Designing a Composable Geometric Toolkit for Versatility in Applications to Simulation Development

Gregory S. Reed
Center for Modeling, Simulation, and Analysis
University of Alabama in Huntsville
Huntsville, AL 35899
(256) 824-4625
reedgs@uah.edu

Thomas Campbell
Independent Contractor/Consultant
5000 Sunset Bluff
Huntsville, AL 35805
(256) 882-2420
twcjr@comcast.net

Keywords: geometric modeling, composability, MATLAB, NASA, Project Constellation

Abstract

Conceived and implemented through the development of probabilistic risk assessment simulations for Project Constellation, the Geometric Toolkit allows users to create, analyze, and visualize relationships between geometric shapes in three-space using the MATLAB computing environment. The key output of the toolkit is an analysis of how emanations from one "source" geometry (e.g., a leak in a pipe) will affect another "target" geometry (e.g., another heat-sensitive component). It can import computer-aided design (CAD) depictions of a system to be analyzed, allowing the user to reliably and easily represent components within the design and determine the relationships between them, ultimately supporting more technical or physics-based simulations that use the toolkit. We opted to develop a variety of modular, interconnecting software tools to extend the scope of the toolkit, providing the capability to support a range of applications. This concept of simulation composability allows specially-developed tools to be reused by assembling them in various combinations. As a result, the concepts described here and implemented in this toolkit have a wide range of applications outside the domain of risk assessment. To that end, the Geometric Toolkit has been evaluated for use in other unrelated applications due to the advantages provided by its underlying design.

1. PROBABILISTIC DESIGN ANALYSIS

In support of Ares I, a multidisciplinary team of individuals from Marshall Space Flight Center (MSFC), the University of Alabama in Huntsville (UAH), Jacobs Engineering Group, and other organizations was formed to develop risk assessment and design analysis simulations of the spacecraft.

In the risk assessment process, the spacecraft's Fault Tree, Failure Modes and Effects Analysis (FMEA), and other system documentation are analyzed in order to formulate scenarios of potentially serious risks to mission success or crew safety. Such scenarios can include pipe leaks that may damage other nearby components, birds striking the spacecraft during ascent, and other concerns. The risk assessment team then creates models for each of these scenarios, runs them, and analyzes the results of the simulations. This analysis helps to determine the level of risk involved in a given scenario, usually as a function of certain parameters such as time.

The risk assessment team provides input to other organizations within NASA. The design engineering teams typically use the team's analyses to determine which components or subsystems need revision to reduce risk; alternatively, an analysis might also point to components that have been significantly over-designed with respect to their risks, allowing design engineers to relax their standards on those components and thus provide a weight and cost savings to the spacecraft design. Aborts systems groups use the information and its associated timing data to determine when failures need to be detected in order to mitigate risks as well as where to place appropriate sensors and controls. Reliability and safety teams use it to provide detail in the Fault Tree and FMEA so that all parties involved have a more refined view of the risk situation of the spacecraft.

1.1. Approach and Modeling Process: Designing toward Modularity

Rather than develop each model in isolation, the team opted to plan toward the concept of an overall dynamic ascent model for the Ares

