Living with Uncertainty

Patrick Gaffney

Living with Uncertainty

Patrick Gaffney

Bergen Software Services International (BSSI)
Bergen, Norway
`pat@bssi-tt.com`

Abstract. This paper describes 12 years of experience in developing simulation software for automotive companies. By building software from scratch, using boundary integral methods and other techniques, it has been possible to tailor the software to address specific issues that arise in painting processes applied to vehicles and to provide engineers with results for real-time optimization and manufacturing analysis. The title provides the focus and the paper describes how living under the shadow of uncertainty has made us more innovative and more resourceful in solving problems that we never really expected to encounter when we started on this journey in 1999.

Keywords: electrocoat simulation, computational science, simulating painting processes

1   Prologue

- On the one hand, this is a story of doing software business with over 14 automotive manufacturers spanning the USA, Europe, and Asia during the period 1999 2011. The story is true, I have withheld the names of the guilty and innocent, and the story is ongoing.
- On the other hand, it is the story of how little influence we, as a community of computational scientists and numerical analysts, have had on engineers, those doing the daily business of manufacturing, wherever they may be.
- But mainly, the story is about VALIDATION the process by which a manufacturer thinks they can assess the quality of software.

In the attempt to assess the quality of software, manufacturers often forget that it takes a level of skill to understand how to use software properly. Without a clear understanding of the requirements of software and, more importantly, an understanding of the imperfections in the manufacturers own processes, it is impossible to come to a valid conclusion given the present scheme of VALIDATION undertaken by almost all automotive manufacturers.

This paper provides evidence to support these claims and gives some simple recommendations that make it easier for the manufacturer to make wiser decisions.

2   Electrodeposition

The software used to exemplify our thesis is one simulator from a family of software simulators called Virtual Paint Operations, or VPO^{TM} Software for short. This family contains software predictors for the processes involved in electrocoating a vehicle, namely drainage, electrocoating, void detection, drag-out, and baking. Everything said in this paper applies to all members of the VPO^{TM} Family but for the purposes of clarity the paper focuses on the simulator for predicting the electrodeposition of paint on a vehicle body, frame, or part. Automotive manufacturers and the coatings industry call this type of paint **e-coat** and it is the first coat of paint applied to protect a vehicle from salt spray induced corrosion and stone chipping.

The application of e-coat is through a process called electrodeposition that resembles the electro-plating technique that most people are familiar with. In this case, a vehicle is immersed in a tank containing the e-coat material and an electric potential (voltage) is applied to anodes positioned along the sides, and possibly the top and bottom of the tank. The resulting electrical current transports paint solids in the e-coat from the tank to the surface of the vehicle. There, and as a result of the hydrolysis of water that also occurs, the paint coagulates and adheres to the surface. As the paint adheres to the exterior surfaces of the vehicle, an electrical resistance builds and since electricity always seeks the path of least resistance the electrical current automatically flows to areas of the vehicle not previously coated. To access recessed and other areas of the body that are difficult to reach, engineers construct pathways, by manually placing holes in the body of the vehicle to allow current to flow through.

Areas of the vehicle with an inadequate coating of e-coat are liable to corrode. Therefore, the correct identification and subsequent choice of pathways are vital to the vehicles quality, its corrosion protection, and the reduction of the vehicles warrantee costs. Relying on manual experience and testing to insert a suitable configuration of holes, of the right size, number, and placement, is not a good idea, especially when software simulation guarantees success at far less cost to the manufacturer.

An *adequate coating of e-coat* usually means exterior surfaces of the vehicle have a thickness of paint, called film build, which for many manufacturers is around 20 microns while interior surfaces can have less thick film build, usually around 10 microns.

It is this difference in film build requirements between exterior and interior surfaces that contributes to the cost of applying e-coat because to ensure interior surfaces have an *adequate* film build often requires the exterior surfaces to have more than adequate coverage. One goal of any manufacturer is to avoid excessive coverage on exterior surfaces while maintaining adequate coverage on both exterior and interior surfaces.

