Computational Solution Verification Applied to a Thermal Model of a Ruggedized Instrumentation Package

S.N. Scott, J.A. Templeton, J.R. Ruthruff, P.D. Hough & J.P. Peterson
Sandia National Laboratories, Livermore, CA 94551

Abstract

This paper details a methodology for quantification of errors and uncertainties of a finite element heat transfer model applied to a Ruggedized Instrumentation Package (RIP). The proposed verification process includes solution verification, which examines the errors associated with the code's solution techniques. The model was subjected to mesh resolution and numerical parameters sensitivity studies to determine reasonable parameter values and to understand how they change the overall model response and performance criteria. To facilitate quantification of the uncertainty associated with the mesh, automatic meshing and mesh refining/coarsening algorithms were created and implemented on the complex geometry of the RIP. Similarly, highly automated software to vary model inputs was also developed for the purpose of assessing the solution's sensitivity to numerical parameters. The model was subjected to mesh resolution and numerical parameters sensitivity studies. This process not only tests the robustness of the numerical parameters, but also allows for the optimization of robustness and numerical error with computation time. Agglomeration of these studies provides a bound for the uncertainty due to numerical error for the model. An emphasis is placed on the automation of solution verification to allow a rigorous look at uncertainty to be performed even within a tight design and development schedule.

Keywords: Uncertainty Quantification, Verification, Mesh Resolution, Numerical Parameters

1 Introduction

In any numerical model, uncertainties due to the pedigree of the software and numerical parameters make the calculation of an exact answer for a given problem impossible. Understanding the uncertainty associated with numerical errors is crucial to make meaningful predictions or comparisons to experimental results. Quantification of these errors is done by preforming code and solution verification. The purpose of solution verification is to quantify the uncertainty induced in simulation results by the algorithms used to approximate the solution. Ideally, each result used in the assessment would be based on an analytical solution of the governing partial differential equations (PDEs) in the geometry of interest, but solving these equations by hand is often not possible. Computation offers a means to approximate these solutions, but introduces errors by relying on a discrete approximation of the underlying PDEs. It is therefore important to know how much the calculated solution may differ from the exact solution. Quantification of this uncertainty involves varying numerical parameters such as mesh resolution, time integration methods, etc. The resulting uncertainty is used to provide simulation error bounds when making predictions or comparing to experimental data. This work summarizes the solution verification of the thermal model of the Ruggedized Instrumentation Package (RIP).

2 RIP Definition

The RIP is an assembly of electronics, batteries, and circuit boards packaged in metal housings as seen in Figure 1. The goal was to create an FEA thermal model using SIERRA Thermal/Fluids code [1] in order to monitor the temperature in specified locations to ensure that electronics remained in their operating temperature range.
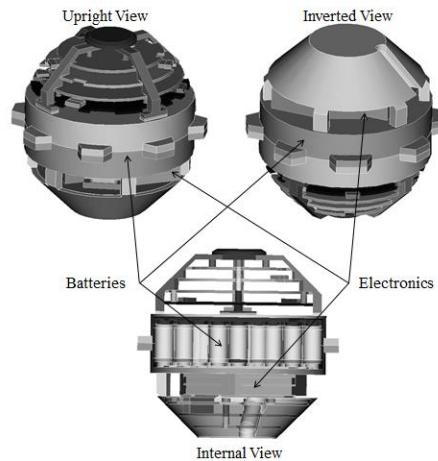

Figure 1: Ruggedized Instrumentation Packager (RIP) in three views with heat sources labelled.

The RIP is approximately 0.15 meters in diameter. Heat sources in the RIP are 48 2/3 A-cell batteries and two electronic packages (EP). In the simulation, the RIP is heated by these electronics in a vacuum (no convection) for one hour. Each EP produces 17 Watts of heat, however while EP1 is on for the entire hour, EP2 is only turned on in the final minute of the simulation. The batteries each produce 0.3 Watts and are constantly producing heat throughout the hour. There is no contact resistance specified in the model, other than between the EPs and the battery housing. This represents a thin thermal gap pad, which was difficult to mesh. The temperature was monitored in six locations (Figure 2), corresponding to the location of thermocouples (TC) in experiments that will be performed on the RIP. TC 1, 2, 5 and 6 are located on the battery housing and TC 3 is located on EP1 and TC 4 on EP2.
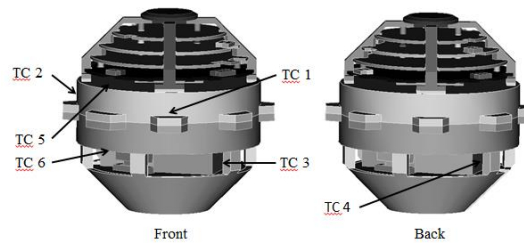