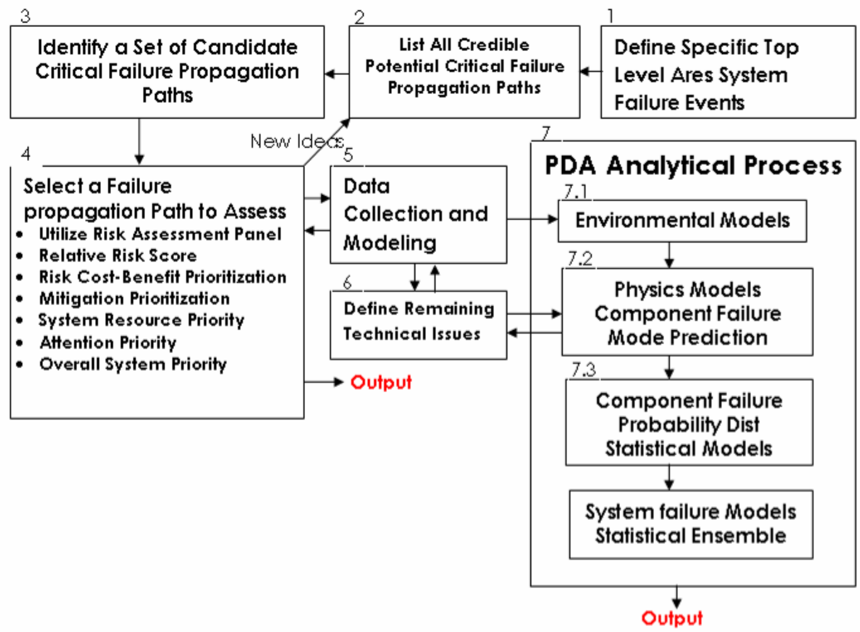


Figure 1. Top-level overview of the team's analytical approach [1].

spacecraft by introducing design principles that fostered reusability, commonality, and meaningful communication between models. This would make the models much more useful in the long run, as potential failures could be propagated throughout the entire system in order to determine their further-reaching effects and provide a much more accurate portrayal of the system's overall risk and reliability.

Consider an example of the application of this idea: a model that determines which engine components are affected by a rupture in a pipe. If the results of the model indicate that an oxygen line is damaged by the flow of gases emanating from the pipe's rupture point, then it could trigger another model that determines what occurs if liquid oxygen does not reach a pump. Since the models are stochastic and may produce different results each time they are run, a given failure may affect different components—and therefore may trigger a different chain of models—in subsequent runs.

Figure 1 shows the overall approach taken by the team, from analyzing potential risk scenarios to developing the models. Of particular note is the analytical process in block 7, which shows a hierarchy of models.

In this approach, the environmental models in block 7.1 collectively serve as the backbone of the overall risk simulation, controlling the flow of information from each of the individual physical risk simulations in block 7.2. As the overarching backbone, they also provide certain initial conditions, including the size, shape, and position of components involved, to the physical models. A given failure event could be triggered either manually or by random chance, and the appropriate physical model is run with certain initial conditions. Outputs from this model can specify whether any secondary failures have been caused by an initial failure.

Rather than use computationally expensive physics models for runs of the overall risk simulation, probability distributions of failures occurring as a function of certain parameters (including time, initial conditions, and other failures that have already occurred) can be created by running the stochastic physical models numerous times under all possible conditions. These new statistical models, depicted in block 7.3, would simply become lookup tables of events, times, parameters, and probabilities, which could be executed several orders of magnitude faster than the physics models while producing statistically equivalent results. The complete set of these statistical models, 7.4, could then be directly called by the environmental models to analyze failure propagation, providing a more complete picture of the overall risk involved in the spacecraft