Software experiments are an ideal way of investigating how to achieve this goal because, unlike physical testing, they can guarantee success. Physical testing has limitations —the amount of material wastage to perform the tests is bad for theenvironment, it is financially expensive, and because the number of physical

tests is limited, they are unsure of success. There are no such limitations when using software but what is required is an understanding of the physical process and the knowledge of how to use correctly a software tool that provides accurate predictions of this process.

3   Validation

Since electrical resistance increases with paint thickness, one way of modeling electrodeposition is to model resistivity correctly. This is the approach taken by the simulator of this paper. A paint model, whose parameters are all measurable and determined by a simple laboratory experiment, encapsulates the behavior of e-coat over time, and allows accurate predictions at any instant the vehicle is in the tank.

Verification of the simulator is the responsibility of the software developer and in the present case; verification is a continual process of daily life, beginning in 2003 and continuing today.

Validation is the process of comparing results from the simulator against an experiment performed by the manufacturer. Validation should be a contract between the customer and the developer, specifying precise conditions of the experiment to ensure comparisons are fair. This specificity often falls short because of uncertainties in the way the manufacturer conducted the experiment.

A manufacturer usually requests an experiment with the following components for validating electrodeposition software.

1. The manufacturer applies e-coat to a vehicle and then bakes the vehicle.
2. Either the manufacturer or a third-party cut up the vehicle (a process called teardown) and the manufacturer measures the thickness of electrocoat at measurement points chosen by the manufacturer.
3. The manufacturer compares the measured thickness of electrocoat to the predicted values obtained by the simulator.

On the face of it, these three components seem simple enough until one delves into the details of how and when the manufacturer conducts the experiment.

4   The Vehicle to E-coat

Western manufacturers usually intend to conduct item 1 of the experiment using a **production** vehicle, in other words, one that contains all of the reinforcements and other structural entities that are in the actual vehicle that a customer purchases. However, a manufacturers good intentions often go awry because the description of the vehicle that is passed to the software developer misses some important material, such as reinforcements, bolts, separators, and non-conductive material, the presence and position of all of which are necessary if the simulator is to predict accurately the electrodeposition behavior.

On the other hand, Asian manufacturers will often have a specific point in their design schedule for when they will perform a teardown. This teardown is for reasons other than validating electrodeposition software and can occur at a time when the vehicle does not contain all of its reinforcements or other structural entities. These manufacturers are reluctant, and often refuse, to perform a second teardown for validating specific software. Regardless of the reasons for this attitude, it behooves the software developer to accommodate the manufacturers wishes and in this case provide an alternative benchmark by which to compare the software predictions.

5   The Time Elapsed

Another uncertainty the developer needs to address is the time elapsed between item 1 above and the comparison mentioned in item 3. For Asian manufacturers this time can be on the order of years, because of their rigid teardown schedule, and this means it is doubtful the e-coat used in step 1 still exists. Without a sample of this e-coat, it is impossible to produce a paint model that encapsulates the behavior of the e-coat used to paint the teardown vehicle.

The manufacturers paint vendor has been known to suggest that the laboratory experiment required to determine the model parameters is performed using e-coat that has *"the same characteristics"* as that used to paint the teardown vehicle. This is tantamount to using fresh e-coat paint and is definitely not acceptable for validation because the resistivity of such paint tends to be much lower than the e-coat drawn from the manufacturers tank. Low resistivity implies high film builds and the use of fresh paint will therefore result in distorted and incorrect predictions from any simulator that uses properties of the e-coat paint.

When a sample of the e-coat used to paint the aged teardown vehicle does not exist then **validation** is inappropriate because it is impossible to reproduce in software the precise conditions under which the vehicle was painted.

Provided the CAD file of the teardown vehicle exists, it is possible to model the vehicle and if the e-coat used to paint the vehicle does not exist then one can use a sample of the e-coat used to paint the existing vehicles of the same model and type. In this way it is possible to evaluate *trends* in film build coverage over the 3-D model of the teardown vehicle.

6   The Measurement Points

The manufacturer selects points on the vehicle surface and takes measurements of the film thickness at them using one of the methods described in the following sub-section. However, recording the accurate position of these measurement points is not always easy for the manufacturer to remember and this is a potential major source of error when it comes to obtaining predictions from the simulator at the same measurement points. Very often, the person running the simulator receives a photograph or image showing the positions and then has

to interpret them for input to the simulator. Ironically, to produce a predicted result at an arbitrary point on the surface of a 3-D computational model requires a substantial amount of programming and technical effort, which is of little use when the manufacturer *forgets* the position of a measurement point.

