

## DEVELOPMENT AND USE OF AN ANSA COMPONENT FOR ISIGHT PROCESS INTEGRATION

<sup>1</sup>Dr. David Kokan\*, <sup>2</sup>Amol Dongare

<sup>1</sup>Dassault Systèmes SIMULIA, United States, <sup>2</sup>3DPLM Software Solutions, India

KEYWORDS – Process Automation, Multidiscipline, Optimization, Components, Morphing

ABSTRACT – In today's competitive marketplace, successful companies are designing and delivering high quality, robust products while simultaneously reducing their design cycle times. Simulation and process automation software are key enablers that allow these conflicting goals to be managed. Simulation predicts performance while process automation captures the simulation process involving multiple design tools, greatly increasing the number of design alternatives that can be studied. ANSA, a product of BETA CAE Systems, is an advanced multidisciplinary CAE pre-processing tool which is used to prepare a design alternative for simulation. Isight, a product of SIMULIA, enables users to automate simulation process flows and leverage advanced techniques such as Design of Experiments, Optimization, Approximations, and Design for Six Sigma to thoroughly explore the design space. Isight comes equipped with a standard library of components which form the building blocks of Isight process flows. A component is a container with its own interface for integrating and running a particular simulation application directly within Isight. For years, Isight has interacted with ANSA through generic components. We determined that a more specific integration between ANSA and Isight was required in order to make it easier to develop Isight process flows that contain ANSA as one of the steps. This paper describes the development tools used to create the Isight ANSA component. A simulation use case, the design of a piston rod, is described to demonstrate the ANSA component. Common real-world issues are highlighted that the user of such a component must consider in order to be successful with process automation and optimization.

### 1. INTRODUCTION

In today's competitive marketplace, successful companies are designing and delivering high quality, robust products while simultaneously reducing their design cycle times. Simulation and process automation software are key enablers that allow these conflicting goals to be managed. Simulation predicts performance while process automation captures the simulation process involving multiple design tools, greatly increasing the number of design alternatives that can be studied. This allows companies using FEA to move from a reactionary mode – if something breaks, analyze to see why – to a more proactive and upfront consideration of numerous alternatives in a fraction of the time [1].

ANSA, a software product of BETA CAE, is an advanced multidisciplinary CAE pre-processing tool which is used to prepare a given design for simulation. ANSA takes geometric data, meshes it, and applies loads and boundary conditions to produce run-ready solver input decks for a variety of popular analysis codes including NASTRAN, LS-DYNA, PAM-CRASH, RADIOSS, Abaqus/Standard, Abaqus/Explicit, and ANSYS Solver. Furthermore, ANSA provides a powerful morphing capability that allows variations of geometry or finite element models to be made. This feature is ideal for use in combination with process automation software such as Isight.

Isight, a software product of Dassault Systèmes SIMULIA, provides engineers with a suite of visual and flexible tools for creating simulation process flows. These sim-flows execute a variety of applications, including commercial CAD/CAE software, internally developed programs, and Excel spreadsheets in order to automate the exploration of design alternatives and identification of optimal performance parameters. Isight enables users to automate sim-flows and leverage advanced techniques such as Design of Experiments,

---

Optimization, Approximations, and Design for Six Sigma to thoroughly explore the design space.

Isight comes equipped with an ever increasing library of components (Figure 1) which form the building blocks of Isight sim-flows. A component is a container with its own interface for integrating and running a particular simulation application directly within Isight. General purpose components such as Simcode and Data Exchanger allow Isight to interact with any generic ASCII file driven batch process. These components require the end user to have an understanding of the input and output file format of the application. Other more specific components, such as Abaqus or Nastran components, are built to support a particular CAE application. These components understand the specific solver file formats and are able to scan them for potential Isight parameters and update or extract values as necessary without the end user providing any logic/instructions.

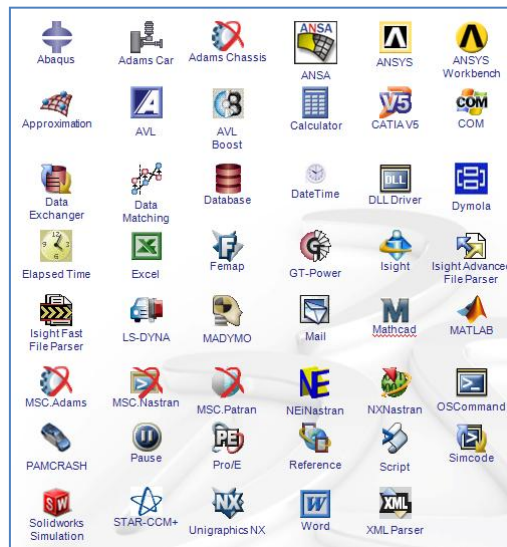


Figure 1 – Isight Application Component Library [2]

The Isight component architecture also supports the integration of customer-proprietary applications. Therefore users of Isight can create and publish their own custom components to meet their process integration needs. Isight provides development tools and templates to assist in the component development process.

SIMULIA has developed an ANSA component that provides a seamless integration between ANSA and Isight. The component exposes all design variables of the ANSA model's Optimization Task for use within Isight. During set up of the sim-flow, the component retrieves each variable name, current value, type, and range. During execution, Isight sends a new set of design variable values to ANSA for processing and generation of a finite element solver input file. The updated design variable vector can be specified manually by the Isight user or generated automatically by any available Isight design driver. All finite element formats supported by ANSA, including Abaqus, ANSYS, NASTRAN, etc. can be generated by the component. The solver input deck is also stored within the Isight database and can be mapped to downstream simulation components for subsequent execution and results post-processing. For example, Figure 2 shows a sim-flow consisting of an ANSA component using mesh morphing to drive changes to an Abaqus input deck, an Abaqus component executing the simulation and extracting results from the .odb file, and an Excel component post-processing the Abaqus simulation results.

In this paper, we describe the steps involved in building the ANSA component using Isight's open SDK and the Isight Developer Eclipse<sup>TM</sup> environment, as well as application of the component within a realistic automated Isight sim-flow. Furthermore, we discuss various real-world issues that must be taken into consideration when performing process automation involving finite element models.