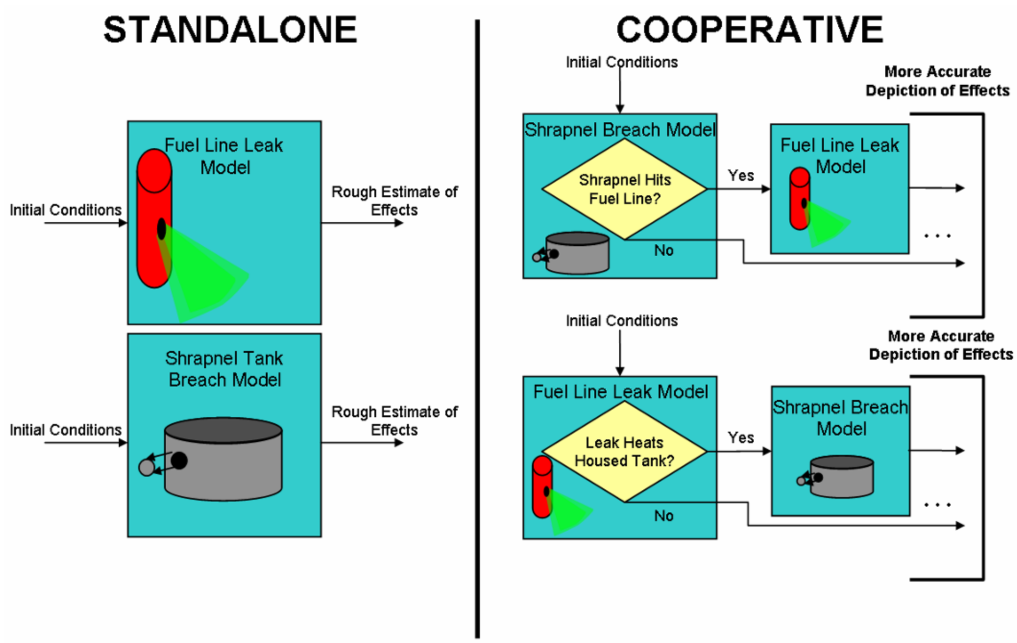


Figure 2. Example of how designing models that work together aids the design analysis procedure.

design.

This simple set of lookup tables could become a database of conditional probabilities related to failure mechanisms, modes, and effects within the system. Essentially, they become answers to the general question: "Given that a certain failure mode has occurred within a certain component, what is the probability of another certain failure mode being caused in another particular component?" Starting from one triggered failure, these probabilities can be strung together to determine answers to many questions regarding complete risk to certain key components and can even help quantify overall system risk. In this way, by simple probing of the failure space (triggering failures and determining how they propagate through the system), the complete picture of risk to the system can be studied.

If all risk scenario models were designed to take advantage of this common framework and to work together, then, as new models were created, more and more of the failure space can be explored. As can be shown with a few simple examples, the concept of cooperative risk models has a profound effect on the quantity of questions that can be answered as well as the quality of the answers themselves. This overall approach to risk assessment has been very thoroughly documented by the team [1].

1.2. Example Risk Scenarios and Failure Propagation

As an example of how risk simulation models might affect each other in a risk assessment simulation, consider the models in Figure 2. The left diagram presents two examples of risk models constructed in isolation. In this scenario, the fuel line leak model could answer questions directly related to the leak itself. This means answering the general question: "Given that there is a leak in the fuel line, what are the effects on the system as a whole?" Specifically, this model may determine whether a leak will cause the engine to shut down due to a lack of fuel. Similarly, consider a shrapnel tank breach model that deals with a portion of an engine component breaking off and penetrating the tank that encapsulates it. This model could also answer certain questions related to that specific scenario, such as whether a breach of shrapnel will relieve pressure in the tank.

Neither of these models alone can answer certain other questions related to risk, even within the limited scope of the components for which they are responsible. For example, if a leak in the fuel line also releases hot gas onto a nearby heat-sensitive component, then that component may fail and cause further unforeseen effects within the system. Moreover, shrapnel breaking through the tank can strike and damage another component, which may also cause other effects within the system. In fact, just as on the right diagram on Figure 2 suggests, shrapnel may strike the fuel line and cause it to leak, or a fuel leak may heat up the tank if it is nearby and cause components inside it to fragment and then to release additional shrapnel. Certain complexities, such as parameters passed from one model to the other, are not shown in the diagram (even if shrapnel strikes the fuel line, it may not cause a leak if the line is struck with a large angle of incidence, if the velocity of the shrapnel is small, or if the line is resilient to impact-induced stresses) but it signifies just how the models might interact.

This design principle offers the ability for the same models to answer many additional questions related to overall system risk. One component's failure could potentially trigger a certain failure mode (e.g. melting or distorting) via some failure mechanism (e.g. heat from a fuel line leak's spray of hot gas) as a property of another component. The secondary effect on the component in question may cause a higher-order failure within the system, or it may even cause further secondary effects within other components.