7   Measuring Film Build

Whichever geometry the manufacturer uses as a benchmark against which comparisons are made of the software predictions, it is necessary to consider how the manufacturer measures the thickness of paint noted in item 2 above. There are essentially two methods in common use by well known reputable manufacturers. The first method consists of a special instrument designed specifically for this purpose and is usually considered to be accurate. The second method however, is manual and involves:

1. Measuring the thickness of the vehicle part.
2. Using sand paper to remove the paint layer and reveal the phosphate layer.
3. Measuring the thickness of the vehicle part again.
4. Computing the difference between the two measurements in 1 and 3.

Needless to say, item 2 of the second method is undesirable when considering using the measured values as sufficiently accurate to be part of a validation process. A small undue pressure on the sand paper can remove more of the paint layer and thereby result in an incorrect measurement of film thickness. However, this is unlikely to happen because usually the person performing the removal has acquired a deft touch from experience over many years. Passing on this experience to a new trainee will be problematical though.

8   Measured Film Build

An automobile manufacturer usually records measured film build by writing on the surface of the teardown vehicle the thickness value at the measurement point. A continuous sequence of voids usually indicates a problem in one or more stages of the electrocoating process as opposed to a problem with the e-coat itself.

The phosphate process, that the vehicle undergoes prior to electrocoating, cleans all surfaces of the vehicle from residual stamping oils and other contaminants, and deposits on the surface a protective layer of zinc and iron phosphate crystals. The color of adequate phosphate crystal deposits are dullish gray and rough and these facts help identify problem areas. For example, in the present case, by examining a color photograph of the teardown or better still, the actual teardown vehicle, the shiny color of the metal surrounding the voids indicates that the phosphate system has not been effective in removing stamping oils, and therefore the e-coat could not get to this surface of the vehicle causing the voids to occur.

An engineer rarely divulges information about areas of the vehicle with phosphate problems or grease problems and therefore the simulator has no way of knowing that contamination has compromised e-coat coverage in these areas of the vehicle. Consequently, predictions from the simulator cannot be correct in these areas of contamination.

Surprisingly, engineers rarely understand this fact and tend to expect the simulator to still produce accurate predictions in these areas even though they know the areas are contaminated. This is a great pity, for in the spirit of a validation *contract*, it is possible for the contractual partner —in this case the software developer —to provide **variations** of the simulator to take account of *known* uncertainties such as contamination from bad phosphating or grease problems, or voids due to air pockets. For example, in the case of air pockets, a variation of the simulator discussed in this paper can identify their location and predict electrodeposition in their presence, which makes it unnecessary for the engineer to know in advance where the air pockets may occur. The same is true of grease spots where new research at BSSI may lead also to the identification of grease spots a priori.

9   Towards a Turnkey HPC Solution

The attitude inherent in the last section is prevalent among engineers and is due in large part from a combination of their bad experiences with software and its providers, their lack of understanding of the power of mathematics, and their lack of technical experience with software simulation.

The latter is especially true in countries with a low standard of education in computational science. Unfortunately, there are few signs that this situation will improve. Many Universities no longer teach the basic numerical methods that are required to use software correctly, and employer-based training schemes have little understanding of non-business or technical courses. Therefore, to optimize the chances of a manufacturer using software simulation to improve their processes, it behooves the developer to write software so that the level of technical skill required to use it is minimal.

The software must include and convey the required expertise and knowledge in a way that makes the software easier to use and *control* by non-expert users. This means the developer must write the software to take account of imperfections in the input to the software and must convey enough information to the user in a way that allows them to either correct the input or to abandon the software run. If done correctly then, the software supplements the present lack of knowledge and expertise of an individual user in a way that opens up for the use of the software by a much wider community, which is a beneficial outcome for the manufacturer since a broader user space allows workers to migrate from one responsibility to another without the company having to sacrifice the use of the software.

Software produced in this way will not fail because invalid input is trapped and either fixed automatically or reported to the user in a way that will allow a correction.