Figure 2: Locations where temperature is monitored

2.1 Geometry

The analysis team received production style ProEngineer [2] (computer aided design software) models of the RIP. To prepare this model for meshing, many of the details needed to be removed, such as screws, fillets, threaded inserts, cables, and electrical connectors and features used for tolerancing, such as small gaps. Once these changes were made, the model was exported as a STEP file and imported into CUBIT [3], a geometry creation, manipulation, and mesh generator created by Sandia National Laboratories. When the geometry was completed, the model was comprised of 107 blocks.

3 Mesh Resolution Study

When performing any computer analysis of PDEs, the continuous mathematical problem must be converted into a discrete representation. Doing so requires the generation of a mesh, which discretizes the continuous model into finite elements, for solving and storing the approximate solution. An unavoidable consequence is that information is lost relative to the original continuous problem, resulting in uncertainty. The quantification of the sensitivity of the solution to the mesh size is accomplished through a mesh resolution study involving a family of topologically similar meshes with a range of characteristic

length scales. The finest mesh in the set is considered to be the 'true' solution, because as the mesh size decreases, the mesh-based approximations to the continuous derivatives also improve. When the coarser meshes are compared to the 'true' solution, the error associated with discretion can be assessed.

3.1 Original Mesh Creation

One of the reasons CUBIT was chosen as the mesh generator for this project was that it has a built in Python (open source scripting language) interface. A script was created using Python that would automatically generate a tetrahedral mesh for a complex geometry. To handle models, such as this one, where the blocks varied drastically in size, this script was built with 'intelligence' to choose appropriate mesh sizes for each block. The meshing script worked according to the following algorithm:

- Obtain geometric information about the blocks, surfaces, and curves
- Use curve information to set a global largest and smallest mesh size by evaluating the smallest and largest curves in the model. The default is to set the smallest mesh size as half the length of the smallest curve and the largest as 200 times the smallest mesh size (the latter being user-specified).
- Surfaces are ranked from smallest surface area to largest.
- For each surface, beginning with the smallest, a mesh size is determined by dividing the length of the smallest curve in half. The mesh size is compared against the longest curve on the surfaces, to ensure that surfaces with no small curves still have an appropriate mesh size. Next, a check is conducted to make sure the mesh size falls between the globally allowed values. Finally, the surface is meshed.
- The blocks are ranked by volume from smallest to largest.
- Starting with the smallest block, the volume is meshed. Next, a list of adjacent blocks is created and ranked from smallest the largest. The smallest block in that list is then meshed and the process begins again. If the smallest block is already meshed, the program moves down the list until it finds an unmeshed block.

If at any point in the program an unacceptable mesh is created (CUBIT automatically checks the quality of the mesh it creates), the program exits, providing the user information about the error. The program can then be restarted where it left off. Using this program, a mesh of 3,647,629 elements was created in approximately 10 minutes. This mesh will be referred to as the nominal mesh.

3.2 Refined Mesh Creation

A second Python program was created in order to create a family of meshes for the mesh resolution study. This program, referred to as 'remesher', is meant to be applied to an existing mesh, in this case the nominal mesh, and a user supplied scale factor by which the mesh is to be changed. The program's

objective is to create a new mesh, topologically similar to the original, but with a smaller or larger number of elements. The program operates in the follow way:

- Obtain geometric information about the mesh on blocks, surfaces, and curves in the model and store it. The mesh is then deleted and reset.
- The curves are ranked from shortest to longest. For each curve, beginning with the shortest, a new mesh size is created by multiplying the old mesh size by the user specified scale factor. An optional check that the new mesh size does not violate a user supplied maximum mesh size is then conducted. Finally, a mesh size is set for the curve and the curve is meshed.
- The surfaces are ranked from smallest surface area to largest. For each surface, beginning with the smallest, the surface is meshed.
- The blocks meshed in the same manner as the nominal mesher.