Moreover, the probability of certain failure effects occurring is also dependent upon the probability of their associated failure modes being triggered. This is somewhat complicated by the fact that failure modes can typically be triggered in several ways. For example, consider the possibility that a leak in the fuel line may be caused by factors internal to the component, such as over-pressurization within the line, or external factors, such as a strike from shrapnel caused by another seemingly unrelated failure in another component. Thus, the models that contend with these failure modes must be able to take into account the ability for other models to potentially cause them—the probability for a fuel line leak to occur is then a function of other probabilities, including the probability for over-pressurization to occur and the probability for shrapnel to break through the tank and strike the fuel line. The second of these factors is also comprised of certain probabilities, such as the probability for the tank to fail by shrapnel breaking through it in the first place and the probability that, given the fact that shrapnel has broken through the tank, the shrapnel will strike the fuel line. In addition, any secondary effects that can be induced by a leak in the fuel line are also dependent upon all of these probabilities. In fact, some of these dependencies are circular; a fuel line leak may cause shrapnel to break through the tank, or vice versa. This problem can be solved through iteration, a form of which naturally occurs when the overall risk model is run numerous times and appropriate data is collected.

In terms of probabilities of the two leak-causing events occurring, the overall probability of a leak in the fuel line can be quantified by their associated probabilities as follows:

$$
\begin{aligned}
P(\text{Leak}) = \; & P(\text{Over-Pressurization} \\
& \quad \text{Causes Leak}) \\
& \times P(\text{Over-Pressurization} \\
& \quad \text{of Fuel Line Occurs}) \\
& + P(\text{Shrapnel Striking Fuel} \\
& \quad \text{Line Causes Leak}) \\
& \times P(\text{Shrapnel Striking Fuel} \\
& \quad \text{Line} \mid \text{Shrapnel Breach}) \\
& \times P(\text{Shrapnel Breach Occurs}))
\end{aligned}
$$

With virtually endless scenarios that could cause a certain failure to occur, the overall picture of component and system risk becomes more complicated than may have been previously foreseen. However, the reduction of these risks to mere probabilities in lookup tables eases the process of assessing risks due to the ability to determine these probabilities beforehand with physical models, then chain them together using calls to these lookup tables. If the risk models are designed to interoperate, then they can create a more accurate representation of the probabilities associated with failures within the system.

2. MOTIVATION FOR THE GEOMETRIC TOOLKIT

The magnitude of secondary failure effects is dependent upon several parameters. In the case of the fuel line leak example, such factors may include the material properties of nearby components, a quantification of the shape and magnitude of the leak, and the geometric relationship between the fuel line and the components in question. Although the methodology behind solving the problem is different, the scenario involving shrapnel breaking through a tank and striking another component requires similar parameters. Indeed, certain general capabilities, including statistical analysis, visualization, and evaluation of spatial relationships between components within the spacecraft, are required for many of these risk simulations. These and other capabilities have been realized through a series of software toolkits, such as the Geometric Toolkit.

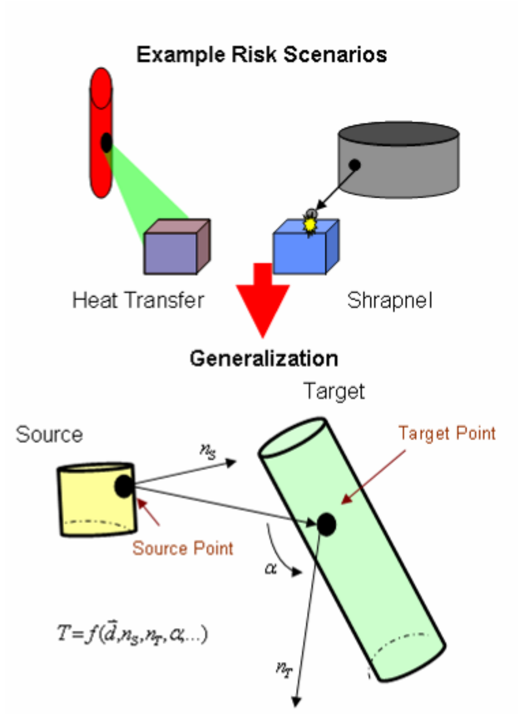


Figure 3. Generalization of the problem of a "source" component affecting a "target" component.