Towards the goal of providing software that is significantly easier to use, the designers of all the VPO^{TM} Software made the following three conscious decisions:

1. A single user interface provides access to all VPO^{TM} Software.
2. The computational mesh of the vehicle is the same for all VPO^{TM} Software.
3. The system provides Software Generators to specify the input of tanks (VPO^{TM} Tank Generator) and bake ovens (VPO^{TM} Oven Generator) and to produce their respective 3-D models automatically.

Item 3 enables the manufacturer to specify dip tanks, e-coat tanks, and bake ovens without the need of CAD files for these structures, up to date versions of which often do not exist.

Item 2 ensures that only one mesh is required to model all the processes addressed by VPO^{TM} Software and thus the manufacturer only has to maintain a single mesh for modeling painting processes. It is also the only item for which a CAD file is required and the odds that this exists for a so-called Body In White (BIW) are greatly improved.

Item 1 provides access to the VPO^{TM} Software components from a single interface and this makes it easier for users to invoke the individual related tools for the topic of painting. A single user interface also makes it easier to *combine* the use of tools for modeling related processes, for example, to determine the effects on drainage from hole configurations used by electrocoating and vice versa.

10   Coping with Imperfections in the Input

The input to the simulator discussed in this paper requires the manufacturer to specify three datasets describing

(a) The e-coat material
(b) The e-coat tank, and
(c) The vehicle.

11   The E-coat Material

Characterization of the e-coat material is determined from a laboratory experiment, usually performed by the automobile manufacturers paint vendor, according to a set of rules specified by BSSI. As mentioned previously, a sample of e-coat drawn from the operational e-coat tank is crucial for the correct determination of the model parameters that simulate the e-coat used to paint the vehicle. This is the main contender for error in specifying the e-coat material input for the simulator, as shown by the following items.
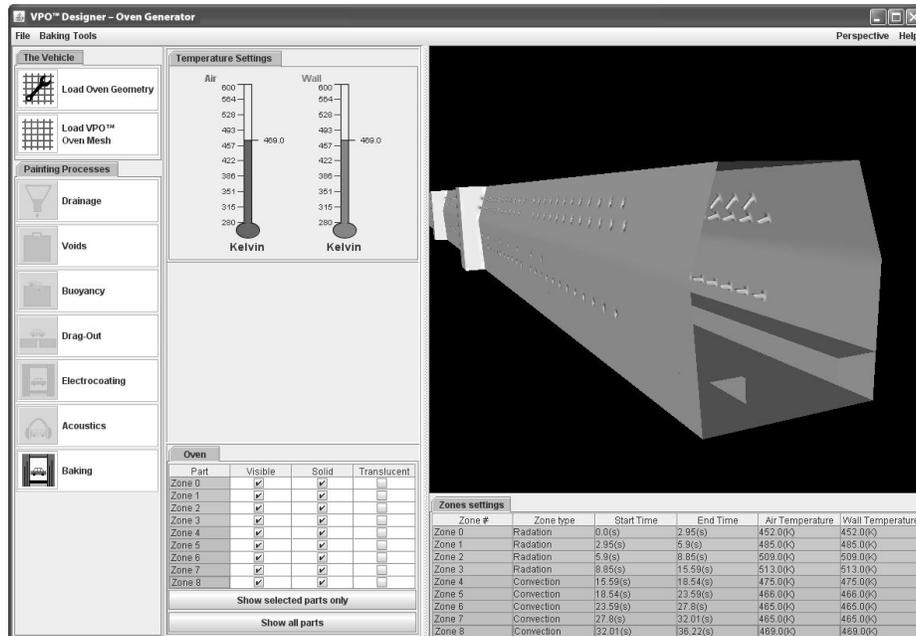
Fig. 1. A single user interface to all tools makes it easier to use the software. In this picture, invoking the Baking tool gives access to the Oven Generator.

1. The automobile manufacturers paint vendor asserts that material used for the laboratory experiment has the same characteristics as that in the operational tank, and this is blatantly impossible if the material is fresh or it has not been drawn from the tank.