Using this program eight meshes were created. The meshing time ranged from under 5 minutes for the coarsest meshes to over an hour for the finest. Table 1 gives information about these meshes and Figure 3 shows select meshes. Their names are derived from the change in number of elements from the nominal mesh (3,647,629 elements).

Table 1: Meshes generated by remesher program

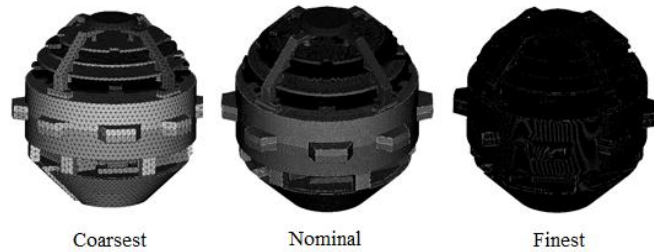| Mesh Name	# elements	Mesh Name	# elements
Coarse 16x	228,485	Fine 2x	7,380,550
Coarse 8x	454,463	Fine 4x	14,622,783
Coarse 4x	919,198	Fine 8x	29,336,251
Coarse 2x	1,845,807	Fine 16x	58,802,650

Figure 3: Meshes generated by remesher program

3.3 Mesh Resolution Study Results

The 16x mesh was taken as the 'true' solution because it was the most refined mesh generated. As it is not actually the true mathematical solution, mesh convergence can only be assessed in the Cauchy sense, as discussed in Hughes [4]. Thus, the rate of convergence can be identified, but the difference between the nominal and 16x mesh only provides an estimate for the absolute difference between the numerical results and analytic solution. Two standard norms were

used to assess the rate of convergence: the L_2 and L_∞ norms. The L_∞ norm is the absolute maximum difference between the two solutions, also normalized by the fine mesh. This norm is mathematically defined as (in 1D):

$$L_\infty = \lim_{n \to \infty} \frac{\left(\int_\Omega (T_F(x) - T_C(x))^n \, dx\right)^{1/n}}{\left(\int_\Omega T_F^n(x) \, dx\right)^{1/n}} \tag{1}$$

where Ω is the domain, $T_F(x)$ is the temperature of the fine mesh at location x and $T_C(x)$ is the temperature of the course mesh at location x. The L_∞ norm is expected to converge as h^{-1}, where h is the mesh length scale. It provides a measure of the worst-case local error that can occur and is useful for verifying that lower dimensional parts of the mesh are accurate, such as faces and contacts. In contrast, the L_2 norm measures the mean square error between two solutions, normalized by the fine mesh, and is expected to converge as h^{-2}. It is useful for assessing the overall quality of the solution.
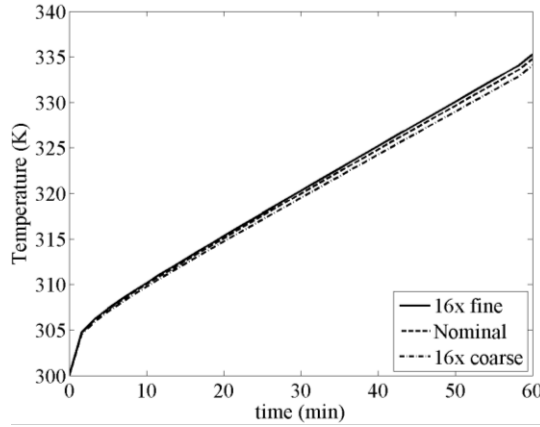

Figure 4: Time vs Temperature for the TC 3 for the 16x fine mesh, nominal mesh, and 16x coarse mesh