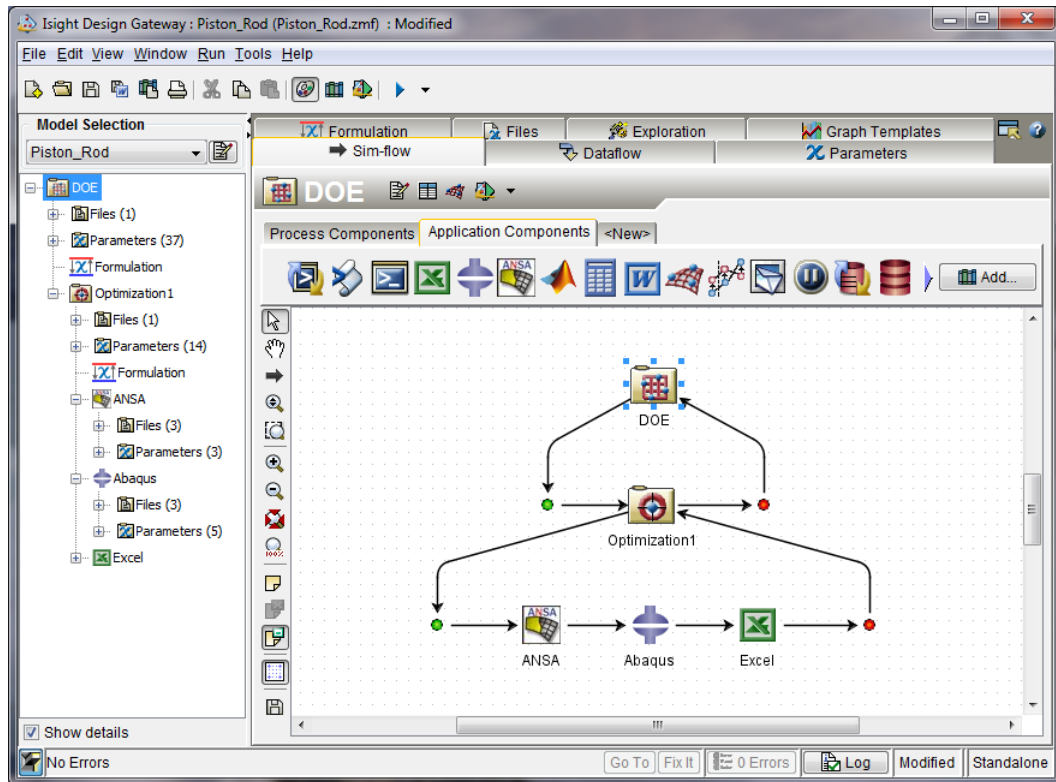


Figure 2 – Isight sim-flow consisting of ANSA, Abaqus, and Excel components driven by a DOE operating on a sub-optimization

## 2. DEVELOPMENT OF THE ANSA COMPONENT

Isight is an open environment that allows users to extend and customize the available functionality in Isight by developing their own specialized “extensions” using the Isight Software Development Kit (SDK). There are a number of possible ways to extend Isight, including [3]:

- **Applications.** Custom programs/tools for building, editing, and executing models and retrieving results in a programmatic fashion.
- **Components.** Independent tools/applications or design methods for which adaptors have been written to provide a specific service that can be used in an Isight sim-flow.
- **Plug-ins.** General utilities for extending the capabilities of components (e.g., optimization/DOE techniques, random variable distributions, approximation techniques, etc.).
- **Units.** Attributes that can be specified for parameters to identify the standard of measurement for the values of those parameters (e.g., meters, seconds, degrees Celsius, etc.).
- **Visuals.** Custom graphs/tables that provide a special way to visualize results.
- **Clients.** Custom interfaces for building and executing Isight models and reviewing results.

In the case of Isight interacting with ANSA, a component was selected as the extension mechanism. Its purpose is to connect to an ANSA model, retrieve a list of potential parameters, modify those parameters based on input from Isight design drivers, and execute ANSA to produce an updated solver input deck.

To create a component for Isight, the developer must produce a Java JAR file that includes the following:

- An **Editor** Java class to provide a GUI for configuring the component.
- An **Executor** Java class to provide the runtime execution functionality.
- A **Handler** Java class (optional) to provide logic for validating the configuration of the component and reacting to changes in the model.
- A **PostProcessor** Java class (optional) to provide functionality for carrying out the actions after the execution of the component.
- An **API** (Application Programming Interface) Java class (optional) to allow programmatic configuration of the component in an intuitive and stable manner.
- **Resource files** such as an icon image file or an internationalization catalog file.
- An **XML descriptor** file that specifies the above classes, resource files, and other quantities (properties/parameters) that specify input information required by the component.
- **Any other files necessary** to support these adaptor classes or ANSA.

In the case of the ANSA component, an Editor, Executor, Handler, and API were provided. No specific PostProcessor class was provided.

### Isight Developer & Component Templates

Isight provides support for creation of components through an integrated development environment (IDE) called “Isight Developer” and through component templates.

Isight Developer is a plug-in to the popular open-source Eclipse integrated development environment [3]. The Isight Developer Eclipse extension simplifies the process of building Isight components by providing a custom wizard for creating development projects and a custom GUI perspective which includes Isight component specific information. Isight can be launched in a debug mode from within Isight Developer in order to step through component source code. Immediate syntax checking against the Isight API is also provided.

Isight Developer also supports the concept of component templates. Templates provide much of the common component source code, allowing the developer to focus on the part of the source code specific to the application being driven. The template that was selected for the ANSA component is called the “Command with File Input/Output (I/O)” template. This template supports a common component type that updates an input file, executes a system command, and extracts data from an output file. The template provides fully implemented functionality to properly handle Isight parameters, properties, and user-supplied options. In addition, the template provides a general working implementation of the classes for the component Handler, Executor, API, and preference panel. Figure 3 shows the initial user interface that is automatically created when this template is selected.

### ANSA Optimization Task

ANSA’s Task Manager provides a special functionality called the Optimization Task by which a user can set up an optimization problem within ANSA. The Isight ANSA component uses this Optimization Task mechanism to interact with the ANSA model. A specified sequence of ANSA functions can be defined within this task and design variables can be created to drive these functions. Design variables can be morphing parameters, ANSA parameters, session commands, user scripts, or any combination thereof [4,5]. This powerful and flexible mechanism allows many different types of potential variables to be created, from global parametric changes of geometry to mesh controls to individual values of ANSA cards like shell thickness or material density. Furthermore the user can define allowable ranges for design variables (minimum and maximum values), allowable steps, or a discrete list of allowable values, thereby establishing the valid design space of the model.

---

Once an Optimization Task has been defined, ANSA produces a special design variable file, or “DV file”, which contains all the information needed by Isight to execute the task. Specifically, this file includes the name of the reference .ansa file as well as all design variable names, initial values, and allowable values. The Isight user browses to the DV file from the ANSA component in order to configure the component in Isight (Figure 4). No knowledge of the format of the DV file is required on the part of the user as has been the case in the previous studies involving ANSA and Isight [6,7]. Because a component template was used as the starting point for the development, the only programming required was new code to parse the DV file and define ANSA specific execution options (Figure 5). This allowed the initial development of the component to be accomplished in just a few days. Because all tools used to develop the ANSA component are released with Isight as part of the public SDK, anyone (not just SIMULIA R&D) can easily build their own components and make them available within Isight.