2. The paint vendor does not take adequate precautions when transporting the e-coat from the operational tank, especially if it is possible for the e-coat to be frozen during transit.

BSSI has had to contend with these and other related issues from different automobile manufacturers at one time or another. Since the manufacturer or the paint vendor does not divulge these issues, it was necessary for BSSI to develop techniques, independent of the simulator, to analyze the laboratory results ahead of calculating the paint model parameters. These techniques, which, among others, identify the issues of 1 and 2 above, now form part of an *Input Verification* package that all input to the simulator must pass through successfully before running a simulation.

12   The E-coat Tank

The dataset that describes the e-coat tank includes information about the anodes contained in the tank: their type, the voltages applied to them, and the time the vehicle spends in front of them.

Initially, many automobile manufacturers thought they had to provide information about the *shape* of the tank and they instigated special laser measurements to determine this information for their tanks. However, BSSI has removed the burden from the manufacturer of providing this information because a built in *Tank Generator* includes the information from the provided dataset (b) to generate automatically a 3-D model of the tank suitable for the simulator.

Thus, the *Input Verification* package mentioned in the previous section only has to verify the anode information provided in the dataset. However, the veracity of this information relies heavily on the records kept by the automobile manufacturer and in some cases these can be questionable. Unfortunately, only a physical inspection of the e-coat tank can verify these records, when they exist, and this is something that BSSI has accepted they have to do for manufacturers.

13   The Vehicle

Every form of software simulation requires a computational mesh that captures the physical features of the simulated process. Unstructured (or irregular) meshes are usually required to model physical processes involving complex 3-D geometries that contain recessed areas and other places difficult to reach. This type of mesh is difficult to generate and requires experienced resources that most automotive companies do not possess. Therefore, BSSI incorporated a *Mesh-Suite* into VPO$^{\text{TM}}$ Software and made it accessible to the individual components of the software, since they all use the same mesh of the vehicle.

The starting point for this *Suite* is the native CAD file of the BIW, which is essentially the frame of the vehicle containing the reinforcements, bolts, separators, and non-conductive material present in the vehicle when it enters any of the tanks or the bake oven. For the electrodeposition simulator described in this paper, dataset (c) is this native CAD file. Unfortunately, for one reason or another, the automotive company rarely provides a CAD file of the BIW because one or more of the listed items is usually missing. The omission of any one of them will affect the predictions of e-coat but omitting one or more reinforcements will have the most serious effect.

The present validation process defined by most automobile manufacturers makes it very difficult to ascertain when the supplied CAD file differs from the vehicle used to provide the benchmark measurements of film build. However, in some cases where a manufacturer is willing to share these measurements with the software developer it is possible for BSSI, using techniques it has developed, to determine where differences occur in the geometry of the vehicle and then together, the manufacturer and BSSI, can evaluate the situation and agree on how to proceed. This is the ideal way of working but is all too often rare.

14   Observations

The standard validation process, used by most automobile manufacturers, can never work because a painted vehicle is never free from imperfections. Unless the manufacturer is prepared to accept this fact and to point out these imperfections BEFORE making simulations, predictions can never be totally correct. Identifying imperfections makes it possible to use a simulator, modified to take imperfections into account.

In an ideal world, validation is a contract meant to assist both the manufacturer and the developer. Unfortunately, for one reason or another, most manufacturers, but not all, view the validation process as adversarial. Until this attitude changes, manufacturers will not reap the benefits of using software to minimize physical experiments.

To optimize the chances of a manufacturer using software simulation to improve their processes, the developer must write software so that the level of technical skill required to use it is minimal. Since the use of the software requires the INPUT to be correct in order for the OUTPUT to be likewise (garbage in garbage out), and since the input requires three pieces of information that may be contaminated with imperfections, the software must be written to recognize this fact, inform the user, and supply enough information for corrections to be made. To enhance this process and to foster a good working relationship, BSSI has instigated the following new validation procedure, which is a modification of the standard one.

At the manufacturers premises anywhere worldwide, BSSI will inspect the vehicle while it is painted and baked, noting:

- Areas where grease or phosphate problems may have occurred prior to applying e-coat
- Any other imperfections in the vehicle or in the operations
- How the manufacturer takes measurements and transfers them to BSSI.

