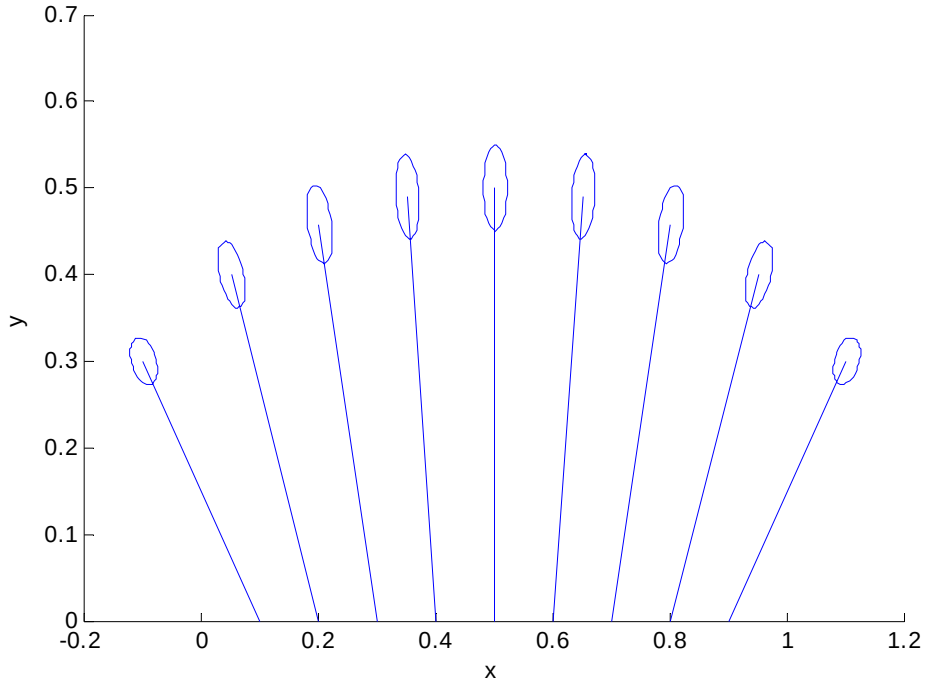

Figure 11: "Lollipop" plot of a vector quantity and its associated coverage areas.

Some quantities can benefit from having their variation animated. For instance, if the final deformed shape of some sample under load is the output quantity, the best way of comparing the shapes at different levels of cumulative probability could be in animated form. Similarly, the colour contours shown in figures 30, 31 and 34 in section 4.4.3 could be animated to show the evolution of the mean temperatures and their associated uncertainties through the thickness of the sample.

3.4 Recommendations

- Visualisation of uncertainties and visualisation of measurement data are very similar, so the methodology and techniques described in the SS*f*M Good Practice Guide on Data Visualisation [8] can be applied to visualisation of uncertainties as well.
- The definition of the visualisation problem needs to include details of the inexact output quantity and what details of its associated uncertainty are to be visualised.
- There are fewer software packages for visualisation of uncertainties associated with continuous models than there are for visualisation of "exact" continuous model results. In particular, the post-processing parts of many finite element and CFD packages are unable to visualise anything but model results.
- Many techniques for visualisation of data can be applied to uncertainties without alteration, but some techniques require adjustment for use with uncertainties.

4 Case Study: Laser Flash Thermal Diffusivity

This case study was developed from work done on a problem involving the determination of thermal diffusivity of an isotropic linear material. The work reported here describes the physical problem and resulting mathematical model, the inexact input and output quantities, and the visualisation methods that were used to investigate the output quantities and their uncertainties associated with their values.

The report does not describe the validation procedure for the model, described in a report on model validation in continuous modelling [2], nor the details of the calculation procedure for the uncertainties, described in a report on uncertainties in continuous modelling [1].

4.1 The physical problem

Thermal diffusivity, usually denoted α, is a measure of how fast the heat is transported through a material. It has units m^2s^{-1} and is defined as

$$\alpha = \lambda / (\rho c_p),$$

where λ is the thermal conductivity, c_p the specific heat capacity, and ρ the density. When the properties λ, c_p, and ρ are not known to sufficient accuracy, thermal diffusivity must be determined experimentally.

In the experiment, a cylindrical sample of material is held in a furnace in near-vacuum conditions, so no heat is lost from the sample by convection. The sample is supported by three tiny pins, so negligible heat is lost to the surroundings by conduction. One of the circular faces (the "front face") of the sample is exposed to a laser flash, and the temperature at the centre of the opposite face (the "rear face") is measured as it rises due to the laser pulse and then falls due to radiative heat loss. These measured temperature data are processed to calculate α. The experimental set-up is illustrated in figure 12.

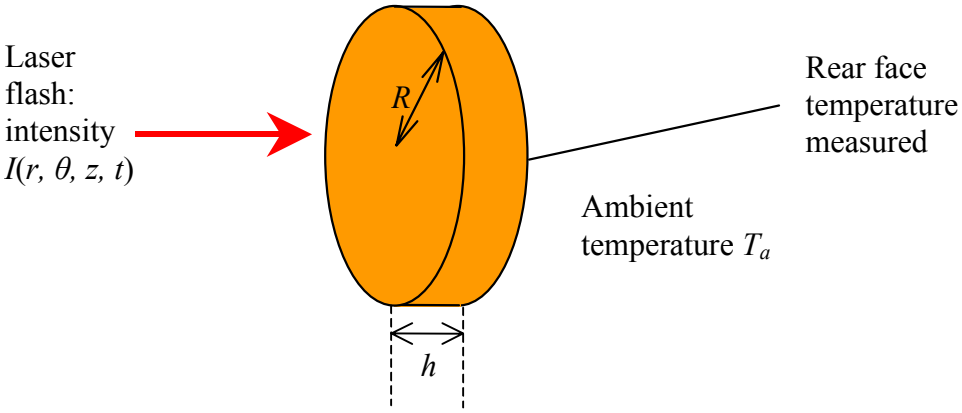

Figure 12: Illustration of the laser flash thermal diffusivity experiment.

The simplest form of the data-processing algorithm [5] used to obtain α makes various assumptions about the experimental set-up. The assumptions include: that there is no heat loss due to conduction or convection, that the material sample is isotropic and uniform, and that the sample's thermophysical properties are not affected by the small rise in temperature caused by the laser pulse. All of these assumptions are reasonable in practice. The algorithm also assumes that the laser pulse is instantaneous and uniform.

These assumptions are less likely to be valid. In practice, the laser pulse is of finite duration, and the laser profile is not uniform, which leads to uneven heating of the front face. The finite duration of the pulse is corrected for using an empirical expression in a more advanced form of the data processing algorithm [10]. Some work [11] has been done on the effects of spatially non-uniform laser profiles, but at present the algorithm for processing the experimental data does not take this non-uniformity into account.

The inexact input quantities used in this case study described the spatial variation of the laser profile. The output quantities investigated were the temperature profiles and the thermal diffusivity.

4.2 The mathematical model

In its continuous form, the model of the experiment is

$$\frac{\partial}{\partial t}\left(\rho c_p T\right) = \nabla.\left(\lambda \nabla T\right) + \delta(z) I(r, \vartheta, t), \quad 0 \le r \le R,\ 0 \le z \le h,\ 0 \le \vartheta \le 2\pi,\ 0 \le t \le t_0,$$

with boundary and initial conditions

$$T(r, \vartheta, z, 0) = T_a,$$

$$\lambda \left.\frac{\partial T}{\partial r}\right|_{r=R} = -\sigma_{\mathrm{SB}} \varepsilon_{\mathrm{hem}} \left(T^4(R, \vartheta, z, t) - T_a^4\right),$$

$$\lambda \left.\frac{\partial T}{\partial z}\right|_{z=0} = \sigma_{\mathrm{SB}} \varepsilon_{\mathrm{hem}} \left(T^4(r, \vartheta, 0, t) - T_a^4\right),$$

$$\lambda \left.\frac{\partial T}{\partial z}\right|_{z=h} = -\sigma_{\mathrm{SB}} \varepsilon_{\mathrm{hem}} \left(T^4(r, \vartheta, h, t) - T_a^4\right),$$

where (r, θ, z) are cylindrical polar coordinates, $\varepsilon_{\mathrm{hem}}$ is the hemispherical emissivity, and σ_{SB} is the Stefan-Boltzmann constant. These equations describe transient heat transfer through a uniform cylinder with radiative heat loss from all faces and a heat source on the face $z = 0$.

A discrete approximation was constructed for this problem by applying a finite difference method to the equations above, using a cylindrical polar coordinate system. The use of a polar coordinate system requires special consideration of the behaviour around $r = 0$ due to the definition of the grad and div operators in polar coordinates.

An explicit method was used to define the time integration, because the quartic boundary conditions meant that use of an implicit method would require iteration to solve the non-linear equations. Explicit methods have a maximum stable time step that can be expressed in terms of the spatial step sizes and some of the input parameters of the model. For input parameters typical of those used to describe real materials, the stable time step is sufficiently large that model solutions can be generated in a reasonable run time on a desktop PC.

The discrete model assumed that the time and spatial variation of the laser intensity $I(r, \theta, t)$ were independent, so that $I(r, \theta, t)$ can be written as $I_1(r, \theta)I_2(t)$. This meant that non-uniform spatial and temporal distributions could be taken into account. The temporal variation $I_2(t)$ was derived from measurement data of a typical laser flash, and has units Wm^{-2}. The spatial laser non-uniformity model was based on measurements of two real laser intensity profiles. The two profiles were thought to typify a “good” and “bad” laser profile in terms of uniformity. The chosen model was a two-parameter

model of the form

$$I_1(x,y)=\begin{cases} A+B\exp\left(-x^2/\sigma^2\right), & |x|<5 \text{ and } |y|<5, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

where x and y are Cartesian coordinates, B and σ are input parameters (B is dimensionless and σ is in mm), and

$$A = 0.01 - 0.1B \int_{-5}^{5} \exp\left(-x^2/\sigma^2\right) dx ,$$

so that $\int I_1(x, y)\, dxdy = 1.0$ over the area -5 mm $\le x, y \le 5$ mm.

This description of the laser non-uniformity has several features that have a predictable effect on the temperature, enabling the viewer to judge whether the visualisations "look right". Since the laser description is independent of y, and the sample geometry and boundary conditions are symmetric about $x = 0$, the temperature profiles and their associated uncertainties should all be symmetric about $x = 0$. Also, the geometry and laser have reflectional symmetry about $y = 0$, and so the results should exhibit the same symmetry.

If the laser pulse were instantaneous and perfectly uniform, the temperature at the centre of the rear face would rise to some peak value and then decay over time to a limiting value. It is expected that the peak temperature will be higher for $B > 0$ than for a uniform intensity profile, and that for a fixed value of B the peak temperature will increase as σ increases. A value of $B < 0$ will produce a lower peak temperature than in the case of the uniform profile. In general it is expected that the higher the peak temperature is, the sooner the peak temperature will occur.

4.3 Parameter values

The parameter values used for the initial investigation are shown in table 1. The thermal properties chosen were typical of those of copper at 373 K. The only parameters regarded as inexactly known were B and σ, and both were assumed to be distributed uniformly over the intervals given in the table, since no other information regarding likely distribution functions existed. Note that these parameter ranges include $B = 0$, which corresponds to a perfectly uniform intensity profile.

Parameter	Value	Parameter	Value
Thermal conductivity λ	395 $Wm^{-1}K^{-1}$	Radius R	6 mm
Specific heat capacity c_p	397 J $kg^{-1}K^{-1}$	Sample thickness h	1.568 mm
Density ρ	8.933×10^{3} kgm^{-3}	Ambient temperature T_a	373 K
Thermal diffusivity $\alpha = \lambda/\rho c_p$	1.114×10^{-4} m^2s^{-1}	Hemispherical emissivity ε_{hem}	0.1 [dimensionless]
Constant B in intensity profile	[-0.003 6, 0.01]	Constant σ in intensity profile	[1.5, 5] mm

Table 1: Parameter values used in the investigation.

The four extreme laser profiles with these values of B and σ are shown in figure 13. The curve used for $I_2(t)$ is shown in figure 14.