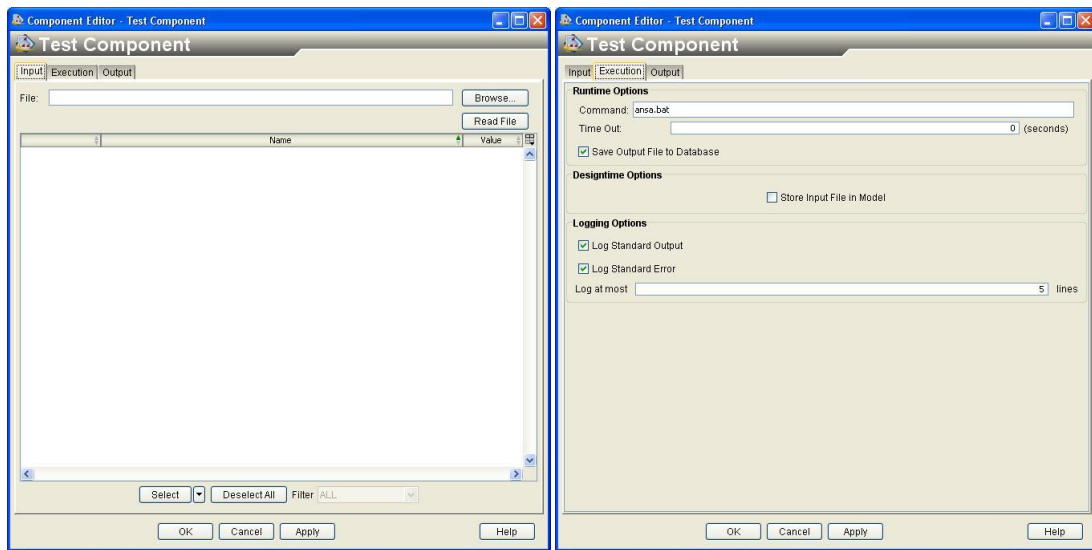


Figure 3 – Standard template component editor

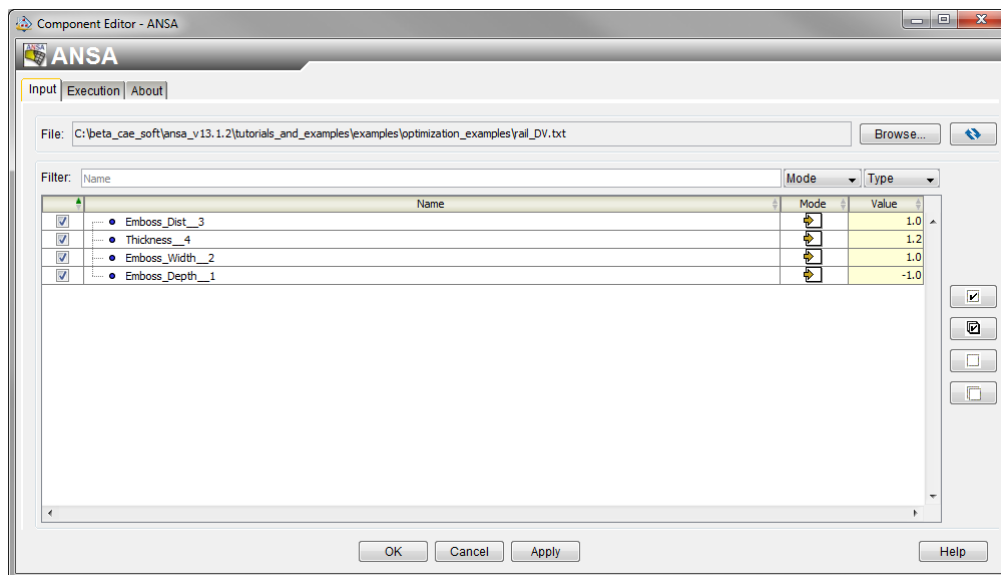


Figure 4 – Custom ANSA component editor – Input tab

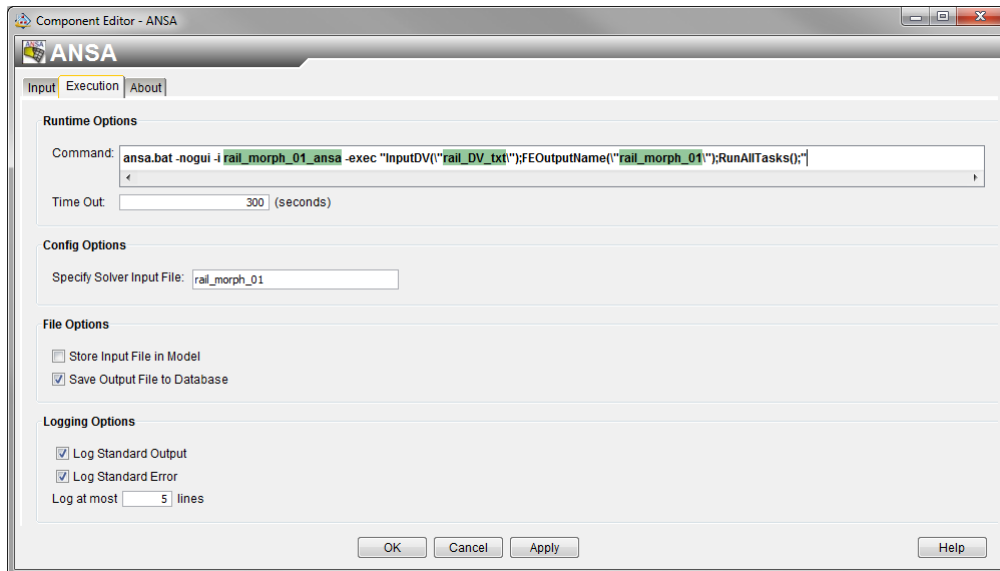


Figure 5 – Custom ANSA component editor – Execution tab

### 3. USE CASE SCENARIO – PROCESS AUTOMATION OF A PISTON ROD

Consider the ANSA model displayed in Figure 6. It consists of a solid piston rod constrained at one end and with a simple static load applied at the other end [6]. Morphing boxes are defined to allow the surface mesh to be varied. Several morphing design variables have been created which control variation of the piston rod, including taper, curvature, and pocket depth (see Figure 7). After each morphing, a volume mesh is regenerated from scratch, thus avoiding high aspect ratio elements sometimes associated with morphing. ANSA outputs a mass report as well as a run-ready Abaqus input deck, which includes all node and element definition, loads, and boundary conditions for the design configuration of interest.