Qualitative trends are presented in Figure 4 showing the differences in the temperature fields at a location of interest for three different mesh resolutions. The maximum temperature difference between the 16x fine mesh and the 16x coarse mesh occurs at 60 minutes and is 1.24K. The temperature difference between the 16x fine mesh and the nominal mesh also occurs at 60 minutes and is 0.5K. This indicates that the solution is converging, which is further supported by the quantitative results for the L_2 and L_∞ norms shown in Figure 5. When plotted on a log-log scale, the slope of the best fit lines in this figure represents $-k$ in h^{-k}. The series of tested meshes have an L_2 convergence rate of -2.65 and a L_∞ convergence rate of -1.07. While these convergence rates are near the expected value, using a finer mesh as the 'true' solution could bring these values closer to the expected values. Based on the results, it was determined that the nominal was the optimal mesh, since it is within the linear convergence regime, but at the coarse end, allowing for shorter computation times.
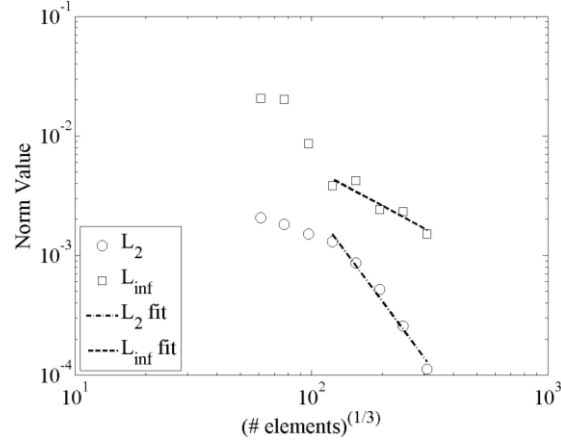
Figure 5: L_2 and L_∞ norms with best fit lines (L_2 best fit slope = -2.65, L_∞ best fit slope = -1.07)

In addition to the model based approach shown in Figure 5, a block by block L_2 and L_∞ analysis was performed. While convergence theory only holds for the entire model, examining the blocks can provide insight about local mesh quality. Figure 6 shows a histogram of the convergence rates of the blocks. For the L_2 convergence rate, most of the blocks fell near the model L_2 rate of -2.65. This is expected, since the L_2 norm is an average convergence rate. In the L_∞ case, most of the blocks fell near -2.5, far from the model rate of -1.07. Since the mesh as well as the temperature inputs from the surrounding blocks is changing, it is reasonable that most blocks have a faster convergence rate than the global rate.
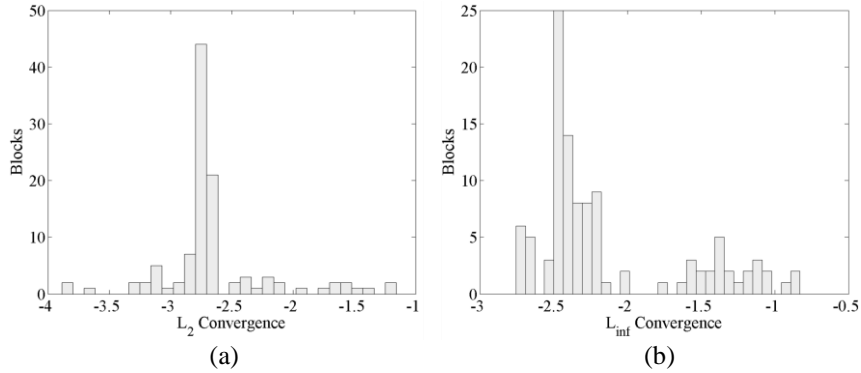

Figure 6: Histogram of the (a) L2 and (b) L_∞ norms for each block in the RIP

4 Numerical Parameters Study

In any simulation, there are a number of parameters that control the behaviour of the underlying numerical algorithms and hence can have an impact on the model

results. Numerical parameters in the RIP thermal model fall into three general categories: linear solver settings, time integration, and radiation solver parameters. The linear solver settings specify the preconditioning and solution methods, and solution tolerance. The time integration algorithm includes parameters governing the number of non-linear iterations and their convergence tolerance, the order of accuracy, and the use of lumped or consistent mass matrices. Finally, there are several parameters governing the performance of the radiation solver.