Of course the implementation of this new procedure will run into trouble when the manufacturer wishes to use a teardown vehicle significantly older than the inspected ones. However, provided the CAD file of the teardown vehicle exists, it is possible to model it. Depending on the age of the teardown, the e-coat used to paint it will most likely not exist and therefore validation is not appropriate. Rather, the only recourse is to evaluate trends in film build coverage over the 3-D model of the teardown vehicle.

15   Conclusions

This paper gives a brief summary of the discoveries made by a naive software developer when trying to interest automotive companies in the use of advanced mathematical simulators of painting processes.

Naiveté on the part of the developer is from two perspectives: (1) the assumption (unjustified) that the manufacturer's people were knowledgeable in

the basics of modeling and understood the physical process and (2) the reality that as a community of numerical analysts and computational scientists we have failed to communicate with the very people we are in business to serve.

In the first case, the manufacturers technical people, almost universally, are deeply skeptical, for whatever reason. In the second case, we have not been successful to the extent that nowadays people who use software have little or no concept of the importance of **input** to that software —the old adage of *garbage-in garbage-out* is just that: a standalone saying that almost nobody thinks about when using software.

Input to the software comes from the manufacturer and it has been our unfortunate experience that almost no single one of them has got all aspects of the input correct when requesting a validation.

One way of avoiding this problem is for the software to diagnose invalid input, correct it automatically where possible, and if not then convey to the user what they need to do to correct the input. Another way is to remove the responsibility from the user of providing aspects of the input that they are incapable of providing correctly. The mesh of the vehicle is one example of both these situations.

For instance, the generation of a computational mesh requires that the elements of the mesh (for example, triangles) are of a size, and placement that accurately reproduces the simulated physical process. For example, to model accurately the flow of paint between two plates, the number of triangles and their size on both plates will depend on the gap between them. It should be intuitively obvious that, if one is to obtain an *accurate mesh* that provides accurate predictions from the simulator then many smaller triangles will be required on each plate for a gap of 1 mm than for a gap of 3 mm or greater. In this context, an accurate mesh is a mesh that has converged in space and mesh builders unfortunately often overlook this aspect of mesh generation. It is not uncommon to find the same mesh spacing used for gaps of 1, 2, 4, 8, and 10 mm and accompanying wonderment that the predicted results look nothing like what they should be!

One way to avoid this problem is for the simulation system to generate the converged mesh automatically without input from the user. Another way is for the simulation system to monitor the mesh input by the user and determine if it has converged in space or not. If it has not converged then, before continuing with the run, the system informs the user that the mesh is too crude to get accurate predictions and suggests where more refinement may be necessary.

I am definitely not advocating an artificial intelligence or expert systems approach to using the type of computational simulator addressed in this paper. What I am advocating is much simpler: **get the input right and the code will work**.

Since we know exactly the three pieces of information needed for the input then we know what getting it right means. The three pieces of information come from three different sets of people:

1. CAD people know CAD but not meshing - therefore let them provide the CAD and the simulation system will do the meshing properly. The Mesh-

Suite mentioned above without input from the user produces an irregular 3-D mesh that is both mathematically correct and converged in space.

2. Paint vendors know paint - therefore let them supply what is required and the simulation system will monitor and analyze their data to see that it is correct.

3. The tank operators - they may delegate the operation of the tank to the paint vendors, who, in turn, may not record the settings of that tank. Unfortunately, only a physical inspection of the e-coat tank can verify these records, when they exist, and this is something that BSSI, as a developer, have accepted to do for manufacturers.

In the paper, I have tried to make it clear that to perform computational simulation successfully the software must minimize the information required of the operator or engineer because they cannot easily get this information. Instead, the software must be written in a way that either avoids the user having to provide any information or that provides questions that the user is able to answer.

As mentioned in the prologue, this process is ongoing. One tool that is essential to ensure accuracy of code changes and to assist with parallel programming implementations is Brian Smiths Test Harness (TH) [1]. This tool is an indispensible part of working towards a turnkey HPC solution.

16   Afterword

"We are the heirs to a tradition that has left science and society out of step." – Jacob Bronowski, 1951

Over 60 years ago, the eminent scientist and philosopher Jacob Bronowski made these remarks, in his seminal work *The Common Sense of Science* [2]. His words are as true today as they were then.