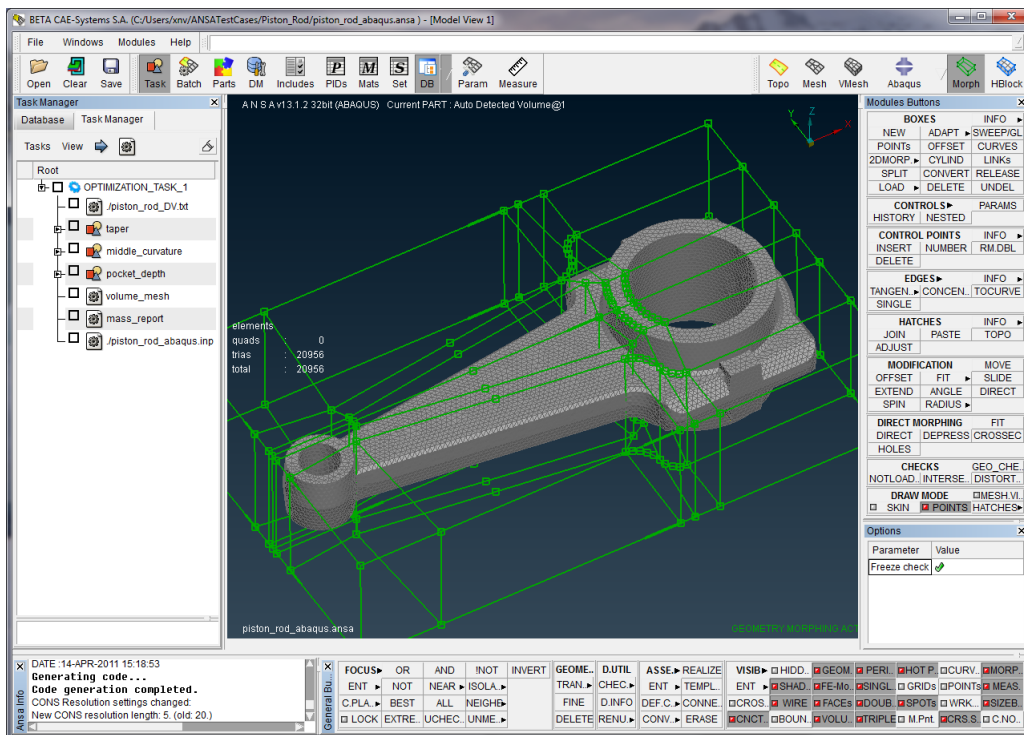


Figure 6 – Piston rod ANSA test model

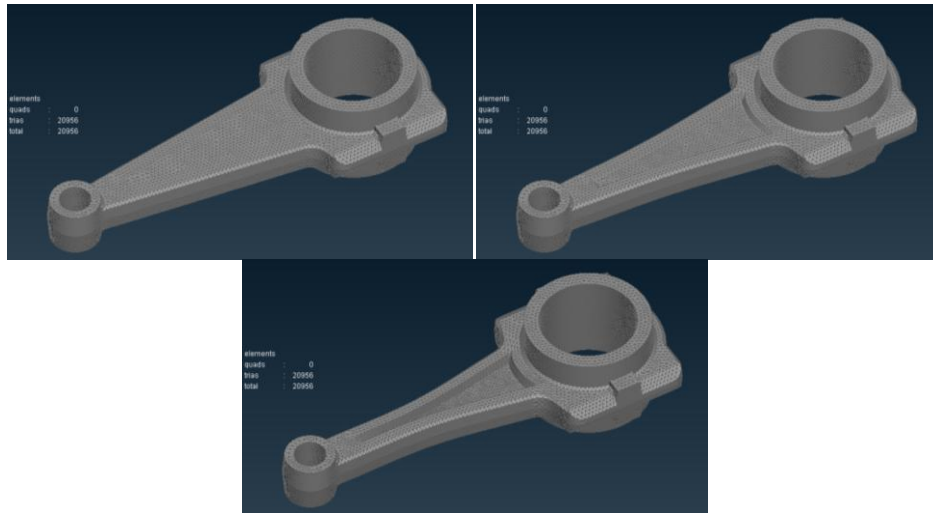


Figure 7 – ANSA Parametric variations of the piston rod

To configure the piston rod for execution from within Isight, ANSA and Abaqus components are dragged and dropped into the Isight “Sim-flow” tab (Figure 8) [8,9]. Each component is configured by browsing to the following reference files:

- ANSA component – DV file
- Abaqus component – Abaqus input deck (.inp), Abaqus output database (.odb)

Once the user selects each of these files, a list of potential Isight parameters is presented and the parameters of interest for the Isight study can be selected (Figures 9 and 10).

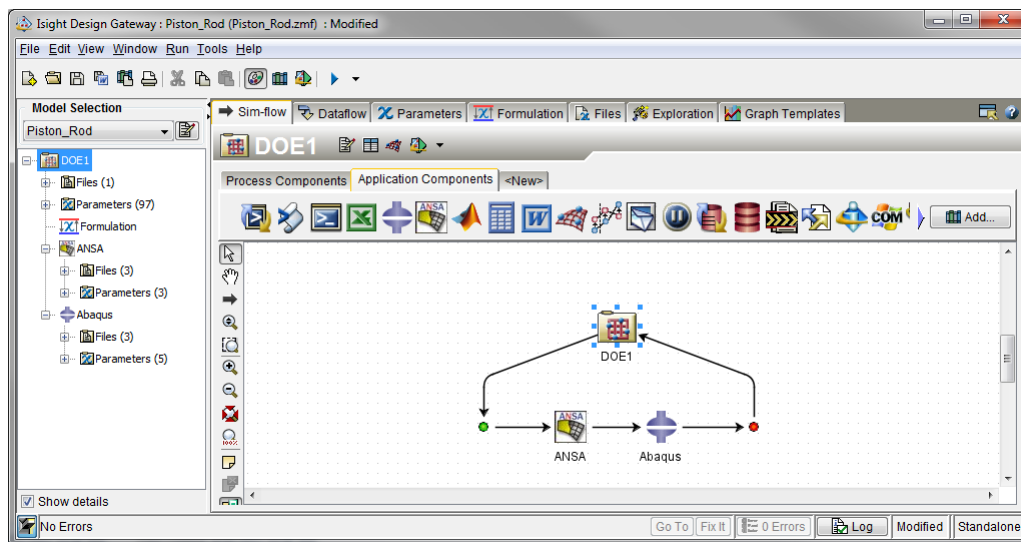


Figure 8 – Isight Sim-flow – ANSA component feeding Abaqus component driven by DOE