Regarding the problem of geometric relationships between components, many of the scenarios involved one "source" component affecting another "target" component. In many cases, the degree to which the target is affected depends upon certain key parameters in the geometric relationship between the two components as well as the relationship between points on their surface. Heat transfer from a leak in a fuel line is strongest in a direction normal to the line's surface and emanating from the point at which the leak occurs, and its strength tapers off as the angle of incidence increases. Similarly, a piece of shrapnel will travel in a straight line (some statistical deviation from the tank's surface normal) and could strike a target. It will transfer much more energy to the target if the angle of incidence is small. Figure 3 shows how these example scenarios can each make use of the same geometric parameters for analysis purposes.

3. IMPLEMENTATION OF THE GEOMETRIC TOOLKIT

As discussed, the problem of assessing the overall risk of a spacecraft had been divided into a series of models that can be connected together in a meaningful fashion. The set of risk models is said to be *composable*—that is, that the components can be combined into a variety of combinations to suit various purposes [2]. This is similar to the concept of interoperability, although it may not necessarily imply an underlying distributed simulation framework. Specifically, these composable engineering simulations are closest to the definition of the *common library, modular* approach to composability. In essence, this means that the individual models can be seen as reusable tools or modules designed by a team and belonging to a large library of modules that are designed to work together [3].

Due to the success of this approach to composability in the development of the individual risk simulations, it was decided to follow a similar approach with the creation of the Geometric Toolkit. Just as the set of cooperative risk models can potentially answer many more questions than standalone models, a large task such as defining geometric relationships can suit a wider variety of applications if it were split into a set of composable tools. Each tool in the toolkit has a very specific purpose in mind, and the tools can be chained together to solve more complex problems.

The programming environment of choice for the bulk of the team's analysis is MATLAB, primarily due to its visualization capabilities, its overall familiarity, and its ability to perform rapid calculations involving matrices and vectors. The Geometric Toolkit heavily utilizes MATLAB's visualization and matrix manipulation capabilities.

Two general approaches to this geometric analysis problem were taken in parallel; the Patch Toolkit and the Analytical Toolkit are both subsets of the Geometric Toolkit. The Patch Toolkit works with MATLAB "patches," which are effectively polygonal representations of complex geometries within MATLAB. The Analytical Toolkit works with mathematically-defined geometries, such as cylinders and spheres, and can produce rapid results for these specialized cases. In some scenarios, tools in both of the subset toolkits can also work together toward a common goal.

3.1. The Patch Toolkit Subset

Like many complex engineering systems, the spacecraft engine analyzed by the risk assessment team has been represented in CAD by its designers. Using methods inspired from a MATLAB script generously posted online (see [4]), the team created a tool to import CAD depictions of components under analysis as the first tool in the Patch Toolkit. The output of this tool, a MATLAB patch representation of CAD components, could then be used by other analysis and visualization tools in the toolkit. This process helps ensure more accurate results in the geometric analysis, since the shapes of the components can be more accurately represented if they come directly from the components' CAD designs.

Given again the assumption that the bulk of an emanation from a source geometry occurs normal to the point on its surface where it "leaks," the next step would then be to determine how a target may potentially be affected by this emanation. The approach was to develop a method to determine the "hit" point on a target geometry for every possible emanation point on a source geometry, given two or more components imported and ready for use by the routine.

In keeping with the concept of composability, this step was broken down into a few interconnecting pieces. First, a tool was developed to determine the intersection of a target geometry with a three-dimensional ray, as defined by a single point and a vector to determine direction. Then, another tool creates these rays by taking a source geometry and determining vectors normal to each point on its surface. Finally, yet another tool wraps over these and the shape importation tool in order to tie them all together.