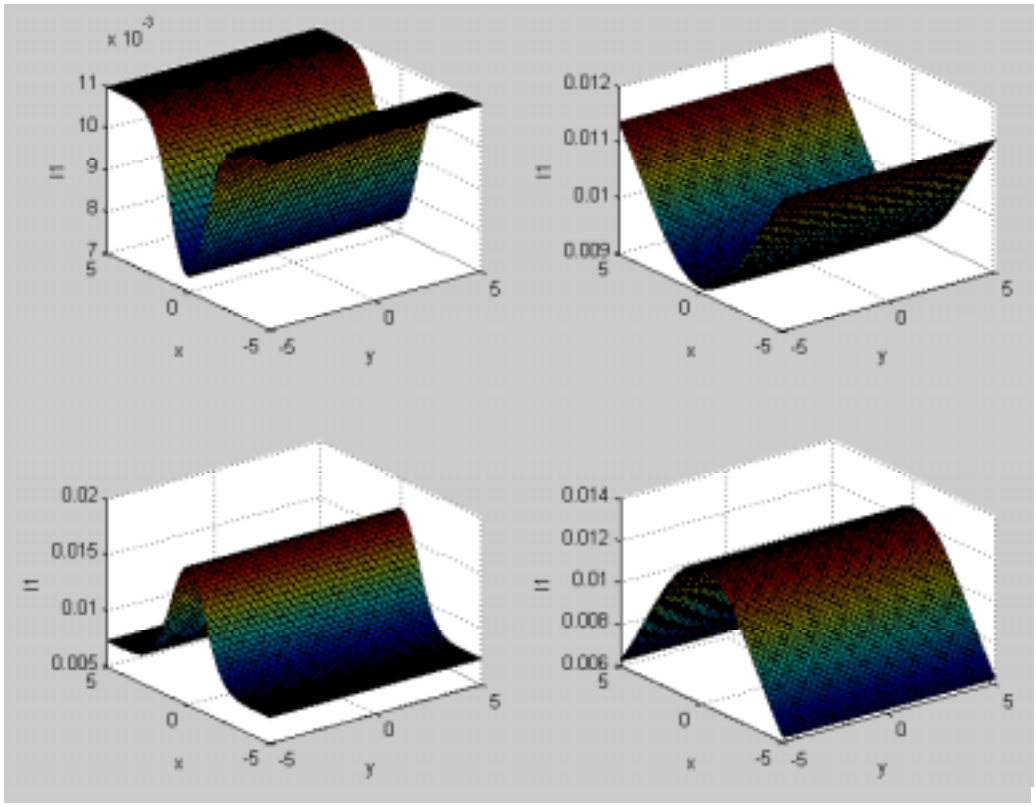

Figure 13: Laser intensity profiles generated with the limiting values of B and σ. The left-hand images correspond to σ = 1.5 and the right-hand to σ = 5. The top images relate to B = -0.0036 and the bottom to B = 0.01.

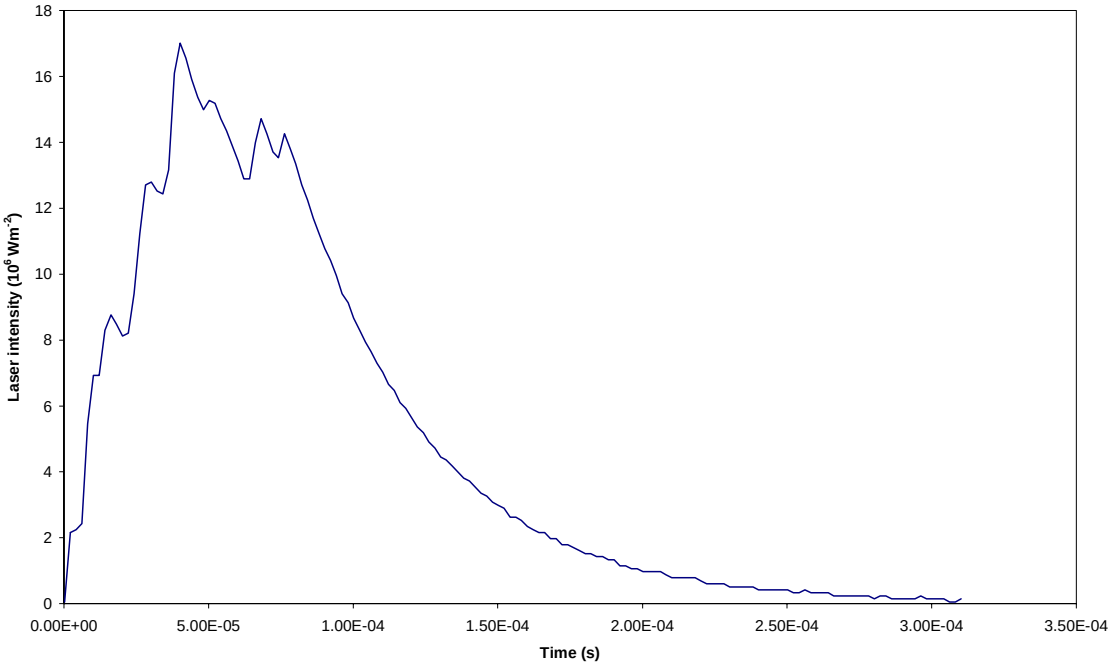

Figure 14: Laser intensity variation over time, $I_2(t)$.

The discretised model contained 11 nodes in the radial direction (one central node and ten in each radial "spoke"), 11 in the z direction, and 36 in the circumferential direction, making a total of $11 \times (1 + 10 \times 36) = 3\,961$ mesh points for which temperature results are calculated. The resulting mesh is shown in figure 15. The model was run until a time $t_0 = 30.64$ (10 000 time steps), which resulted in a typical run time on a desktop PC of about 30 s.

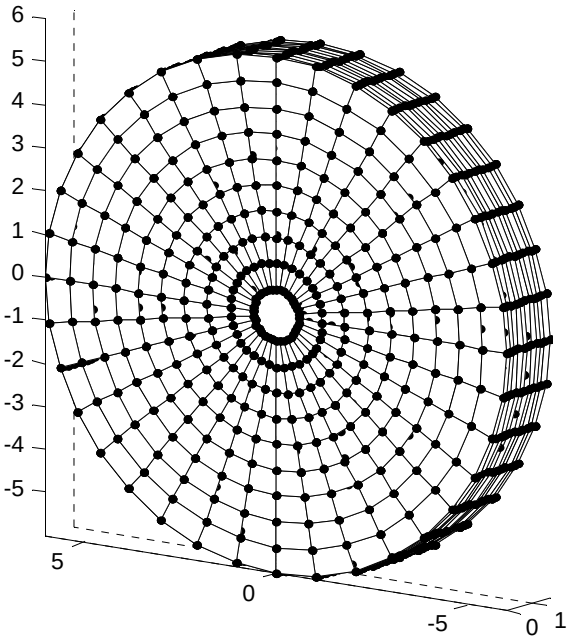

Figure 15: The mesh of the sample used for calculations. The node at the centre of the face has not been plotted.

The uncertainties were generated using the MCS method. Initially, a simulation was run using 10 000 trials, saving the value of α from each trial. This simulation aimed to produce a distribution function for α which could be used as a reference solution. Subsets of these data were used to identify a smaller number of trials that would give a sufficiently accurate approximation to the distribution function of α whilst still making the processing of the model results tractable. Following this process, it was decided that 4 000 trials would be a suitable number.

It was assumed that 4 000 trials would also approximate sufficiently accurately the distribution functions for the temperature values. This was expected to be a reasonable assumption, but convergence checks were carried out on the confidence intervals produced for the temperature results and for α, as recommended [9].

4.4 Visualisation process

The visualisation process as outlined in section 3 of this report was followed. Step 1, the assembly of the data to be visualised, has been described in section 4.3 above. The data consisted of the results of 4 000 MCS trials, and each set of trial results consisted of a single value of α, 1 000 values of the temperature at the centre of the rear face of the model, calculated every 0.364 seconds, and a calculated temperature value profile at each of the 3 971 mesh points in the three-dimensional model at a time 7.28 seconds (after 2 000 time steps).

Step 2 in the process is to define the visualisation problem. The problem that is to be solved is the visualisation of the uncertainties associated with all these values in as

detailed a manner as possible. The output quantities include three-, two- and one-dimensional data so that a wide range of visualisation solutions can be investigated.

Step 3 is to choose the hardware and software. The hardware used is a desktop PC, and two software packages have been chosen as suitable: Excel and Matlab. The choice of two packages makes it possible to validate some of the visualisations by cross-checking. Matlab was chosen because the problem geometry is simple and so it is easy to visualise in three dimensions by using "patch" and "surf" commands. Excel can be used for visualisation of one- and two-dimensional results so that some of the Matlab visualisations can be checked. The Matlab code was not formally reviewed at any stage because it was unlikely to be reused in its existing form. Instead, the code was validated as it was developed. It should be noted that the cross-checking between the two packages only validates the visualisation, rather than the entire package: there are many examples where the packages produce different visualisations of the same data.

Step 4 is development of the initial solution. The initial solution is described in three parts: the visualisation of the uncertainties associated with (a) α, (b) the time-temperature curve, and (c) the temperature distribution at a fixed time. The descriptions of these visualisations also include details of the validation carried out, which is step 5 in the visualisation process.

4.4.1 Visualising the uncertainty associated with α

As α is a single quantity, the visualisation techniques available for the associated uncertainty were the same as those that can be used for discrete model results. The work done on visualising α is included here because many continuous models are developed with the intention of deriving a single parameter from their results, and so these types of visualisation are relevant to this report despite the fact that α is not explicitly governed by a differential equation.

The first visualisation task was to obtain a plot of the distribution function for the value of α. Since this information can be plotted using Matlab or Excel, the two packages were compared for validation purposes. Figure 16 shows the Excel plot of the distribution function calculated from 10 000 MCS trials, and figure 17 the same data plotted in Matlab. The two plots look the same. This gave confidence in the visualisations of both packages.

Figure 18 shows a comparison of two approximations to the distribution function for the value of α calculated using 10 000 and 4 000 MCS trials. The two approximations looked sufficiently similar for the results of the first 4 000 trials to be used for the rest of the visualisation in this report. Additionally, probability density functions were calculated for both sets of results. These are shown in figure 19. The maximum difference between them is approximately 3% of the frequency calculated from the 10 000 trials. The convergence of the mean and standard deviation of α and the endpoints of a probabilistically symmetric 95% coverage interval for α were checked using the method outlined in [9], and each quantity had converged to the same accuracy after 4 000 trials as it had after 10 000 trials. This convergence behaviour adds confidence in the decision to use 4 000 trials throughout the remainder of the work.

No attempt was made to assign a parametrised distribution function to the results since figures 18 and 19 are not typical of any common parametric distribution. Broadly speaking, the distribution function could be approximated by the mixture [13] of three uniform distributions over disjoint intervals. It is thought that this unusually shaped function is caused by the data processing algorithm used to obtain α from the

temperature results. The algorithm contains various conditional statements that are likely to lead to a distribution function that can be considered as a mixture of functions on disjoint intervals.

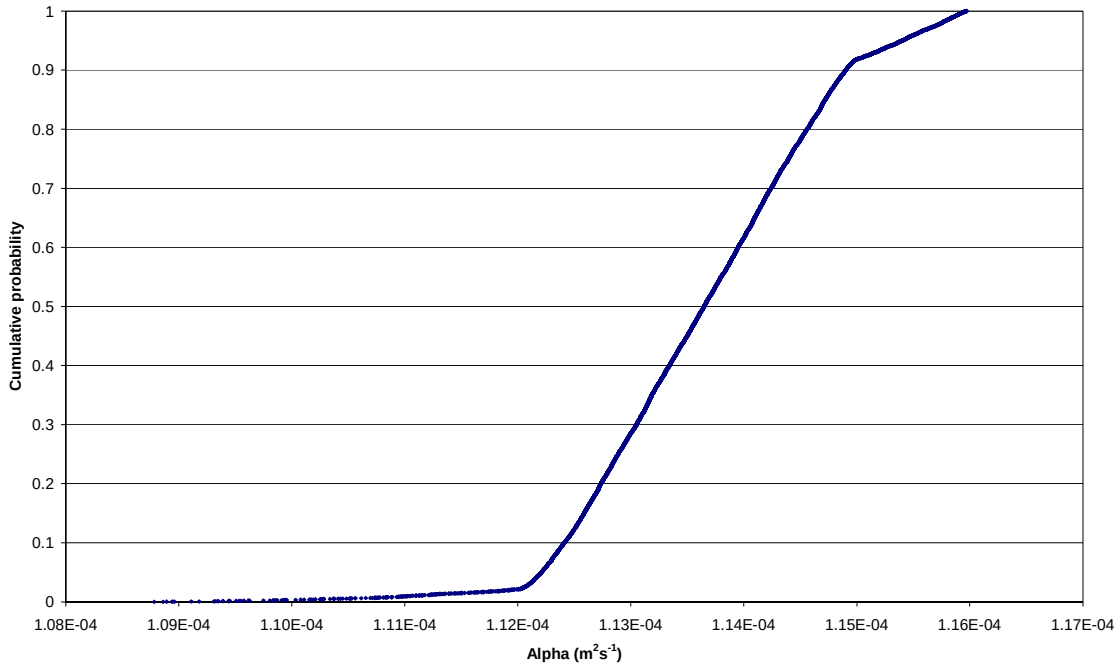

Figure 16: Distribution function for the value of α calculated from 10 000 MCS trials visualised using Excel.