4.1 Approach

For this study, the Latin Hypercube sampling (LHS) capability in Dakota was used to sample the parameters that control the numerical behaviour of the model in question and to generate and record model responses of interest as well as the time required to complete computations. Dakota is a software package designed and developed at Sandia National Laboratories to perform advanced parameter space exploration of computational models [5, 6, 7]. Included are a wide range of algorithms for sensitivity analysis, uncertainty quantification, optimization, and parameter estimation. LHS is a stratified sampling approach in which the parameter space is partitioned in such a way that there are p bins of equal probability for each parameter across its range of possible values, where p is the number of samples. The size of the bins is determined by the probability distribution associated with the parameter. This results in p^n bins for an n-dimensional parameter space. Samples are randomly placed within the bins such that for all one-dimensional projections of the p samples and bins, there will be one and only one sample in each bin.

The data generated by the LHS study was analysed using the R open-source statistical software [8]. Analysis of variance (ANOVA) and correlation analysis capabilities in R were used to determine which parameters had the most significant effects on the model responses and computation time [9, 10].

ANOVA allows for the decomposition of the variance seen in the model response into a function of the parameters. Statistical tests then allow for the identification of the parameters that have the most influence on the response. The key quantity of interest in ANOVA is the p-value, which represents the probability that a more extreme response would be observed despite the parameter having no effect on the response. Therefore, the lower a parameter's p-value, the stronger the evidence that it is influential.

To determine correlation coefficients, the Pearson and Spearman correlation were considered. The former measures the direction and strength of the linear relationships between responses and parameters. The latter is a measure of the monotonicity of the responses with respect to the parameters. Values for both range from [-1,1], with the endpoints representing the strongest relationships.

Aggregate numbers and statistics such as those described above can often hide model behaviour. With the correlations in particular, since they capture only linear and monotonic relationships, it is possible that cancellation effects can result in low values where in fact there might be some nonlinear influence.

For that reason, the effects of parameters on the responses were also visually inspected using scatter plots.

4.2 Results

For the verification study, 27 input parameters were examined and the temperature at the thermocouples and the simulation time was monitored. In addition, the results were compared to the results of a nominal run, and L_2 and L_∞ norms were calculated. The study took approximately a day to set up, and 18 hours to run (on 1120 processor cores running 70 simulations simultaneously). Using the p-values and correlation coefficients as a guide, all values below (Table 2) were considered to be significant for this study.

Table 2: Significant parameters

Parameter Name	Description
Maximum time step	The maximum amount of time allowed between each time step. The allowed range is 10 to 200, nominal is 100.
Residual norm scaling	The method for measuring the error. The choices are none or RHS, which normalizes it against the right-hand side. The nominal is none.
Time integration method	The order of accuracy in for time integration. The choices are first order and second order. The nominal is second order.

Using this data it was determined that many of the nominal settings for the simulation were optimal. For example, when residual norm scaling was set to RHS, the average runtime was higher than with none. Since there was little appreciable change in the results data when residual norm scaling was changed, the choice of using the faster setting is preferable.

For maximum time step, it is assumed both the simulation time will increase as the time step decreases and that solution accuracy will decrease as time step increases. As seen in Figure 7(a), simulation time increased rapidly as the maximum time step was below 50. Figure 7(b) likewise shows a decrease in the spread of the data as the time step decreases.
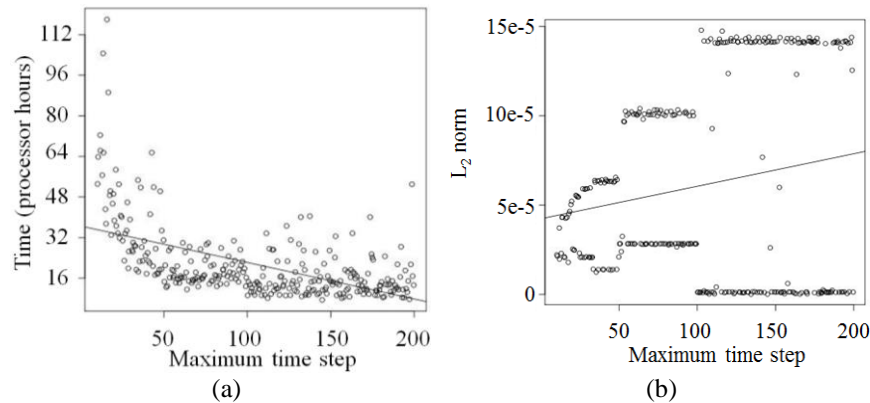