Keeping all of these distinct steps in separate tools allows for a large degree of customization to suit the problem at hand. For example, instead of analyzing each surface point on the source, one can use another tool to determine several random points where a source may leak in order to speed analysis. Using another tool, emanations can bend from the normal vector to varying degrees. Yet another tool is present to supplement analyses of vectors that do not intersect with the target.

It has been demonstrated that, through the extended use of the basic geometric tools in the Patch Toolkit, much more appropriate and accurate analyses can be performed. For example, complex heat transfer models or energy models can be injected into the chain of tools at appropriate points in order to receive and process geometric data about the source and target, then produce results appropriate to solving the problem at hand.

3.2. The Analytical Toolkit Subset

MATLAB patches are represented as polygons, and many of the methods used to analyze geometric relationships between shapes must rely on "brute-force" methods. In any case where engine components can be represented as simple shapes, an analytical, mathematical method would be more appropriate, as it would lead to more rapid and more accurate calculations. Indeed, the Analytical Toolkit is derived from this concept. Many components under analysis are pipes, which can be divided into cylinders and "elbows" (sections of toroids), so the team started with algorithms to define relationships between geometries of those types. These algorithms are the product of rigorous mathematical derivation and are detailed in a separate 46-page document [5]. While only the tools related to cylinders have been implemented and validated, development of tools to work with other shapes is underway.

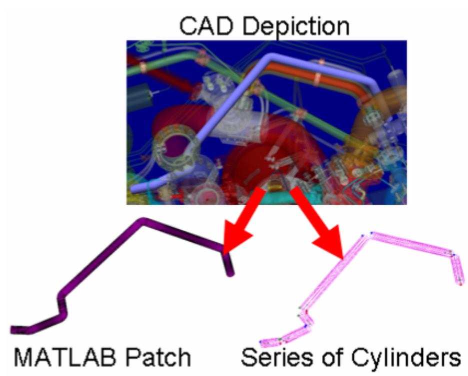


Figure 4. Importing an engine component's shape into MATLAB as a patch and as a set of cylinders.

Similar to the CAD importation tool in the Patch Toolkit, a tool was created to define the cylindrical portions of components so that relationships can be defined analytically. The tool can create an analytical cylinder for use with other Analytical Toolkit tools, using three points on the cylinder as input. One can find these points very easily with a CAD package such as Pro/ENGINEER,. Figure 4 shows an engine component represented in MATLAB as both a patch and as a series of mathematically-defined cylinders.

The geometric relationship tools in the Analytical Toolkit have similar aims as those in the Patch Toolkit, such as calculating factors like the angle of incidence between a point on the source and a point on the target, visualizing the components and vectors, and so forth. The way in which these tools can be combined and recombined is also similar to the way in which tools in the Patch Toolkit can be composed.

4. CONCLUSIONS

The Geometric Toolkit has been used successfully in Ares risk analysis applications, providing geometric relationship data to simulations that carry out the more specialized physics-based analysis for certain risk scenarios. The concept of composable risk simulations has led directly into the concept of composable geometric relationship tools. Due to the sensitive nature of these studies, the exact results cannot be explained in detail; however, it can be stated that having the capability to define geometric relationships between components—as well as the ability to exercise the tools' composability and flexibility—has aided studies similar to the examples that have been discussed.

It has been demonstrated that maintaining a set of relatively simple tools that can work together is often much more worthwhile than complex "end-to-end" solutions. By their nature, the latter category of software solutions have predefined boundaries—their ends—that prohibit their continual reuse to solve problems in the future. By contrast, the small, modular tools owe their distinction to their more streamlined and straightforward capabilities, small memory and disk space footprints, and overall degree of customizability. Most of all, they have the ability to work with other tools, which increases their versatility and power without having an ill-defined scope. Other tools can always be developed to interface with them, so their effective scope is always increasing. Indeed, several Unix-based software utilities initially developed several decades ago, such as grep, awk, sed, and vi, are still in widespread use today due to their adherence to this philosophy of modular software design.