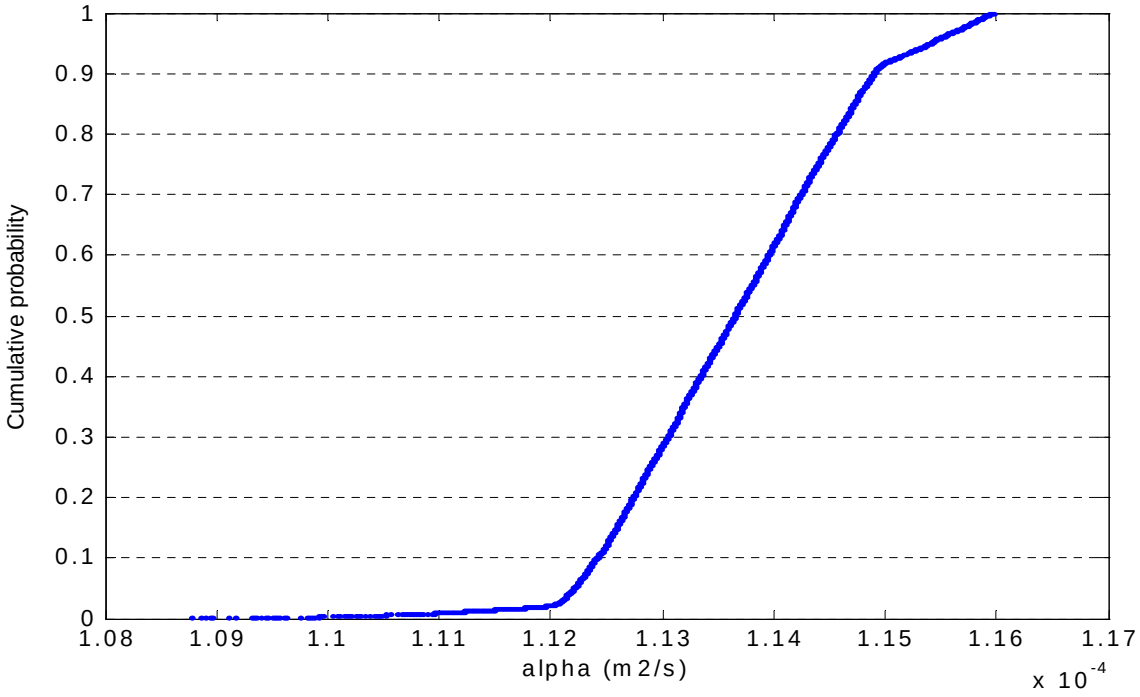

Figure 17: As Figure 16 but for Matlab.

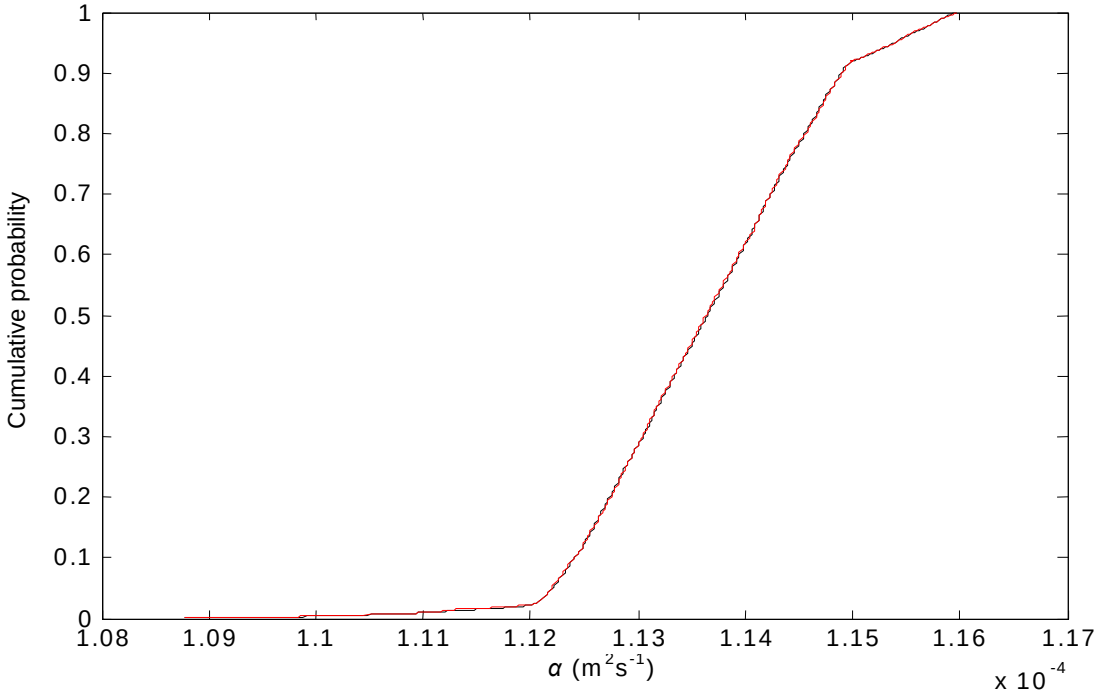

Figure 18: Distribution function for the value of α calculated from 10 000 (black) and 4 000 (red) MCS trials. The two lines are virtually indistinguishable at this magnification.

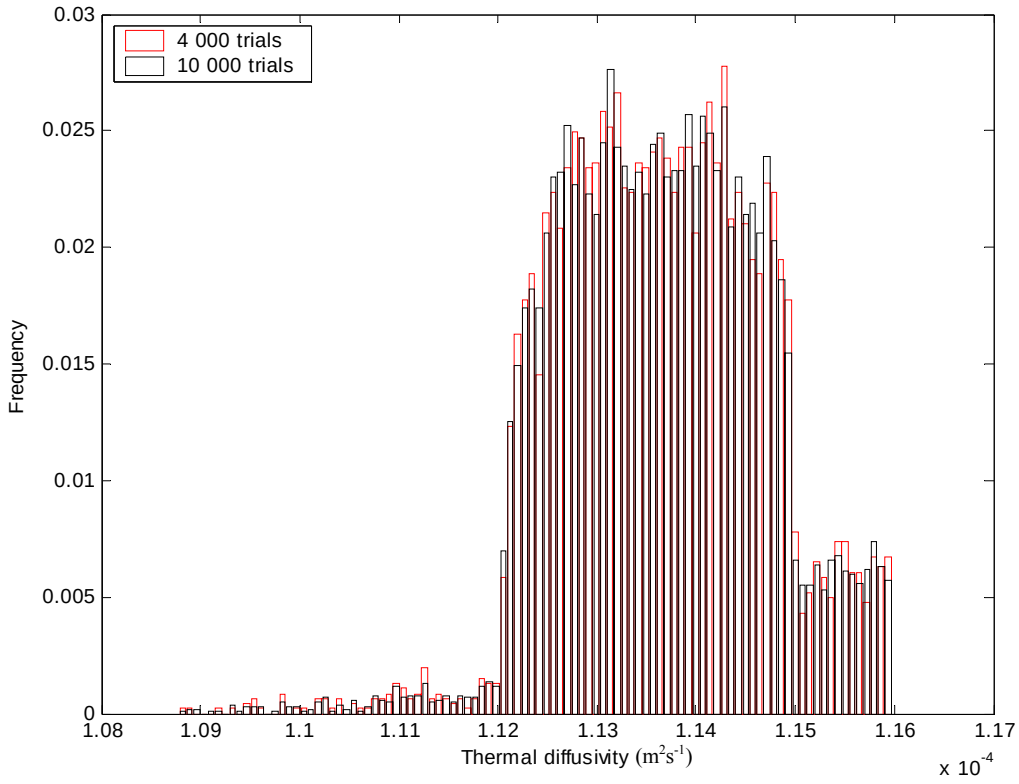

Figure 19: Probability density functions calculated from 10 000 trials (black bars) and 4 000 trials (red bars).

Next, visualisation tools are used to investigate how the inexactly known parameters B and σ affected the value of α. The first method used was to plot two-dimensional scatter

plots of α against B and against σ. These plots are shown in figure 20, but they are not easy to interpret. These data were also plotted in Excel for validation purposes . The Matlab and Excel visualisations looked the same, and so the Excel plots have not been included here.

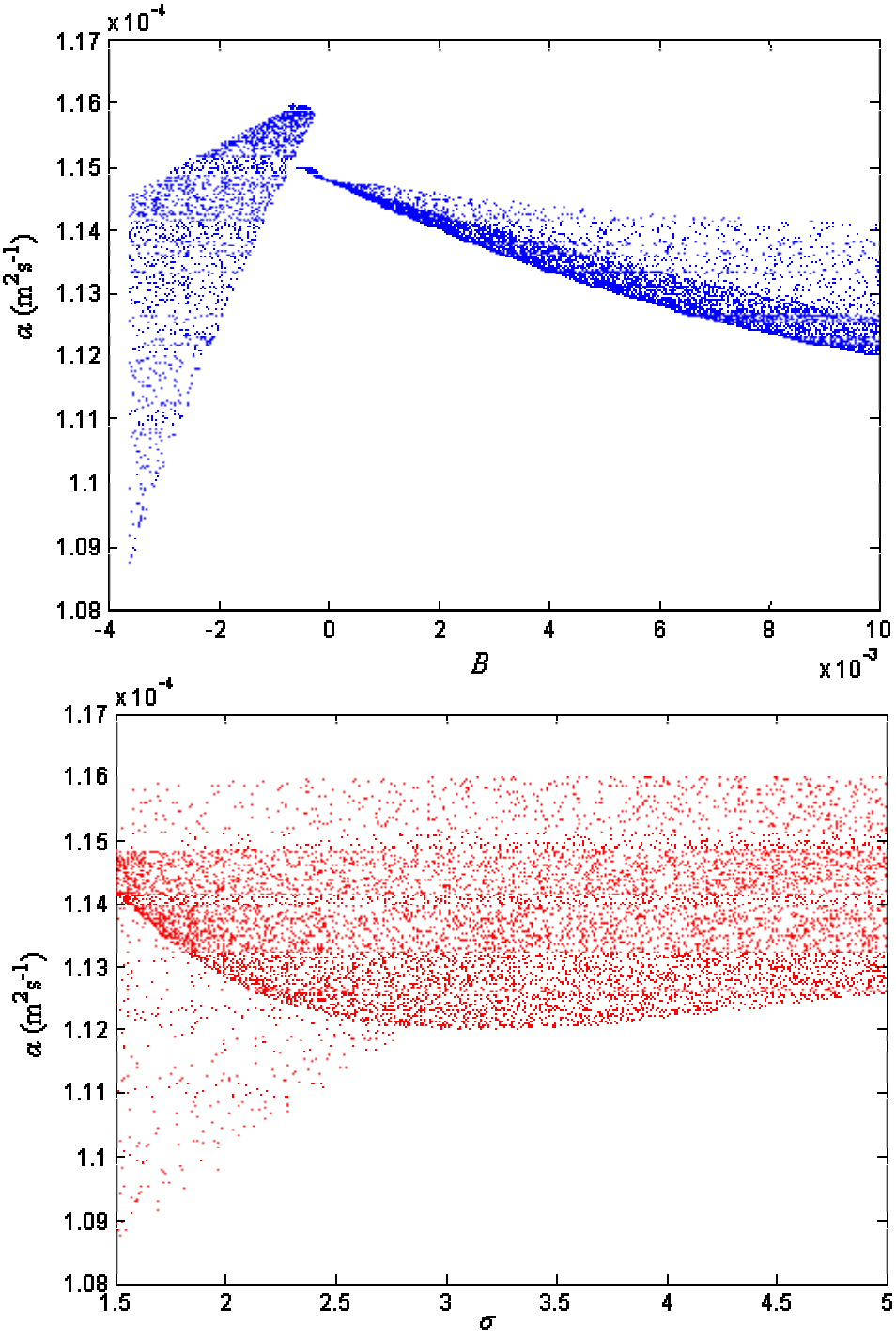

Figure 20: Two-dimensional plots of α against B (upper figure) and against σ (lower figure).

Since the results are obtained at randomly chosen values of B and σ, it is not possible readily to construct plots of α against B at a constant value of σ. Such plots would be more helpful than those shown in figure 20. Instead, a three-dimensional scatter plot has been constructed. This plot is shown from two different viewpoints in figure 21.