Moving to today, I believe the further remark is unfortunately, no less true:

"We are the heirs to a tradition that has left computational science and society out of step." – 2011

I believe both statements are true for the same reasons: a lack of understanding between both parties in each case, borne of a tradition where the practitioners of Science and Computational Science tend to work in an environment populated by the elite. This fact is even more apt today where computational scientists have allowed the gap between themselves and their users to grow.

We cannot hope to stem or reverse this situation and therefore we have to accommodate our knowledge and expertise in our software so that society at large could find it as easy to use as their other *Apps* on their so-called *smartphones.*

References

1. Smith, Brian T.: Measuring Uncertainty in Scientific Computation Using Numerica21's Test Harness. In: this volume (2012)
2. Bronowski, Jacob: The Common Sense of Science. Harvard University Press, Cambridge, MA (1978)

DISCUSSION

Speaker: Patrick Gaffney

Felipe Montes : You mention a series of problems that affect the validation procedures of the simulation code. Those problems appear also in real life experiments (not simulations). So how do car companies reconcile and argue against simulation while real experimental results show the same problems as the model simulation?

Patrick Gaffney : Experiments regarding electrodeposition (the subject of the talk), baking, drainage, or void detection are limited to the extent that experimenting with each of these on full vehicles in real operational situations would require an interruption to the production line and that is a no-no.

Automotive Manufacturers who conduct real life experiments for electrodeposition, usually perform them under controlled laboratory conditions using simplified geometries and they do not exhibit the problems discussed in the talk and presented in the paper. It is only when scaling up to real world operating circumstances that things go wrong, and they do so primarily because one obtains the input for the code from several disjoint entities, many of whom are not privy to the reasons, nor do they understand, why recording and monitoring of conditions are necessary.

William Welch : How are OEMs using the simulator once verified? For example, are they optimizing hole positions? This looks like a high-dimensional optimization.

Patrick Gaffney : Once verified, OEMs use the software for a variety of purposes, primarily at the Design and Re-Work stages, the latter for reducing the costs of re-tooling.

In the Design stage, OEMs are using the software to determine configurations of pathways (not necessarily holes) to enhance the flow of electric current and hence e-coat coverage. For holes and other pathways we have provided software that makes the task of running different scenarios easier.

Similarly, for optimizing hole or pathway positions in a vehicle part, the combination of software tools together with 3-D animation to see how current flows, is sufficient. High-dimensional optimization is not involved nor is it necessary if one needs an answer quickly.

Brian Smith : Your model is that the cars are manufactured to be consistent from one vehicle to the next–e.g., the holes are in the same place, the caps are the same shape. Can you conceive of assessing the "vehicle" so that your software can adjust the painting parameters to the "new" conditions?

Patrick Gaffney : Excellent question! Yes, we can absolutely conceive of this situation, especially with the prospect of using fast GPUs to run the models "on-the-fly" and this is something we are presently investigating.

William Kahan : Can universities, that offer "computational science" minors with engineering (but these courses are mostly about how to use software packages) be realistically expected to teach the kind of fundamentals about numerical methods for PDE's, whose lack you bemoaned to students whose syllabi are already packed tight?

Patrick Gaffney : A "computational science" minor that only teaches how to use software packages is itself performing a disservice because I would contend that in order to use a software package correctly a user should have some basic knowledge of numerics, at least they should understand how important it is to get the input correct.

This is not going to happen in the West, and therefore one cannot rely upon Universities to fulfill this role.

However, the OEM has a responsibility for ensuring proper training of its staff and technical people. For those OEMs, I would suggest that they include a very basic course that exemplifies why the INPUT to the software is so important to get right.

In our experience, it would be beneficial if engineers and those leaving University had knowledge and experience with the following basic items.

- Numbers
- Precision and significant figures
- Floating point arithmetic
- IEEE Standard
- The processes of rounding and truncation
- Rounding error

- Differences and divided differences
- Iteration methods
- Convergence
- Continuity
- Limiting processes

- Basic numerical integration

Anyone wishing more details should contact me directly.