(a)   (b)

Figure 7: (a) Simulation time vs maximum time step (b) L2 norm vs time step

The thermocouples also showed a response to changing the maximum time step. Representative thermocouple responses are shown in Figure 8. TC 1, 5, and 4 had the strongest response, where TC 1 and 5 had similar plots, while TC 4 showed the opposite trend in terms of average temperature. However, both plots show the spread in the data decreases as the maximum time step decreases. This indicated that the maximum time step needs to be set to 50 or less to reduce error. When time integration method (Figure 9) was set to second order, the simulation time decreased and the spread in the temperature data from the thermocouples (except TC 4) decreased. The spread in the TC 4 data may be caused by the variation in the maximum time step.
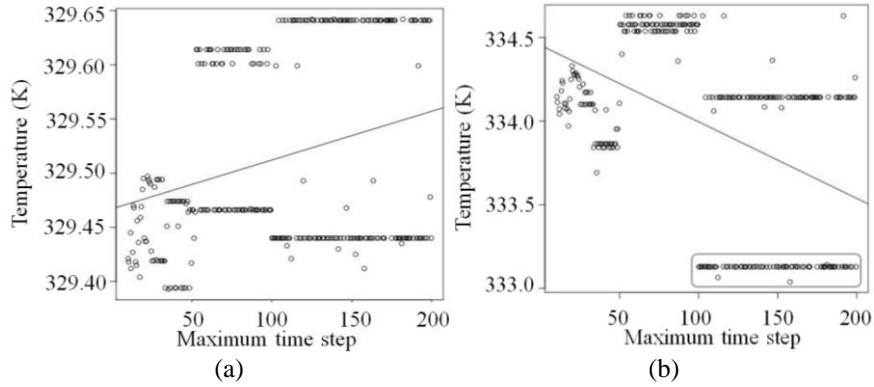

Figure 8: Temperature vs maximum time step for (a) TC 1 and (b) TC 4. Boxed data points refer to boxed data points in Figure 9(b)
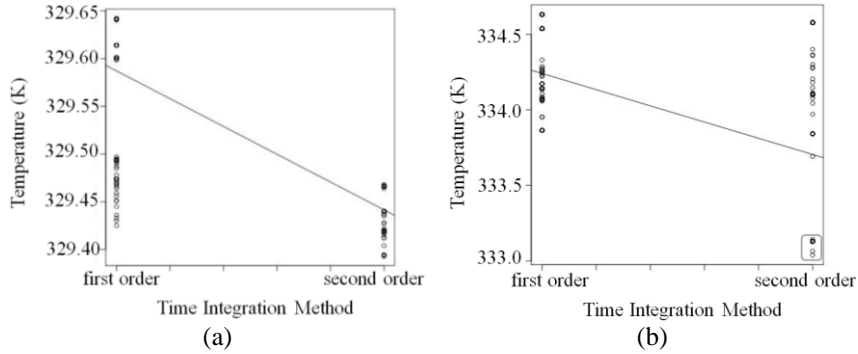

Figure 9: Temperature vs integration order for (a) TC 1 and (b) TC 4. Boxed data points refer to boxed data points in Figure 8(b)

5 Uncertainty Quantification

The data from the mesh resolution and the numerical parameters studies used to quantify the uncertainty. The maximum difference in temperature between the

16x fine and the 16x coarse meshes is a conservative estimate of the uncertainty with regards to the mesh. Table 3 shows these results.

Table 3: Uncertainty for each thermocouple with regards to mesh

Thermocouple	Temperature Uncertainty	Thermocouple	Temperature Uncertainty
TC 1	± 1.10 K	TC 4	± 1.24 K
TC 2	± 1.10 K	TC 5	± 1.08 K
TC 3	± 1.24 K	TC 6	± 1.18 K

In order to determine the uncertainty from the mesh for all points on the RIP, not just the tracked locations, the L_2 and L_∞ norms are employed. Since the norms are a normalized measure of the temperature difference, when multiplied by 100 it can be used as a percent error. Using the L_∞ norm is a more conservative estimate than the L_2 norm, as L_∞ indicates the maximum variation. Although Table 4 reports the uncertainties derived from L_2 and L_∞ for all meshes, since the mesh resolution study showed that the nominal mesh was optimal for our problem, a conservative estimate for the error introduced by the mesh at any point in our problem is ± 0.42%. In addition, the L_∞ correlates well with the temperature difference based approach (335K * 0.42% = 1.4K).