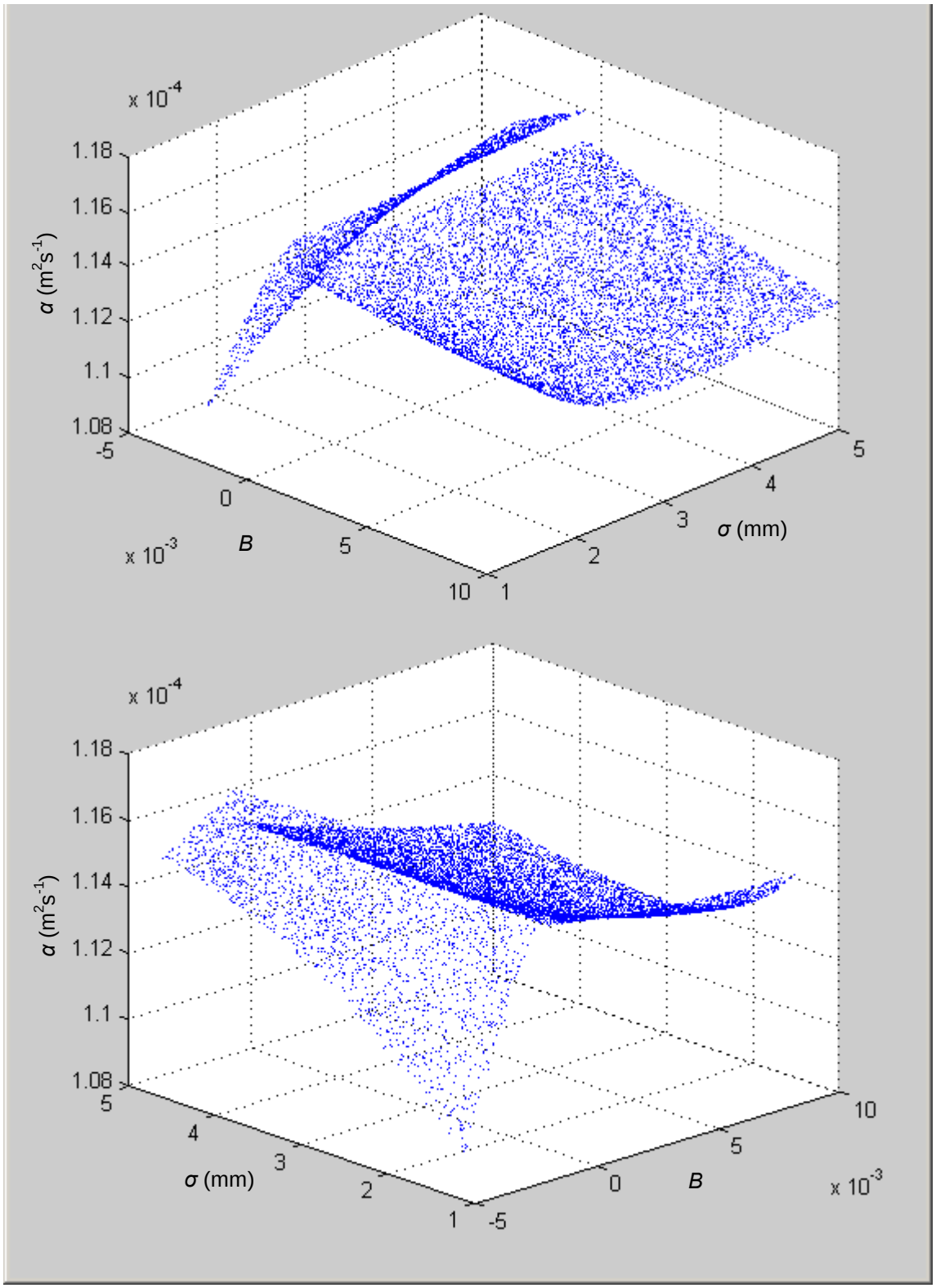

Figure 21: Three-dimensional scatter plot of α against B and σ, seen from two orthogonal viewpoints.

The three-dimensional plot gives a clearer picture of how α behaves as B and σ vary. As would be expected, α is independent of σ for $B = 0$, which increases confidence in the visualisation. Additionally, it is possible to rotate the three-dimensional view so that the plots shown in figure 20 are produced, which helps to validate the three-dimensional display.

4.4.2 Visualising the uncertainty of the time-temperature curve

The time-temperature curves each consist of 1 000 temperature values calculated by averaging the model results over the area of the measurement spot at fixed time intervals. There are 4 000 such curves. The visualisations in this section were not compared with Excel plots because the required manipulations of 4 000 000 data points proved to be problematic in Excel.

The data was processed by sorting the results at each time step independently of all other time steps. This way of processing the data was not ideal, since the nature of the model means that the results at different time steps will be interdependent. However, there was no convenient way of taking the interdependence of the 1 000 sets of data into account. In practice, the sorting orders of the different data sets were almost identical, so for instance the curve with highest temperature at one time step had the highest temperature for all time steps. Consequently, the temperature curves did not intersect very frequently and it was almost possible to treat each curve as a single entity. However, crossovers did occur and so this approach was not used.

The assumption of independence of results at different time steps means that the figures need careful interpretation, as was explained in section 3.1.2. The various curves of temperature versus time are not model outputs for a given set of input values; they constitute a line linking the temperatures associated with a given level of cumulative probability at each time step.

Figure 22 shows five curves: the maximum curve, the minimum curve, the mean curve, and the bounds of a probabilistically symmetric 95% coverage interval [9] for the temperature at each time. This figure encapsulates most of the information about how the temperature profile behaves in a form that is easy to interpret.

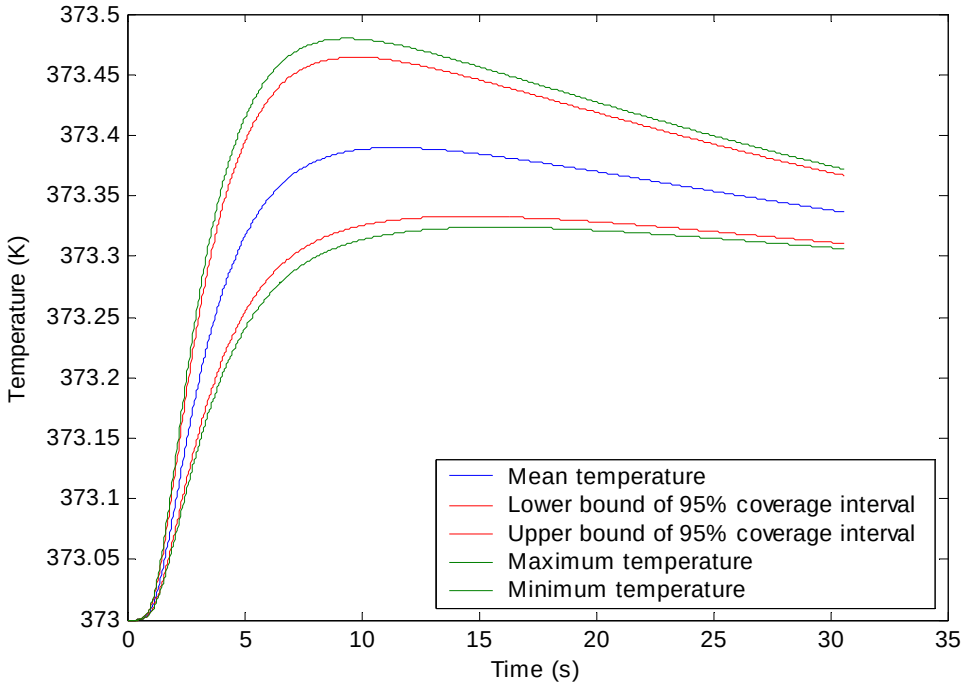

Figure 22: A summary of the information in figure 21, showing plots of the maximum and minimum temperature curves, the mean temperature curve, and a probabilistically symmetric 95% coverage interval.

Figure 22 has several of the features mentioned in section 4.2 that mean it “looks right”.

The curves rise to a peak temperature and then decrease. The peak temperatures occur at about the right time. The mean temperature is about halfway between the maximum and minimum values at most times. These features increase confidence that the visualisation is correct.

Figure 23 shows the standard deviation associated with the value of the temperature at the different time steps, calculated from the MCS results. The conduction within the sample means that the initial distribution of the laser intensity profile on the front face of the sample becomes less significant as time increases, which is why the standard deviation decreases more rapidly than the mean temperature does as time increases.

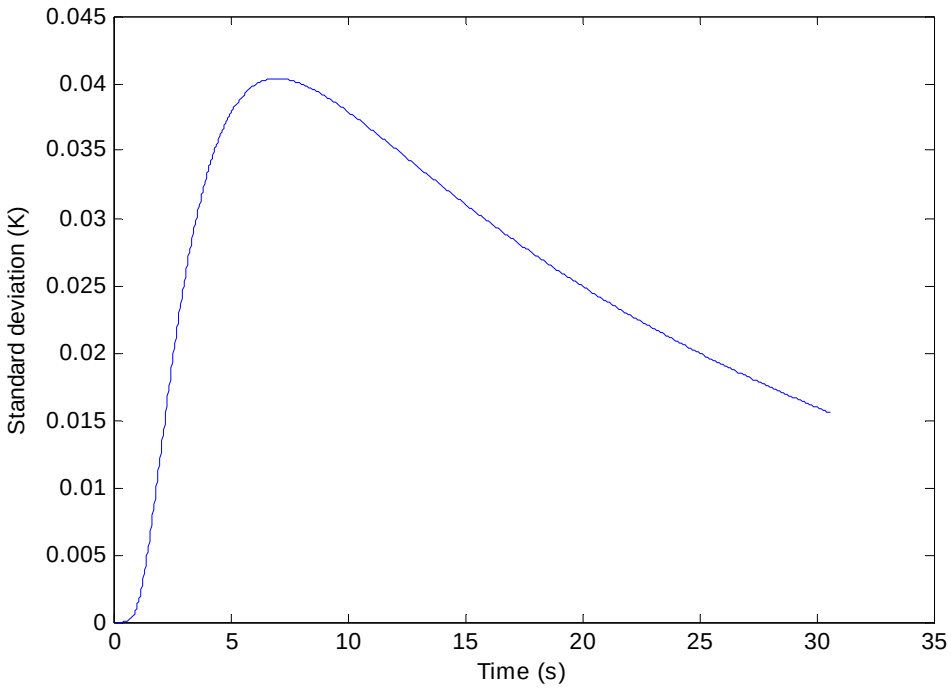

Figure 23: Standard deviation of the temperature at the centre of the rear face at each time step, calculated from the MCS results.

It can be useful for the metrologist to know the probability that the temperature is less than some prescribed value T^*. A plot of how this probability changes over time can be constructed by connecting the probability levels corresponding to the temperature T^* at different times. An example is shown in figure 24. The plot shows four lines constructed using this method. From this plot, the user can tell, for example, that the temperature never falls below 373.3 K after approximately 8s, and that it never rises above 373.45 K after approximately 16s.

The integral of these curves with respect to time divided by the duration t_0 of the experiment gives the total probability that the temperature at the centre of the rear face remains below the temperature T^*. This type of information may be useful if there is some temperature limit that must not be exceeded by the sample, although it must be noted that the temperature shown is that at the centre of the rear face rather than the maximum temperature of the sample. It may be more appropriate to use the temperature at the centre of the front face as an output quantity under such circumstances, since this is most likely to be the maximum temperature.

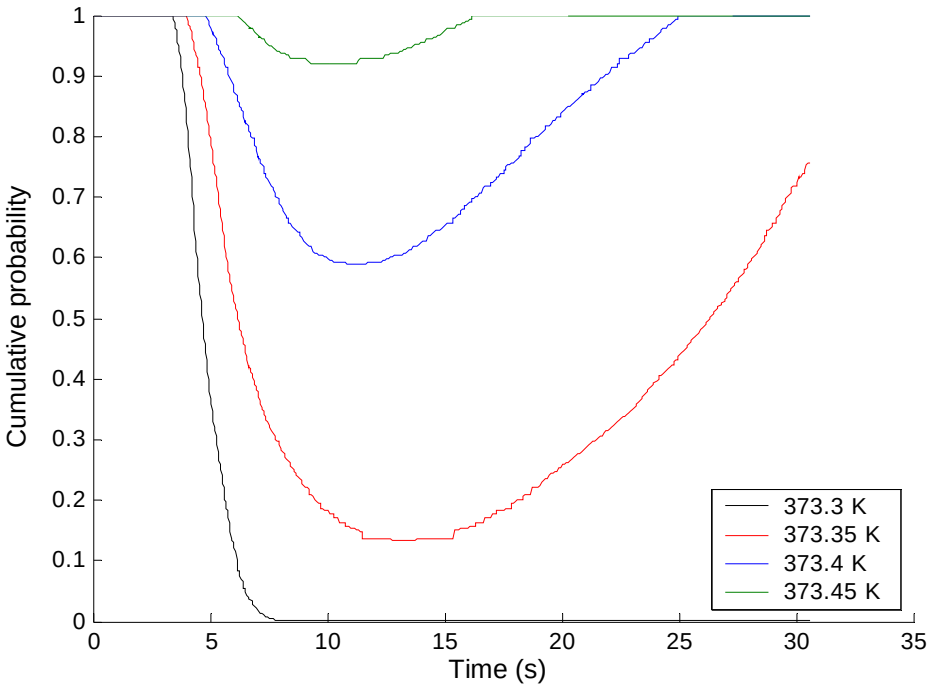

Figure 24: Lines showing how the probability that the temperature of the rear face of the sample is lower than some temperature T^* changes as time increases. Each curve shows a different value of T^*.

The maximum temperature achieved and the time at which it occurs are both of interest during the calculation of α. The distribution functions for these two quantities are shown in figure 25. Displaying these plots together may be slightly misleading as doing so could be taken to imply that results at the same level of cumulative probability were generated by the same input parameters. This is not the case; in fact in general the higher the maximum temperature is, the earlier it occurs. Figure 26 shows the relationship between maximum temperature and time of its occurrence and it is clear that the two quantities are not directly proportional to one another. The nonlinear relationship shown in figure 26 means that figure 25 must be interpreted with care.

It is also of value to consider the effects of the input quantities B and σ on the maximum temperature. The temperature curves are plots of the temperature at the centre of the rear face of the sample against time. The highest maximum temperature will occur when the spatial laser intensity has a large peak at $r = 0$. From the form given in (4), this should occur when σ is large and B is large and positive.