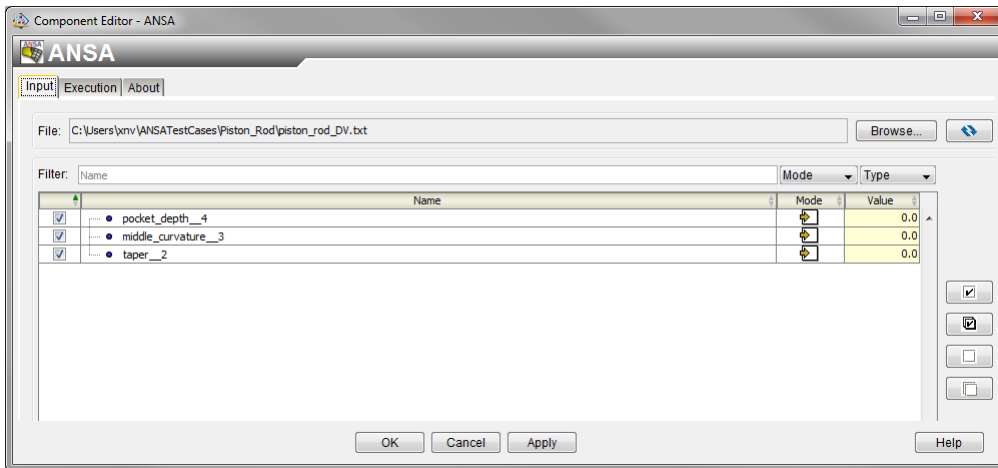


Figure 9 – ANSA component input tab configured for piston rod model

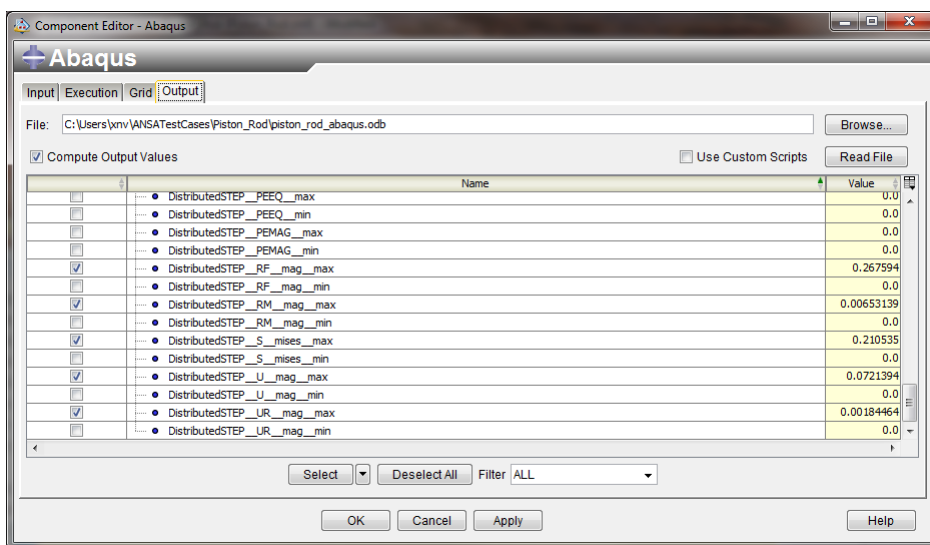


Figure 10 – Abaqus component output tab configured for piston rod model

Once the components are fully configured, the Isight parameters tab shows all Isight parameters associated with each component. The ANSA component configures not only the name and value of each Isight parameter, but also all allowed value information prescribed in the ANSA model (Figure 11).

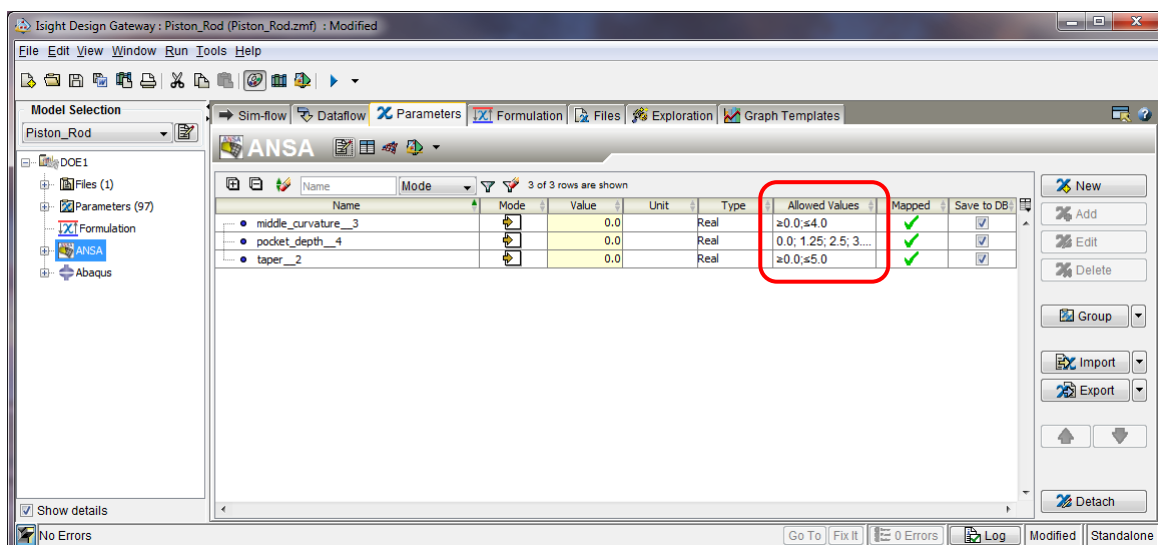


Figure 11 – Isight parameters tab configured for piston rod model



The Isight Dataflow tab allows the user to map parameters and files from one component to the next. In the piston rod scenario, the user must map the Abaqus input deck produced by the ANSA component downstream to become the input of the Abaqus component (Figure 12). No other parameter mappings are required. The solver deck serves as the communication link between the two components.

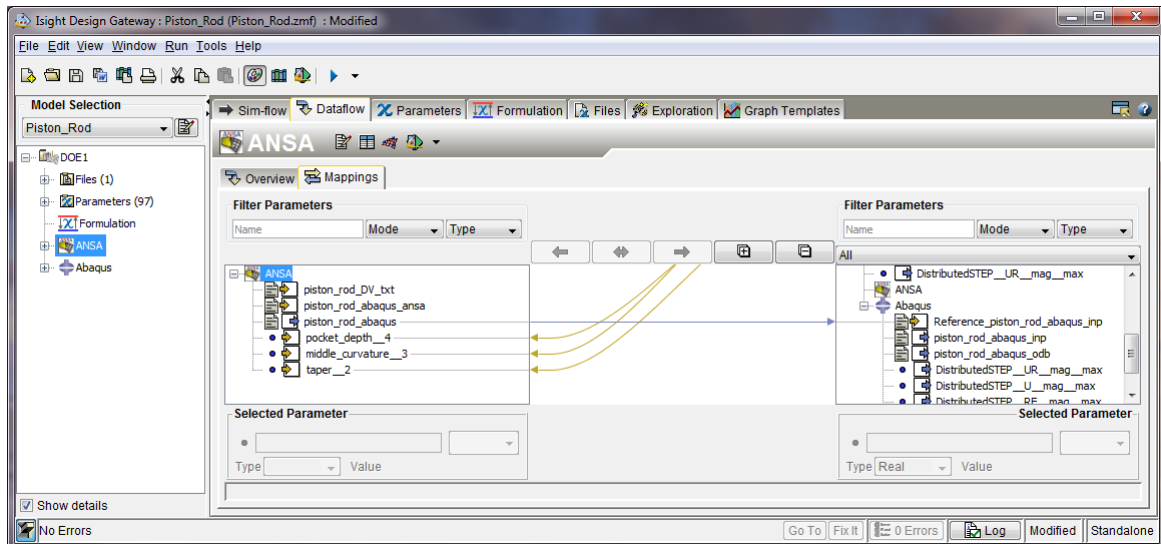


Figure 12 – Isight Dataflow mapping of Abaqus input deck from ANSA component to Abaqus component

Once the ANSA and Abaqus components have been configured, the sequence can be run once or can be driven by any Isight design driver component. Isight provides numerous algorithms to drive a subflow, including simple loop (while, do-until, for, foreach), Design of Experiments (DOE), Monte Carlo simulation, Optimization, Design for Six Sigma, Taguchi Robust Design, and others. Each of these methods can also be dragged and dropped into the sim-flow and configured through its own component editor. For example, Figure 13 shows the DOE component editor configured to execute the ANSA plus Abaqus subflow 100 times by utilizing a Latin Hypercube technique. Figure 14 shows the results of the DOE with data sorted by maximum stress. Note that the user can easily launch tools directly from within Isight to visualize results from any run. For example, Figure 14 shows Abaqus/Viewer being launched to view the stress field associated with one of the designs.