Table 4: Uncertainty associated with each mesh

Mesh Name	L_2 Error	L_∞ Error	Mesh Name	L_2 Error	L_∞ Error
Coarse 16x	± 0.21%	± 2.05%	Nominal	± 0.09%	± 0.42%
Coarse 8x	± 0.18%	± 2.02%	Fine 2x	± 0.05%	± 0.24%
Coarse 4x	± 0.15%	± 0.86%	Fine 4x	± 0.03%	± 0.23%
Coarse 2x	± 0.13%	± 0.38%	Fine 8x	± 0.01%	± 0.15%

To quantify the error introduced by the numerical parameters, the largest spread induced by a chosen parameter is evaluated. Maximum time step and integration order are the parameters driving the uncertainty while TC 4 has the largest response to variation of numerical parameters. At the nominal maximum time step value of 100, the entire spread of temperature for second order is the error (Figure 9(b)), approximately 1.5 K. By reducing the maximum time step to 50, the cluster of points near 333 K (the boxed data on Figures 8(b) and 9(b)) can be ignored, since it is clear on Figure 8(b) that no such temperatures exists when the maximum time step is 50 or less. This reduces the error to 0.75 K.

6 Summary

A verification analysis was conducted on a finite element heat transfer model of the RIP. A conservative estimate of the error introduced into the solution by the numerical parameters and mesh is ± 2 K. This study, once the infrastructure was prepared, took less than two working weeks to conduct. This time scale was acceptable to the development engineers that requested the thermal model. This paper shows that rigours verification of model can be obtained within the constraints of an aggressive design, development, and production schedule.

6.1 Future Work

In the numerical parameters study, TC 4 exhibited responses different from the other five thermocouples. TC 4 is located on EP2, the electronics package that turns on for the last minute of the simulation. This may explain the behaviour of TC 4 and therefore parameters involving contact resistance transient solutions should be studied more in depth to identify an optimal setting. Once the above topics have been addressed, the analysis team will conduct a material sensitivity study and validate the model, using data collected by the experimental team.

7 Acknowledgments

References

[1] Sierra Core Team. Sierra Thermal/Fluids Code. Sandia National Laboratories: Albuquerque, NM, 2012.

[2] Pro Engineer Core Team. *Pro Engineer WildFire 5*. PCT: Needham, MA 2009.

[3] Cubit Core Team. Cubit: Geometry and Mesh Generation Toolkit. Sandia National Laboratories: Albuquerque, NM, 2011.

[4] Hughes, T.J.R. *The Finite Element Method*, Dover Publications: Mineola, New York, pp, 52-56, 2000.

[5] Adams, B.M., Bohnhoff, W.J., Dalbey, K.R., Eddy, J.P., Eldred, M.S., Gay, D.M., Haskell, K., Hough, P.D., & Swiler, L.P., DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.0 User's Manual. *Sandia Technical Report SAND2010-2183*. Sandia National Laboratories, Albuquerque, NM, 2011.

[6] J. C. Helton & F. J. Davis. Sampling-based methods for uncertainty and sensitivity analysis. *Technical Report SAND99-2240*. Sandia National Laboratories, Albuquerque, NM, 2000.

[7] R. L. Iman & M. J Shortencarier. A Fortran 77 program and user's guide for the generation of latin hypercube samples for use with computer models. *Technical Report NUREG/CR-3624, SAND83-2365*. Sandia National Laboratories, Albuquerque, NM, 1984.

[8] R Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing: Vienna, Austria. 2012.

[9] A. Saltelli, K. Chan, E. M. Scott, eds, *Sensitivity Analysis*, Wiley Series in Probability and Statistics, Wiley: Chichester, West Sussex, England, 2000.

[10] A. Saltelli, S. Tarantola, F. Campolongo, & M. Ratto. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models.* John Wiley & Sons: Chichester, West Sussex, England, 2004.