Figure 27 shows maximum temperature plotted against B and σ. The plot shows the expected behaviour, which provides a further check of the visualisation software. The dependence of peak temperature on B and σ is considerably simpler than the dependence of α on B and σ as shown in figure 21. This illustrates the point made in section 4.4.1 about the effects of the data-processing algorithm on the distribution function for the value of α.

Attempts were made to normalise the individual temperature curves as described in section 3.1.2, but the process did not result in a single representative curve. Similarly, the distribution functions of temperature at fixed times were compared, but they also differed too much to normalise and obtain a representative curve.

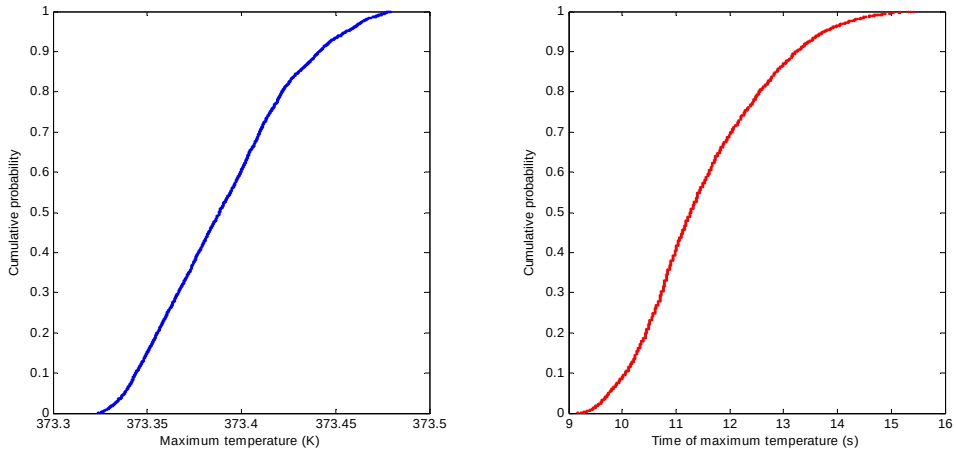

Figure 25: Distribution functions for the maximum temperature (left-hand plot) and the time at which it occurs (right-hand plot).

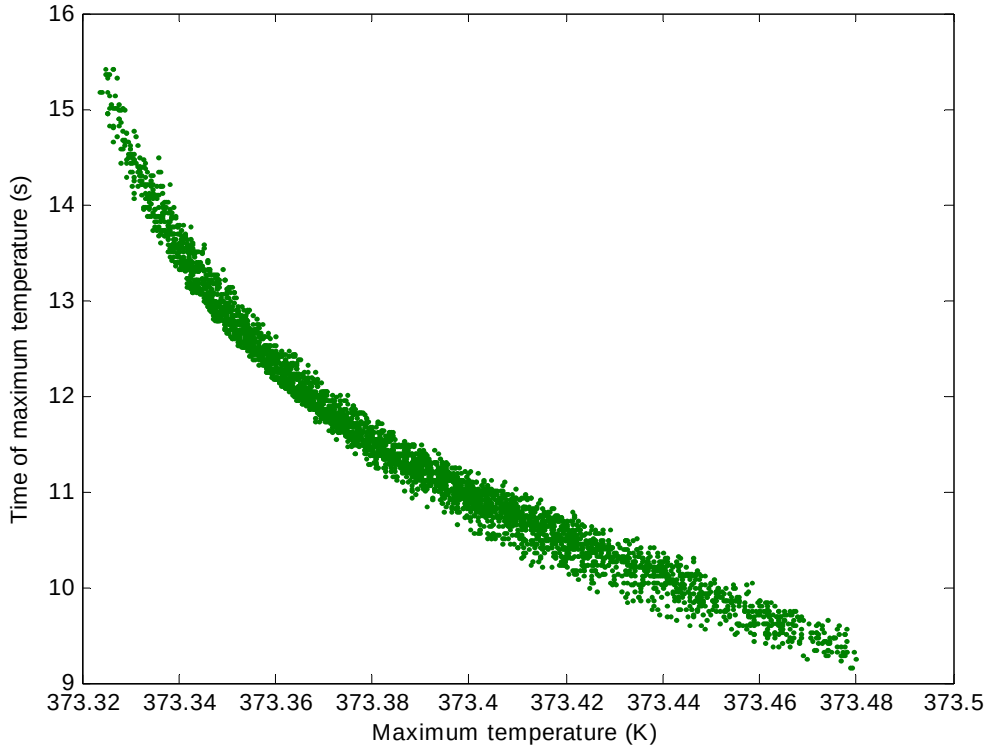

Figure 26: Plot of maximum temperature versus time of its occurrence.

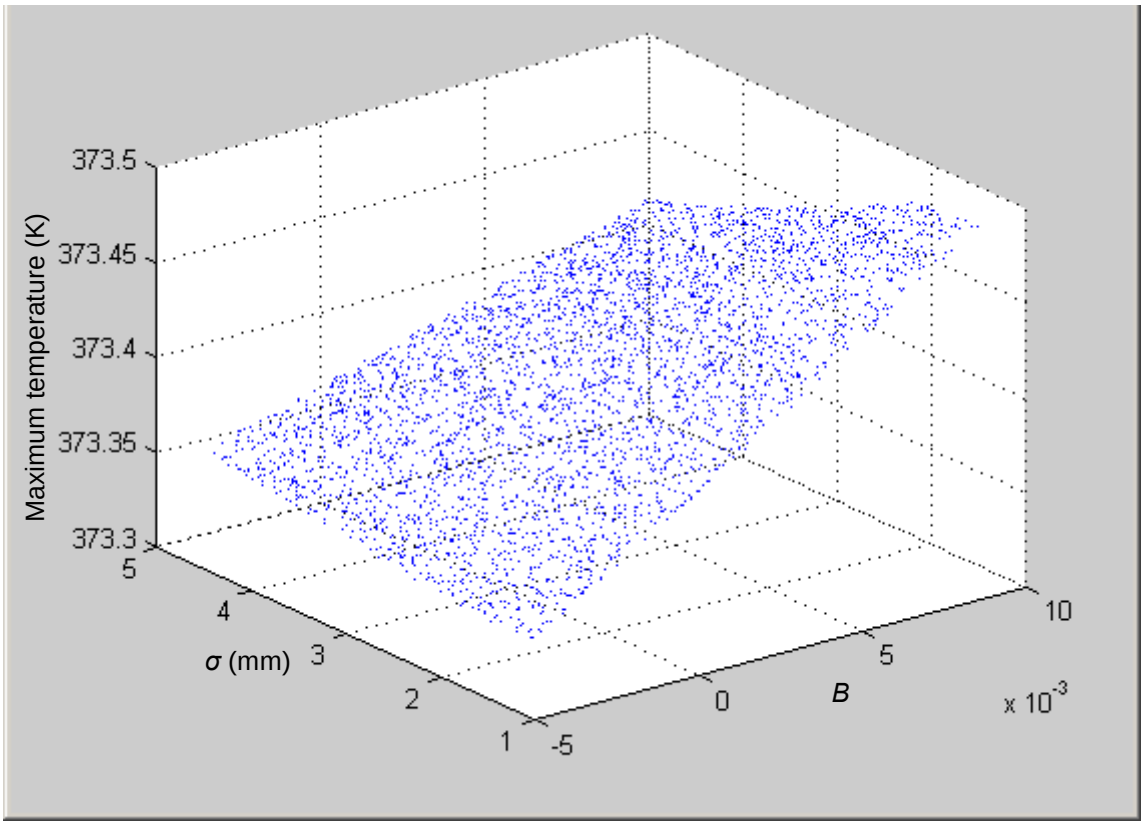

Figure 27: A 3-d scatter plot of maximum temperature against the input parameters B and σ.

4.4.3 Visualising the uncertainty associated with the temperature profile at a fixed time

The temperature profile results for a fixed time were assembled from 4 000 MCS runs, with the temperature being recorded at each of the 3 971 mesh points.

In order to investigate how the use of 4 000 trials affected the convergence of the statistical qualities of the results, the temperature results with the largest and smallest standard deviations were identified. The convergence of the means, standard deviations, and the endpoints of two probabilistically symmetric 95% coverage intervals for these temperature results was investigated using the procedure outlined in [9]. The investigation found that the mean, standard deviation, and the endpoints of the coverage interval had all converged to within 0.01 K after 4 000 trials for the result with the largest standard deviation and the quantities had converged to within 0.001 K for the result with the smallest standard deviation. It may be of interest to plot the convergence of these quantities using a contour plot, but this has not been attempted during this work.

As these sets of results are generated on a three-dimensional grid representing a fairly simple geometry, it was straightforward to plot the means and standard deviations of the results at each point as "cell" plots. These cell plots were generated from the nodal results by defining a suitable area element around each node and colouring the whole of that element with the colour corresponding to the nodal result. The plots are shown in figure 28. Both colour scales are temperatures in K.

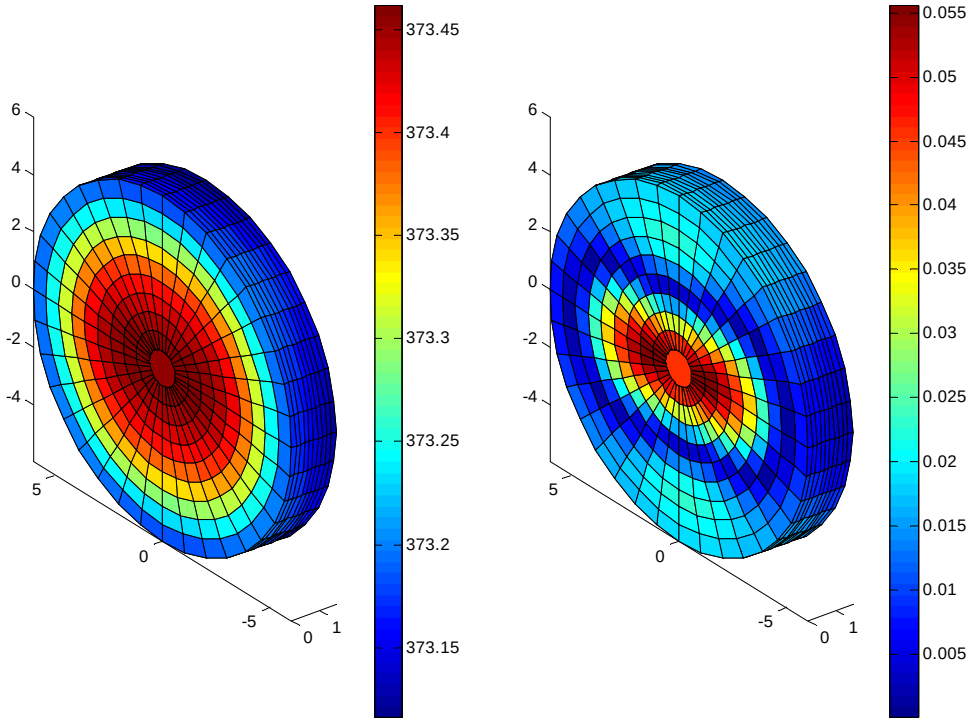

Figure 28: Three-dimensional colour cell plots showing the mean (left-hand figure) and standard deviation (right-hand figure) of the results at each point. The front face, which is directly affected by the laser flash, is facing the viewer.

These visualisations have a number of features that make them "look right". These features include:

- the maximum mean temperature is at the centre of the front face,
- the mean temperature decreases radially,
- the range of mean temperatures (about 0.35 K) seems reasonable,
- the means and standard deviations have reflectional symmetry in two directions, and
- the unsmoothed contour displays do not show any sharp changes of value.

Distribution functions were calculated for the temperatures by sorting the results independently at each point. Figure 29 below shows a probabilistically symmetric 95% coverage interval taken from these distribution functions. The left-hand plot shows the lower end of the coverage interval, and the right-hand plot the upper end. The same information is shown in figure 30, where the left-hand plot is the lower bound of the coverage interval and the right-hand plot is the width of the coverage interval.

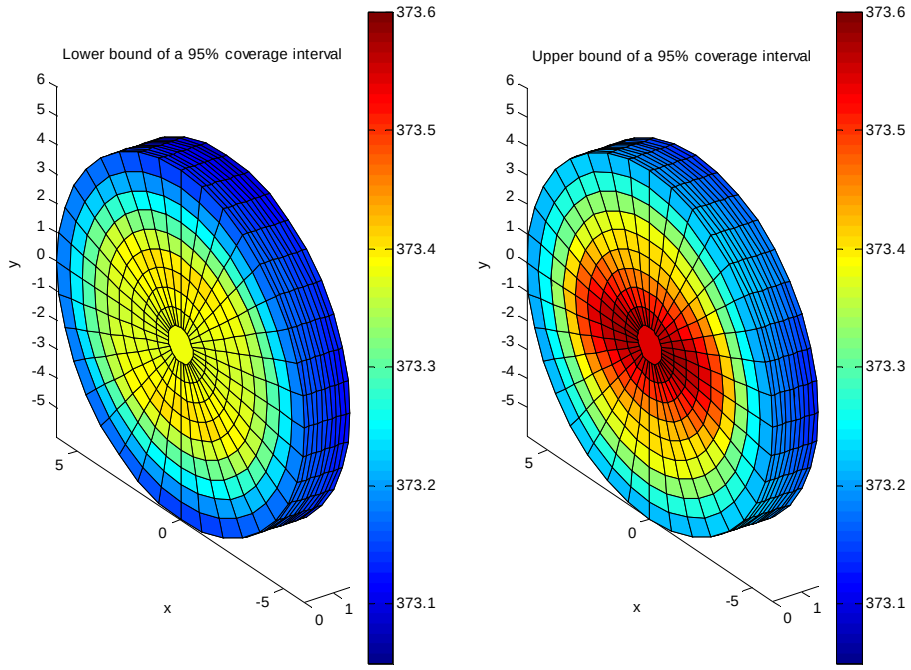

Figure 29: Colour cell plot of the lower (left-hand plot) and upper (right-hand plot) bounds of a 95% coverage interval for the value of temperature.