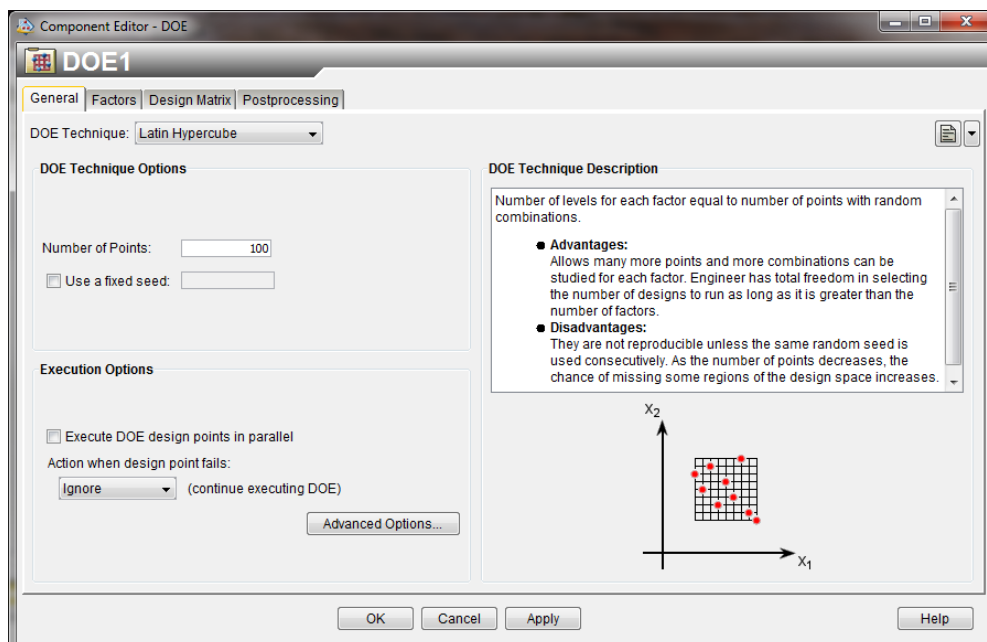


Figure 13 – DOE component editor

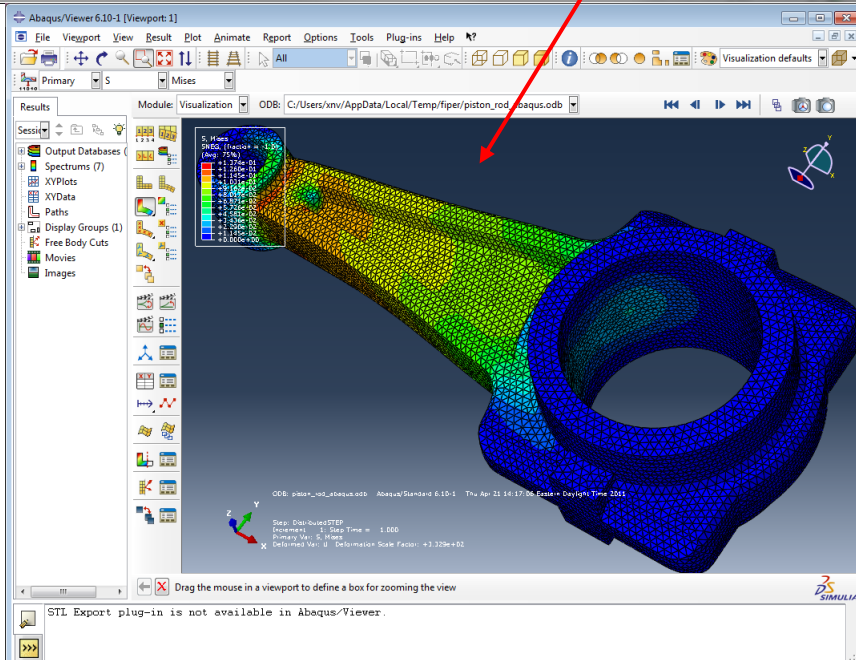
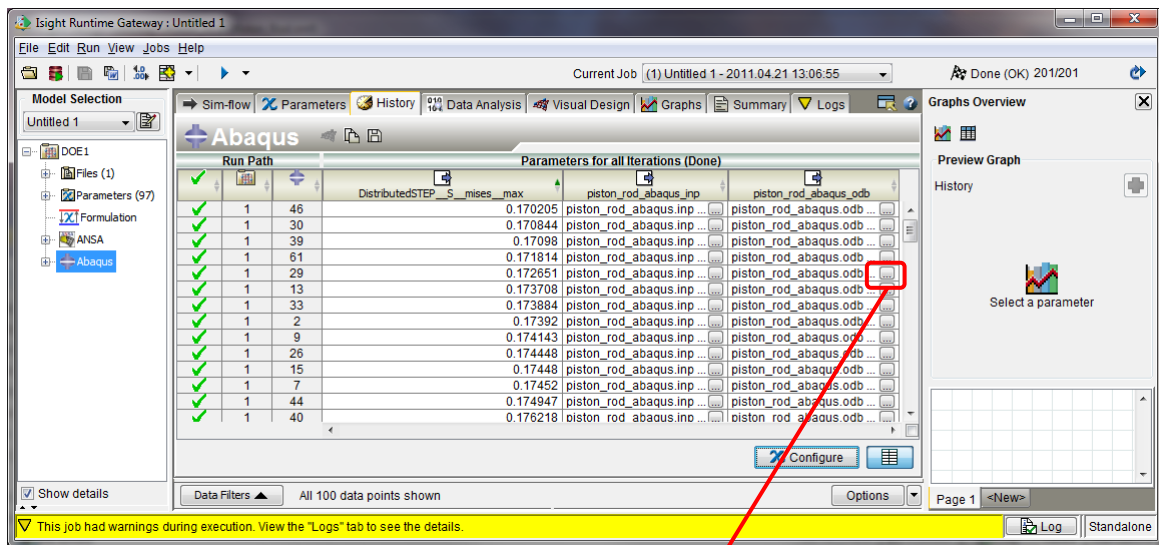


Figure 14 – Isight Runtime Gateway used to launch Abaqus/Viewer

Finally a variety of post-processing options are available to view the results of the DOE study in various forms. Figure 15 shows two such options. The first is a Correlation Map which provides an overview of the parameters and their relations to one another. The second is a visualization of a three dimensional cut through the design space created by applying an approximation through the DOE data.