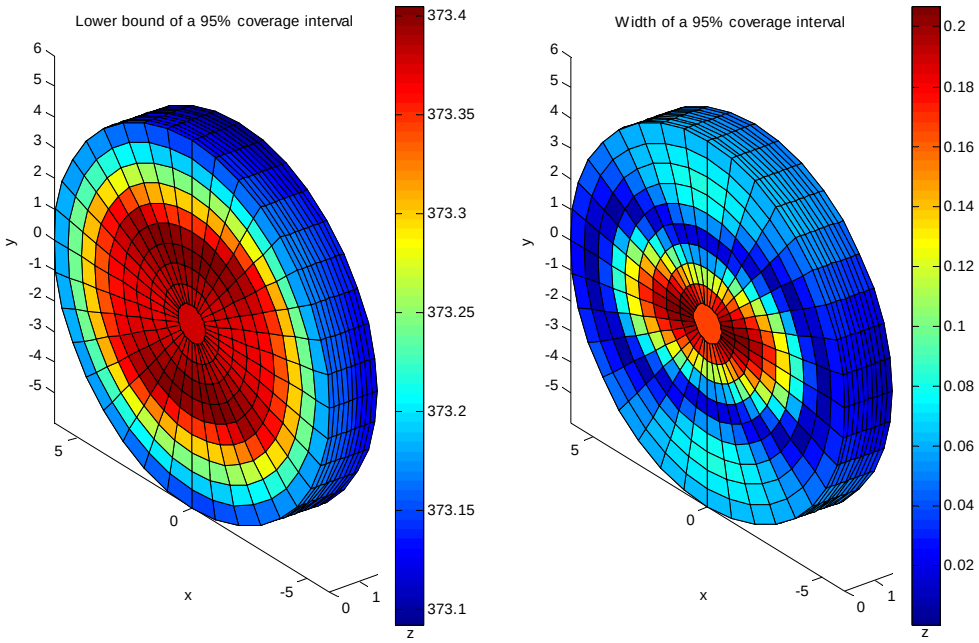

Figure 30: Colour cell plot of the lower bound (left-hand plot) and width (right-hand plot) of a 95% coverage interval for the value of temperature.

It is interesting to note the patches of light blue and green at $x \approx 0$, $y \approx \pm 5$ on the coverage interval width plot. These areas are the regions of greatest discontinuity of the expression (4) for laser intensity, since I_1 has its peak (for $B > 0$) at $x = 0$, but $I_1 = 0$ for

$|y| > 5$. This discontinuity seems to increase the standard deviation of the temperature results in that area.

These three-dimensional plots give a good insight into how the temperatures are varying on the outer surface of the sample. It is also interesting to investigate how the temperature profile varies through the thickness of the sample. The best way of looking at this variation is by plotting two-dimensional cross-sections of the sample. Figures 31 and 32 show examples of such cross-sections. Figure 31 shows the mean temperature at various values of z, and figure 32 shows the (r, z) cross-section at every 10° of rotation. Each set of cross-sections is plotted to the same colour scale.

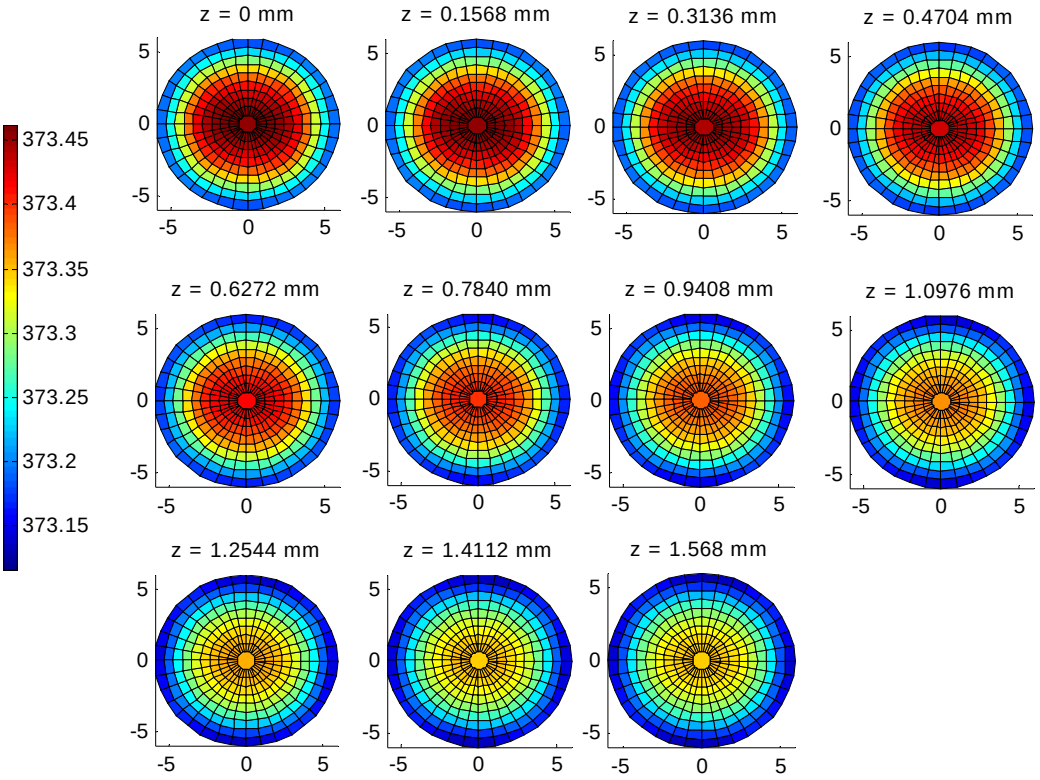

Figure 31: Cross-sections of the mean temperature at different values of z.

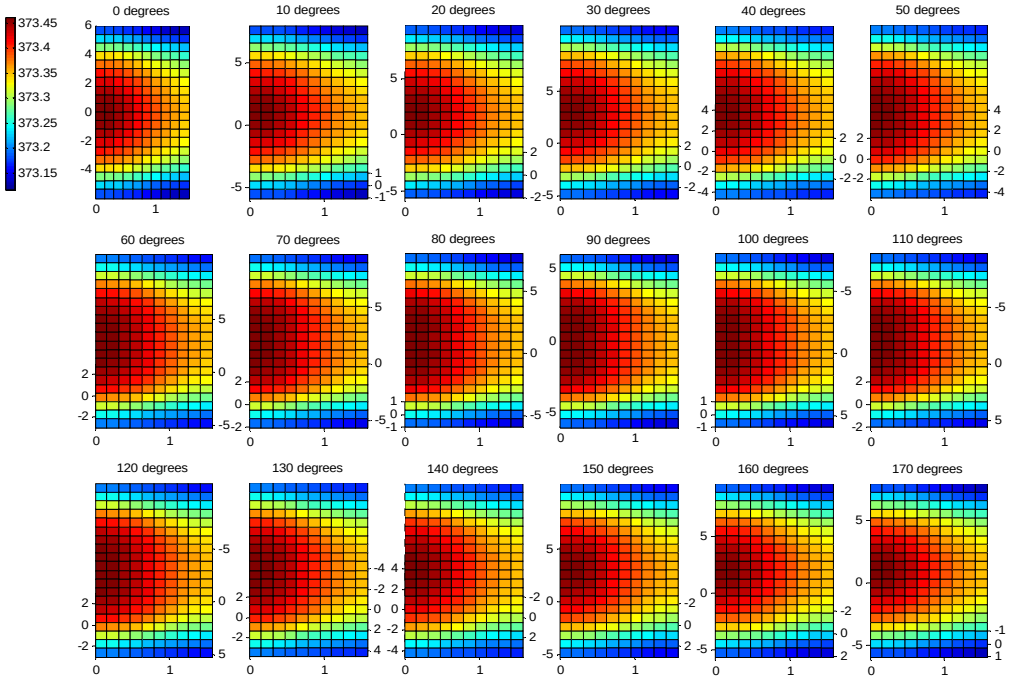

Figure 32: Cross-sections of the mean temperature at different values of θ. The x and y scales should be ignored as they are misleading due to rotation of the axes. The first plot is thinner due to problems rescaling that plot after addition of the legend.

These figures have several features that validate the visualisations. The cross-sections at different values of z show that the temperature decreases and becomes more uniform as z increases. The cross-sections at different values of θ show that the mean temperature is symmetric about $\theta = 0$. The details are not easy to pick out due to the size of the plots, but some variation in the results with θ can be picked out, in particular the size and shape of the central dark red block varies.

For further validation, the first and tenth cross-sections for different values of θ were plotted in Excel. The Excel and Matlab plots are shown in figures 33 and 34. The circular cross-sections could not be plotted conveniently in Excel and so were not used for validation. Care must be taken when comparing the contour and cell plots. The Excel contour plot interpolates smoothly between the nodal values, and so the visualisation is sensitive to the choice of interpolation algorithm, as was mentioned in section 2.1.1.

The plots look broadly the same for the two angles, increasing confidence in the Matlab visualisation. In particular, the slightly bumpy shape of the contours in figure 34 appears in both visualisations. These bumpy contours are caused by the nature of the discrete approximation at $r = 0$. As was mentioned in section 4.2, the element around $r = 0$ requires special treatment. The technique that has been used involves using the average of all of the values $\{T(\Delta r, \theta, z), 0 \leq \theta \leq 2\pi\}$ to calculate $T(0, 0, z)$. This technique will mean that $T(0, 0, z)$ will lie between the maximum and minimum of $\{T(\Delta r, \theta, z), 0 \leq \theta \leq 2\pi\}$. It is expected that the maximum of $\{T(\Delta r, \theta, z), 0 \leq \theta \leq 2\pi\}$ will occur at $\theta = \pi/2$ and the minimum at $\theta = 0$. Hence the tenth cross-section shows that the central temperature is lower than the temperature at $r = \Delta r$.

Additional cross-sectional plots were made of the standard uncertainties associated with

the results at each point. The circular cross-sections showed smooth transitions between the plots on the two outer faces in a similar manner to the plots of mean temperature, and were not of interest. The angular cross-sections are shown in figure 35 below as they are of more interest.

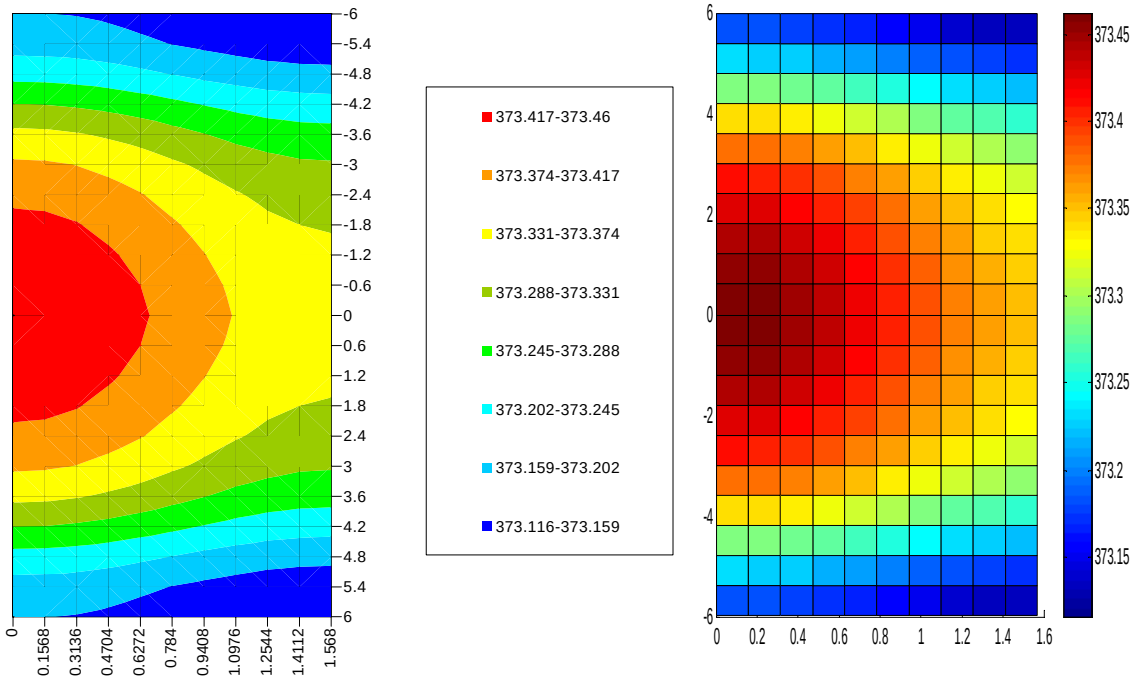

Figure 33: Cross-sectional slice at $\theta = 0^{\circ}$ plotted in Excel (left-hand plot) and Matlab (right-hand plot).

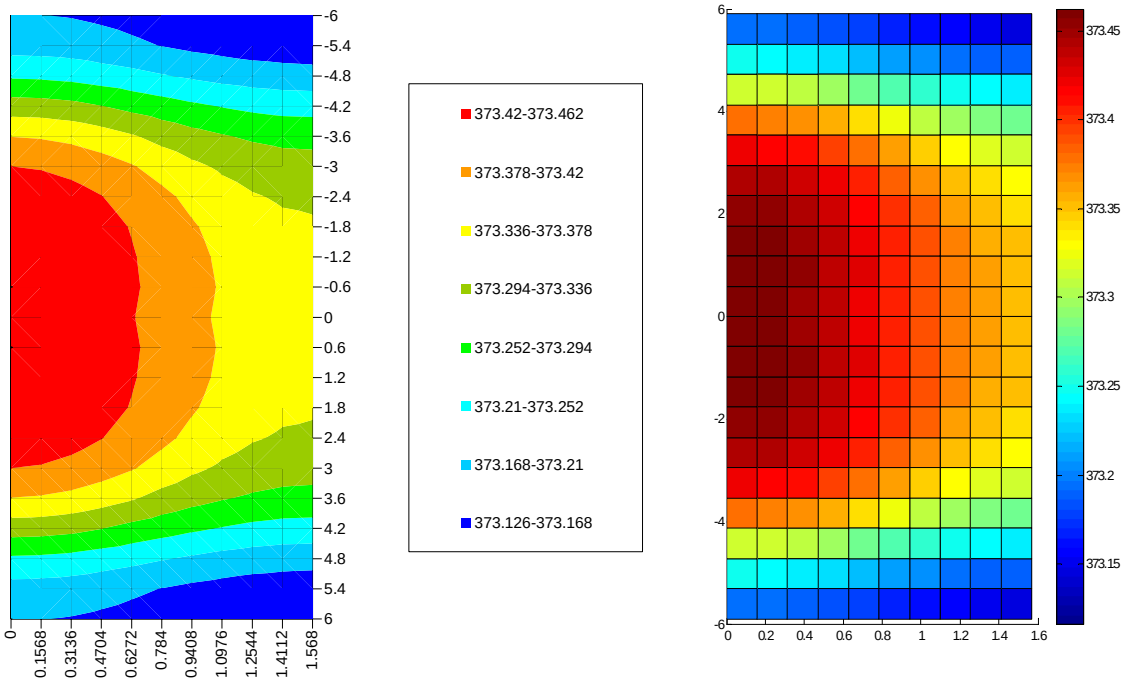

Figure 34: As figure 33, but for $\theta = 90^{\circ}$.

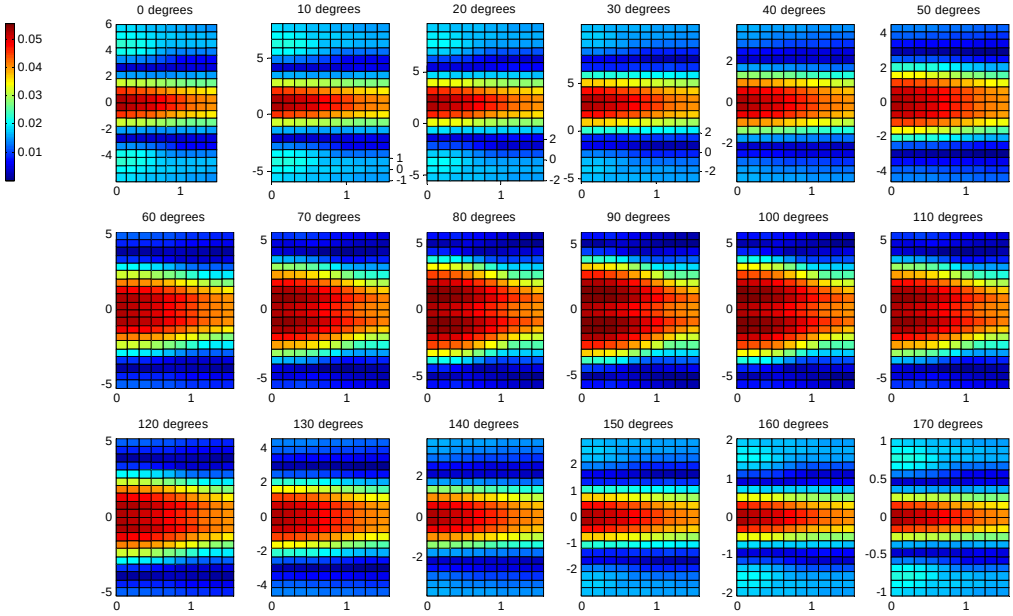

Figure 35: Cross-sections of the standard uncertainty contours at different values of θ.

One interesting feature of these plots is the behaviour of the slices around $\theta = 90^{\circ}$. The largest standard uncertainty does not occur at the centre of the plot, where the laser intensity varies over the widest range of values. It occurs in areas around $x = \pm\Delta r$. This property is true for several z slices. It is likely to be caused by the approximation used at $r = 0$, as was explained above in the description of the bumpy contours in figure 34. It is also interesting to note that the minimum standard uncertainty does not occur on the outer "rim" edge of the samples. This is probably caused by the boundary conditions of radiative heat loss.

The distribution functions for the values of temperature along $r = 0$, $0 \leq z \leq 1.568$ mm were calculated. The values along this line were chosen because $r = 0$ on the front face has the peak value of laser intensity and $r = 0$ on the rear face is the measurement spot for the temperature vs. time curves, which means that the results along $r = 0$ are important for the determination of α.

The distribution functions are shown in figure 36, and are clearly similar in shape. The similarities make the normalisation procedure described in section 3.1.2 a potentially useful tool. The normalised versions of the curves are shown in figure 37, and figure 38 shows a triple plot of the means, standard deviations, and distribution function for the values of temperature along $r = 0$. Attempts were made to fit a parametric distribution to the normalised data, but no adequate distribution was found. Figure 39 shows the pdf for the amalgamation of the values of all the normalised temperatures, and does not resemble any commonly-used distribution. It is not clear why the pdf shown in figure 39 is so jagged. It is possible that more MCS runs would be required to produce a smoother shape, but this has not been investigated further.

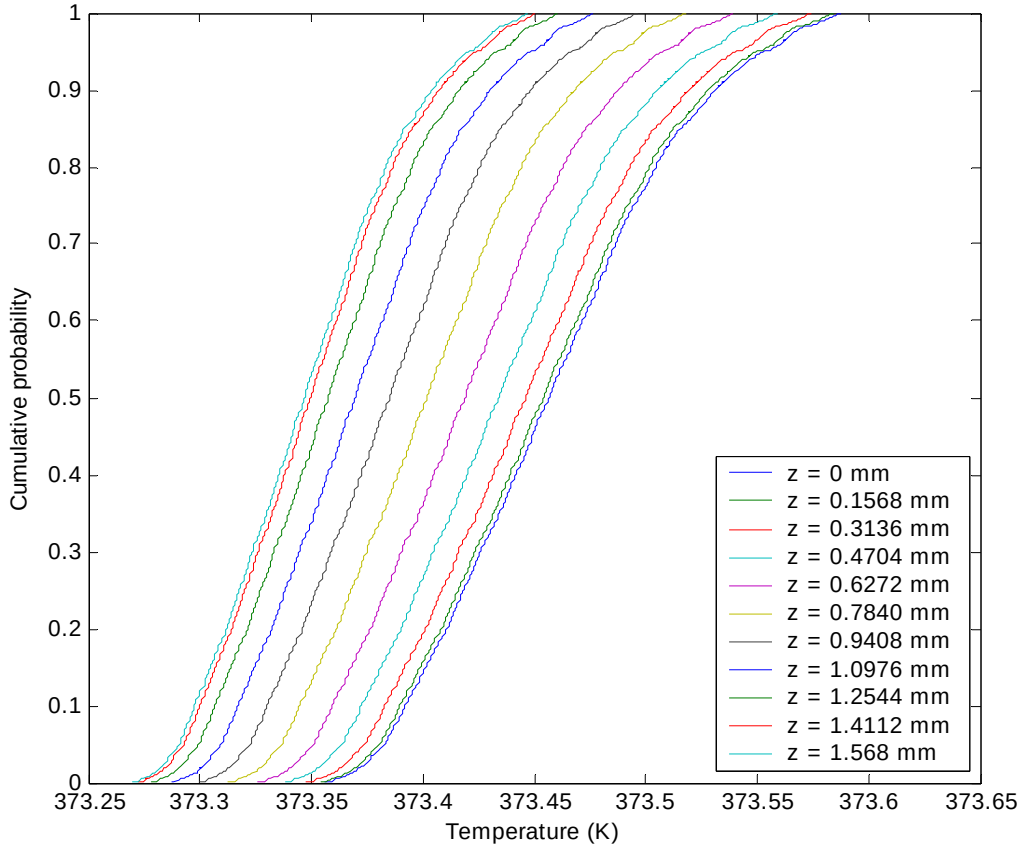

Figure 36: The distribution functions for the values of temperature along $r = 0$.

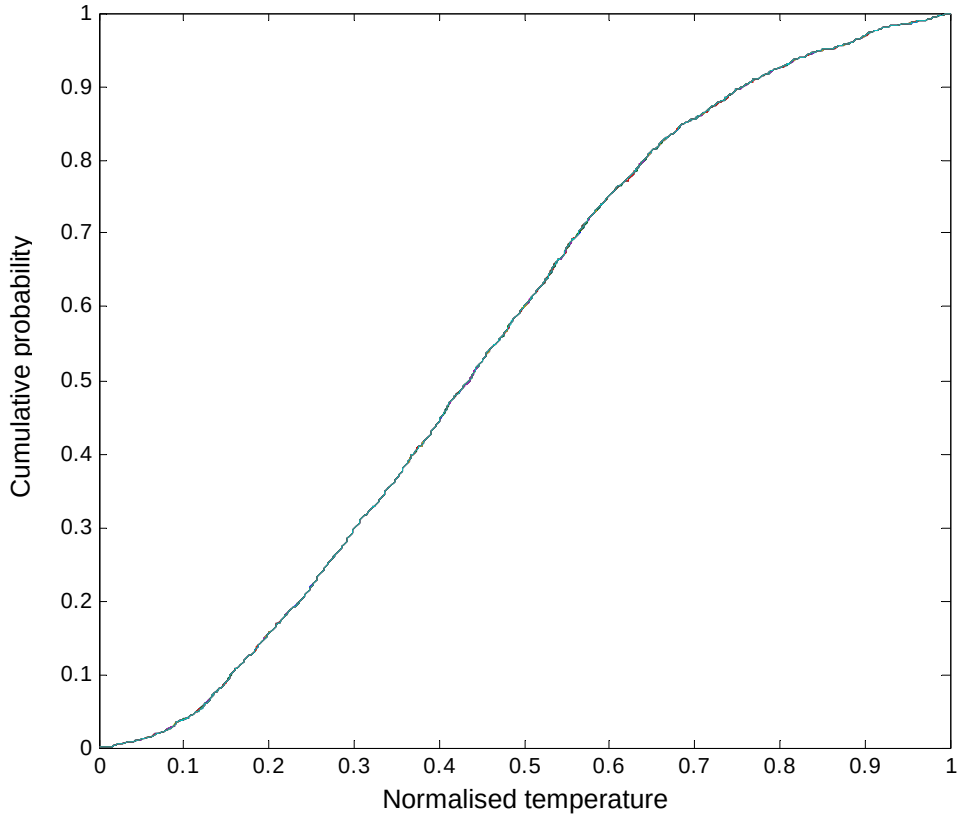

Figure 37: The normalised distribution functions for the values of temperature along $r = 0$. The lines for different values of z coincide.