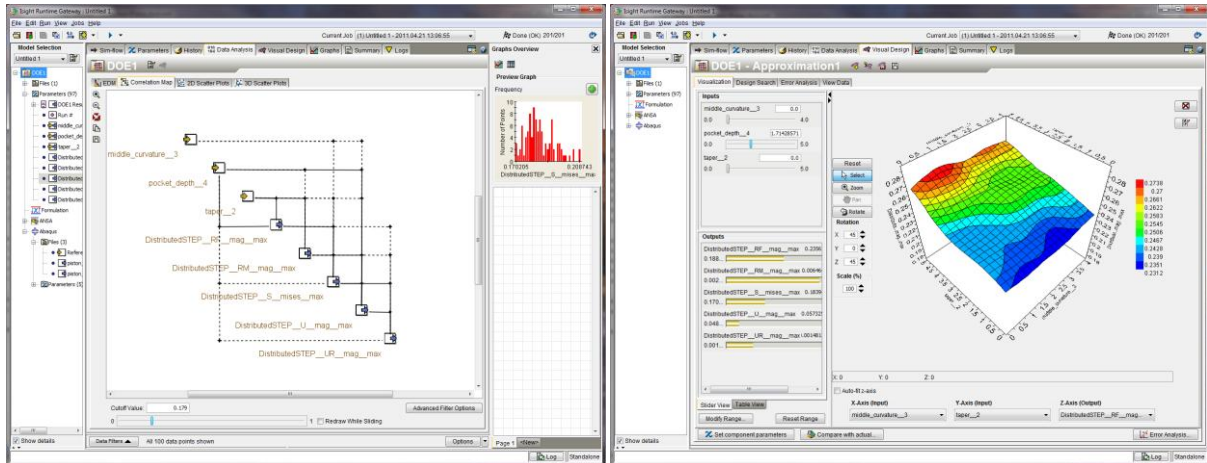


Figure 15 – Correlation Map and 3D visualization of a cut through the piston rod design space

#### 4. REAL WORLD CONSIDERATIONS FOR PROCESS AUTOMATION AND OPTIMIZATION WITH THE ANSA COMPONENT

Thanks to a robust and growing Isight library of application components including ANSA and Abaqus, it is becoming easier to quickly build Isight sim-flows and leverage a powerful collection of multi-run algorithms. However, a number of common issues and pitfalls exist and the user of Isight must be aware of these and how to cope with them.

##### Long Simulation Run-Times

Even as compute speeds continue to increase, so does the size of simulation models and the complexity of physics being modelled. It can take hours or even days to perform a single non-linear structural or CFD calculation. Running such a model as part of an optimization or DOE study can be time-prohibitive. However, techniques exist to reduce the time.

One option is to parallelize the multi-run simulation at the level of the design driver. Instead of submitting just one analysis at a time, Isight offers a variety of algorithms that can submit batches of analysis jobs to a compute cluster. In some cases, such as the Latin Hypercube DOE described in section 3, the parallelization can be achieved up to the total number of runs itself. Therefore, with enough hardware, the DOE can be accomplished 100 times faster than running sequentially. Organizations can therefore trade-off hardware cost against the benefit of accomplishing design studies more quickly.

Isight approximations are also a technology that can help when dealing with long run times. Used to visualize the design space in Figure 15, an approximation can also be used as a replacement to the full calculation when running under an optimization loop. Approximations are also very valuable when performing robust design analysis. Estimating reliability associated with uncertainty of input parameters through Monte Carlo on the approximation is a very efficient solution method. Isight offers several approximation algorithms including response surfaces, radial/elliptical basis functions, orthogonal polynomials, and Kriging. This collection offers a good range of capabilities for approximating various design space topographical features.

##### License Consumption Cost

License consumption cost becomes a major concern when running commercial codes in parallel via Isight. Typically a company has some limited pool of licenses or analysis tokens that are available to execute a commercial code. It is easy for an Isight user to consume all the licenses in his organization on a large parallel set of automated runs, severely impacting

the work of colleagues and, at the same time, incurring a high licensing cost. Isight provides grid plug-ins for popular batch queues such as LSF and PBS/Pro. These queues can be configured to submit the analyses in an orderly way and according to priorities set by the organization. However, the queuing process can again increase the overall time needed to execute a multi-run scenario.

SIMULIA has recognized the issue of high license consumption cost. Starting with Abaqus 6.10 and Isight 4.5, SIMULIA introduced a new token usage policy to encourage use of Abaqus and Isight together for design exploration studies. The new system reduces analysis token usage per run when multiple Abaqus runs are executed within an Isight study. This approach makes it more cost effective to run variations on Abaqus models using Isight than if design alternatives are explored manually. Token reduction applies for as few as two Abaqus runs executed within an Isight study and reaches a threshold 60% target reduction when 10 to 32 runs are executed, depending on the number of runs executed concurrently.

### Approximation Quality

Approximations are valuable for design space visualization and for reducing the time needed to run various Isight design methods. However, depending on the complexity of the physics being modelled as well as the size of the design space being approximated, the accuracy of the approximation may be poor for one or more output parameters. The user must assess the quality of the approximation before making use of it. Isight provides functionality to do just that. By default, Isight performs cross-validation of the approximation, either by generating an additional set of data points or by removing a subset of the points from the main data set, and then comparing exact and approximate values at each point. Various measures of error are computed and displayed as shown in Figure 16. The plot shows actual versus predicted value for each cross-validation point. The closer the points are to the 45-degree line, the better the approximation. The user can improve error levels by reducing the range of the input variable, by including additional sample points, or by utilizing different approximation algorithms until he is satisfied with the accuracy of the approximation.

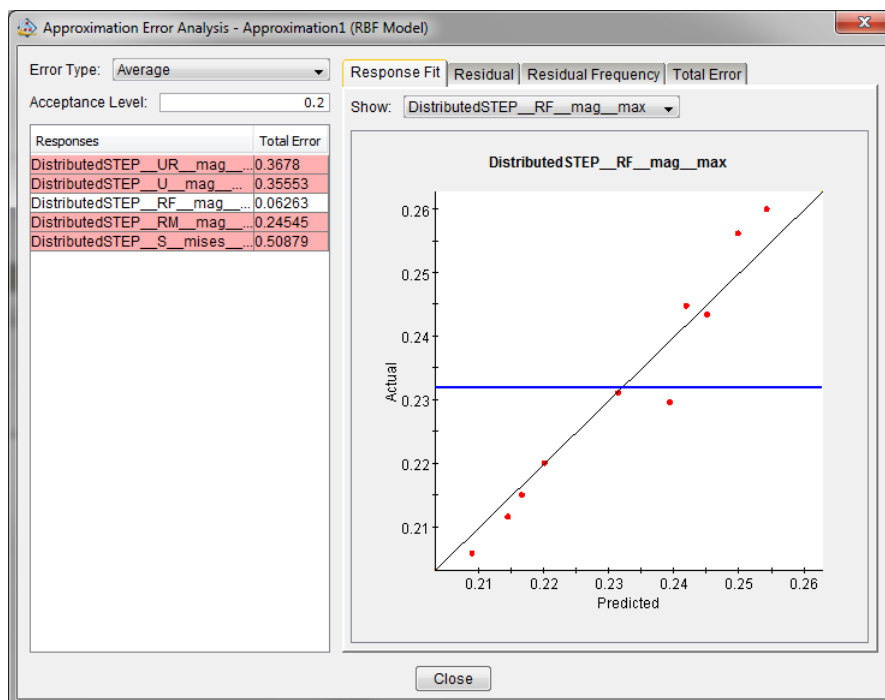


Figure 16 – Isight Error Analysis interface

## Simulation Model Robustness and Accuracy

The quality of the simulation model itself plays a significant role in the effectiveness of any Isight multi-run job. First, the analysis should execute without crashing across the range of the design space. Isight optimization algorithms can cope with code failure rates as high as 40%, but at the expense of efficiency in driving to an optimal solution. Development of ANSA models that morph successfully for all design points of interest is a key to ensuring good optimizer performance. One approach to testing your model robustness is to run a DOE or a Monte Carlo simulation across the entire design space of interest to see if any failures occur and improve the ANSA model to eliminate as many failure modes as possible.