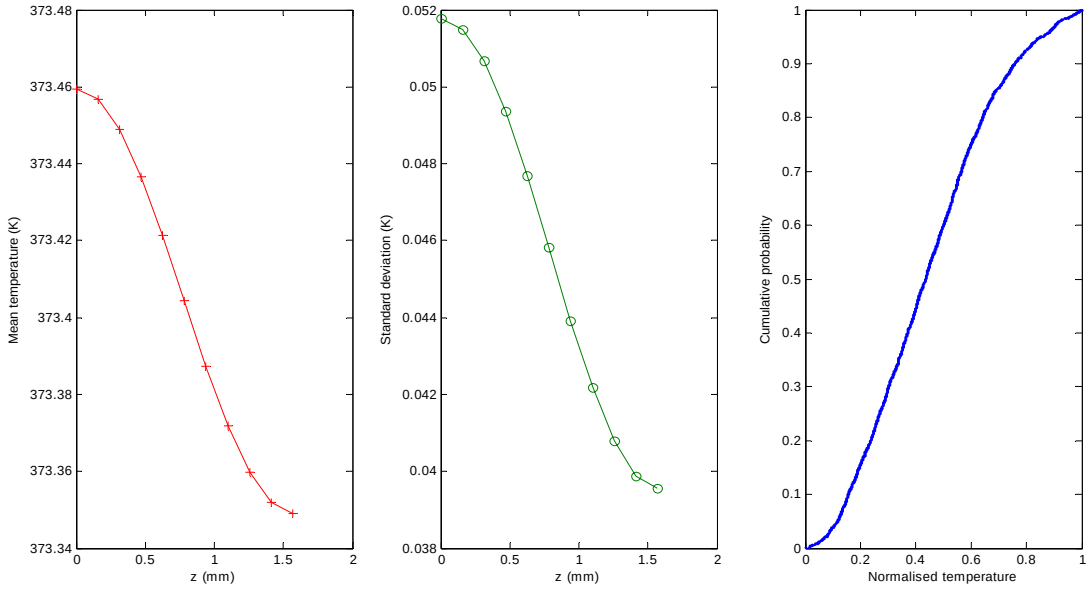

Figure 38: Variation of mean temperature (left hand plot) and standard deviation (central plot) along $r = 0$, and a distribution function for the value of the normalised temperature at each value of z.

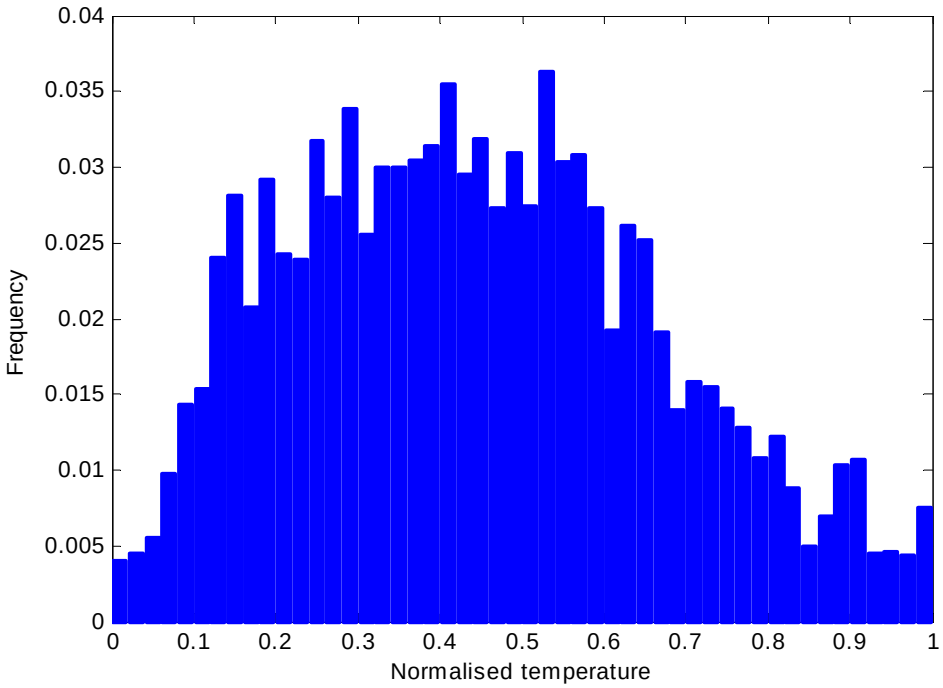

Figure 39: Calculated pdf for the value of the amalgamation of all the normalised temperatures along $r = 0$.

An alternative method of visualising cross-sectional plots is shown in figure 40. In this figure, the z axis represents the quantity that would be represented in colour in a cell plot (mean temperature in the plot on the left, standard deviation in the plot on the right). These plots are more difficult to interpret than the colour cell plots, but may be helpful to colour-blind people.

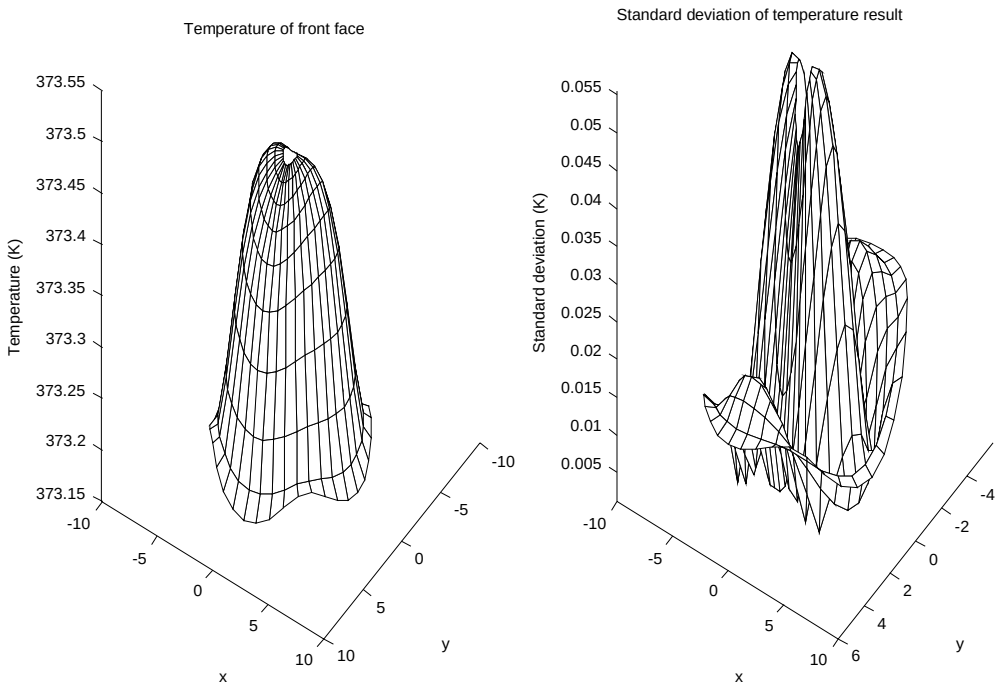

Figure 40: Cross-sectional plots represented as surfaces in three-dimensional space. The left-hand plot is the mean temperature of the front face and the right-hand plot the uncertainty associated with the temperature result.

4.5 Extensions and improvements

A wide range of visualisation techniques have been used to show the uncertainties associated with the values of the various output quantities produced by the model. Wherever possible the visualisations have been validated against alternative displays produced by other software packages. This section describes some of the extensions and improvements that may be of benefit to the user.

The multiple cross-sections as shown in figures 31, 32, and 35 can be difficult to interpret because it is not always easy to relate the images to their geometric positions. The interpretation of these images may be made simpler by animation. It would be straightforward to show the images in sequence in their positions to give a clearer idea of how they are related.

No attempt has been made to inter-link the different output quantities in this study, and in particular the uncertainty associated with the value of α has not been linked to the uncertainties of the temperature versus time curves from which α is determined. It may be useful to link these quantities, for instance by identifying the regions of the distribution function for the value of α that are generated by processing the endpoints of a 95% coverage interval for the temperature versus time curves.

It may be useful to be able to visualise the distribution functions for the value of the temperature at some point in time and space. This would be easy to do, provided that the user knew the coordinates of the point and time of interest. The main difficulty would be the need to store, process, and retrieve the data for 3 971 spatial points, 1 000 time steps, and 4 000 MCS trial runs.

Finally, it may be interesting to try to develop a parameter to describe the laser profile's departure from uniformity and see how that parameter relates to the uncertainties associated with the values of the output quantities.

5 Summary

Confidence in the correctness of conclusions drawn from visualisations of continuous models can be improved in two ways: by increasing the confidence in the correctness of the visualisations, and by enabling the user to visualise the uncertainties relevant to the problem of interest.

The correctness of the visualisations can be checked by considering the following points:

- Check the extrapolation, interpolation and smoothing (see section 2.1):
    - If the derivation of the discretised problem involved approximation functions such as polynomials, visualise the results using the same type of functions for interpolation.
    - Use the derivatives of the functions used for the interpolation of the solution. Visualise derivative quantities such as stresses or strains using the derivatives of the functions that were applied to the interpolation of the solution.
    - If the derivation of the discretised version of the problem did not involve interpolation, assume that the results are constant within an appropriate visualisation element.
    - Wherever possible, turn smoothing effects off to check that they are not disguising unexpected discontinuities in the results.
    - Compare visualisations using smoothed and unsmoothed results to increase confidence in the model results. Visualisations of a well-validated model with continuous results with and without smoothing should look broadly the same.
- Check for algorithm error (see section 2.2):
    - Before using a proprietary package for visualisation, test it with well-understood model results.
    - Wherever possible, use challenging test models (since simple cases will have been tested prior to the release of the software) that are typical of the full range of intended uses of the package.
    - During visualisation, carry out informal checks that the results “look right”.
    - Use a different package or an alternative method to visualise all or part of the results.
    - If the source code of the software used for visualisation is available, carry out a code review.
    - If the package being used has a website, check any user forums, helpdesk areas, or update section regularly for updates, bug fixes, and advice.
    - Report any bugs found so that developers can improve their products.
- Check the visualisation parameter settings (see section 2.3):
    - Read the package documentation to find out which parameters affecting

the display have default values.

- Consider which values of these parameters are most suitable for the results being visualised.
- If it is not clear which value is most suitable, adjust the value and rerun the visualisation to obtain an idea of the effect of the parameter.
- Consider limiting the visualisation to a small part of the model if displaying the full model would produce a misleading picture of the results.

The visualisation of the uncertainties can follow the same process guidelines as visualisation of measurement data, which are encapsulated in a six-step process:

1. Assemble the measurement data (in this case, uncertainties).
2. Define the problem that the visualisation is to solve.
3. Choose the hardware and software to be used.
4. Develop the initial solution.
5. Investigate how the visualisation can be made easier to interpret.
6. Validate the visualisation as far as possible.

This process is explained further in section 3 of the report, and some useful methods for visualisation of uncertain quantities have been described and demonstrated.

The case study in section 4 has applied this advice to a metrology problem requiring the determination of thermal diffusivity and involving different types of output quantity. A variety of visualisation methods have been used and the visualisations have been validated against alternative visualisation methods wherever possible.

Acknowledgements

This report has been produced as one of the deliverables of Software Support for Metrology (2001-2004) project 1.7 on Visual Modelling and Uncertainties. The work was funded by the Department of Trade and Industry's National Measurement Systems Directorate.

The author would like to thank John Redgrove and Bufa Zhang, both of the Centre for Basic, Thermal and Length Metrology at NPL, for their assistance with development of the case study and their provision of measurement data and input parameters. The author would also like to thank Maurice Cox of the Centre for Mathematics and Scientific Computing at NPL and Jeremy Walton of NAG Ltd. for their comments and suggestions.

6 References

[1] G J Lord and L Wright. Uncertainty Evaluation in Continuous Modelling. NPL report No. CMSC 31/03, December 2003.

Available from http://www.npl.co.uk/ssfm/download/documents/cmsc31_03.pdf

[2] T J Esward, G J Lord, and L Wright. Model Validation in Continuous Modelling. NPL Report No. CMSC 29/03, December 2003.

Available from http://www.npl.co.uk/ssfm/download/documents/cmsc29_03.pdf

[3] H R Cook, M G Cox, M P Dainton and P M Harris. A Methodology for Testing Spreadsheets and Other Packages Used in Metrology. NPL Report CISE 25/99, September 1999.

Available from http://www.npl.co.uk/ssfm/download/documents/cise25_99.pdf

[4] Sira Ltd. Visual Modelling and Data Visualisation.

Available from http://www.npl.co.uk/ssfm/download/documents/mpu_8-53-1.pdf

[5] W J Parker, R J Jenkins, C P Butler and G L Abbott. Flash method of determining thermal diffusivity, heat capacity, and thermal conductivity. *J. App. Phys.*, **32**(9), 1679-1684, September 1961.

[6] http://www.nag.co.uk/Welcome_IEC.html

[7] http://www.amiravis.com/

[8] J Gilby and J Walton. SS*f*M Good Practice Guide 13: Data Visualisation. June 2003.

Available from http://www.npl.co.uk/ssfm/download/documents/ssfmbpg13.pdf

[9] M G Cox, M Dainton, and P M Harris. Software Support for Metrology Best Practice Guide 6: Uncertainty and Statistical Modelling, March 2001.

Available from http://www.npl.co.uk/ssfm/download/documents/ssfmbpg6.pdf

[10] R E Taylor and L M Clark III. Finite pulse time effects in flash diffusivity method. *High Temperatures – High Pressures*, **6**, 65 – 72, 1974.

[11] M Sheindlin, D Halton, M Musella, and C Ronchi. Advances in the use of laser-flash techniques for thermal diffusivity measurement. *Review of Scientific Instruments*, **69**(3), 1426-1436, March 1998.

[12] J Walton. Visualisation Benchmarking: A Practical Application of 3D Publishing. Proceedings of Eurographics UK 1996, vol. 2, pp. 339-351 (1996). (also available from http://www.nag.co.uk/doc/TechRep/PS/tr9_96.ps)

[13] B S Everitt. Finite Mixture Distributions. Chapman and Hall, London, 1981.