A common problem in many finite element models is lack of accuracy, both in terms of the output parameter values as well as their gradients with respect to each input parameter. Coarse meshing or improper finite element selection can easily lead to inaccurate models. For example, Figure 17 shows a simple, transversely loaded cantilever plate. The plot displays tip displacement versus cantilever span. We would expect the maximum displacement to increase monotonically with increasing span. Although that trend is observed globally, note the noisiness in the displacement calculation with local gradients alternating between positive and negative values. This behaviour was produced by poor selection of tetrahedral elements that were not appropriate for this bending load case. An FEA modelling expert can remove the noise through selection of different elements as well as a slightly finer hex mesh. Attempting to utilize a gradient based optimizer together with this inaccurate model would certainly lead to poor performance since, between any two points, the gradients are highly inaccurate and alternate signs. It is important that the finite element modeller checks the quality of his model. Isight provides functionality to assist with this. A Noise Analysis component is available which runs a series of one-dimensional parameter scans and computes noise level of each selected output caused by each input scan.

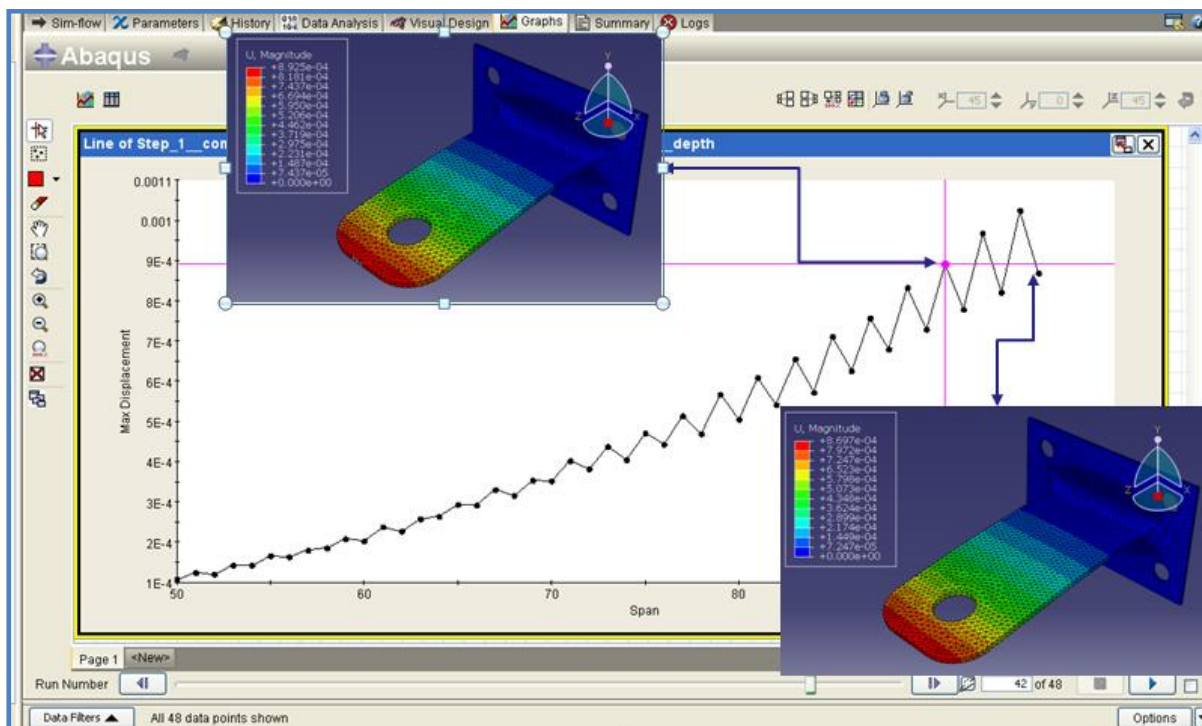


Figure 17 – Maximum displacement versus span for a noisy, inaccurate cantilever plate model

## 5. CONCLUSIONS

In this paper we presented an ANSA component which enables easy process integration within Isight. The component was created using the Isight Developer environment and a component template as the starting point. These tools, which are available to Isight customers and SIMULIA partners, along with Isight's open SDK, greatly reduce the development burden when creating new custom components for Isight. Developers can focus on the third-party code specific parts of the component while leveraging common code for creating the component user interface and executor. A piston rod use case was used to show how straight forward it is to build an Isight sim-flow involving the ANSA component and a downstream solver component. The piston rod was modified using ANSA's powerful morphing capability to produce run-ready Abaqus input decks. An Isight multi-run algorithm using Design of Experiments was executed on the ANSA and Abaqus subflow and Isight post-processing capabilities were highlighted. Finally, common issues with multi-run FEA jobs and Isight capabilities to deal with these issues were discussed. By leveraging Isight and its ANSA component, companies can derive greater benefits from their simulation investment. More alternatives can be considered and better designs produced in less time.

## 6. ACKNOWLEDGEMENTS

The authors gratefully acknowledge George Korbetis from BETA CAE for his valuable insights and technical support during design, development, and testing of the ANSA Isight component. George helped us to create a robust and accurate piston rod model. We also thank Martin Roswall from Dassault Systèmes SIMULIA for assistance with ANSA model building.

## REFERENCES

- (1) SIMULIA Realistic Simulation News, *Baker Hughes Refines Expandable Tubular Technology with Abaqus and Isight*, January/February 2011
  - (2) SIMULIA Component Central Website, <http://components.simulia.com>, Dassault Systèmes SIMULIA, 2011
  - (3) Isight 5.5 Development Guide, Dassault Systèmes SIMULIA, April 2011
  - (4) ANSA version 13.1.2 User's Guide, BETA CAE Systems S.A., February 2011
  - (5) ANSA version 13.x.x Optimization with ANSA, BETA CAE Systems S.A., February 2011
  - (6) ANSA version 13.1.2 Tutorial, OPTIMIZATION Coupling ANSA/μETA to Isight, BETA CAE Systems S.A., February 2011
  - (7) F. Krabchi, *Multidisciplinary Design Optimization (MDO) Using ANSA/μETA Postprocessor and Isight*, 3<sup>rd</sup> ANSA & μETA International Conference, September 2009
  - (8) Isight 5.5. User's Guide, Dassault Systèmes SIMULIA, April 2011
  - (9) Isight 5.5 Component Guide, Dassault Systèmes SIMULIA, April 2011
-