Counter-intuitively, a program or model that can be relegated to a mere link in a chain of tools is often more flexible than one that attempts to solve multiple varied problems on its own. A single model that determines, analyzes, and produces the results of one specific scenario is typically rendered useless after its problem has been solved. One can approach a thorough, top-down solution, such as an integrated vehicle risk assessment, by breaking the problem down and providing solutions from the bottom up. One can also take a subset of these tools and apply them to a wildly different problem. The number of possibilities provided by a set of small tools is virtually endless, if they have been designed with composability and cooperation between models as a goal from the outset.

References

[1] T. Campbell. "Top Level PDA Analysis Plan," 16 April 2008.

[2] M. D. Petty and E. W. Weisel, "A Composability Lexicon", *Proceedings of the Spring 2003 Simulation Interoperability Workshop*, Orlando FL, March 30-April 4 2003, pp. 181-187.

[3] E. W. Weisel, M. D. Petty, and R. R. Mielke, "A Survey of Engineering Approaches to Composability", *Proceedings of the Spring 2004 Simulation Interoperability Workshop*, Arlington VA, April 18-23 2004, pp. 722-731.

[4] D. Riley. "cad2matdemo.m," *MATLAB Central*, Online document at URL http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3642&objectType=File, 23 June 2003.

[5] T. Campbell. "Environment Model for GG and Turbine," 29 April 2008.

GREGORY S. REED is a Research Associate at the University of Alabama in Huntsville's Center for Modeling, Simulation, and Analysis (CMSA). He graduated summa cum laude with a B.S.E. in Electrical Engineering from the University of Alabama in Huntsville in 2006. In 2008, he received a M.S.E. in Industrial and Systems Engineering, and he is currently pursuing a Ph.D. in Computer Science. He has been employed at CMSA since May 2006 in various capacities: as an undergraduate Student Specialist, a Graduate Research Assistant, and in his current position as a Research Associate. At CMSA, he has supported the development of Ares launch vehicle risk assessment simulations from several fronts, including the development the baseline risk model and statistical risk templates, creating general software frameworks and libraries for simulations to use, designing a Verification, Validation, and Accreditation (VV&A) plan for the team's models, and providing configuration control and proper security for the project's sensitive data. His other contributions at the research center have included analyzing of the effects of the Base Realignment and Closure on Huntsville's workforce, creating a web-based interface between physicians and patients, developing enhancements for the America's Army training simulator, and lecturing in classroom and professional development settings on various topics related to modeling and simulation. His research interests include operations research, human behavioral modeling, game and decision theory, behavioral economics, and simulation-based training.

THOMAS W. CAMPBELL is currently consulting in the field of Probabilistic Design Analysis for NASA. He has over 36 years of experience working with the Department of Defense in several fields, including systems engineering; technology development; physics-based modeling and simulation; algorithm and software development; intelligence analysis; radars, antenna, and electronic environments analysis; system security engineering; technology transfer, reuse, and insertion; engineering management and program management; and system risk and vulnerability. He received a B.S. in Physics as well as an M.S. in Physics. His Ph.D. work specialized in Experimental Nuclear Physics with a thesis in low-energy nuclear collisions.

Designing a Composable Geometric Toolkit for Versatility in Applications to Simulation Development

Gregory Reed
UAH Center for Modeling, Simulation, and Analysis

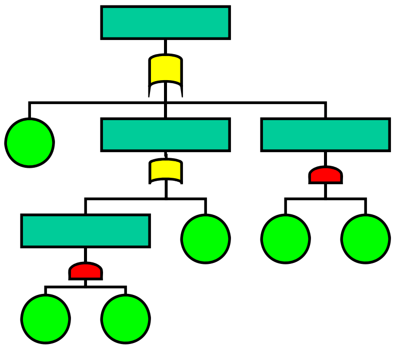
Thomas Campbell
PeopleTec

UAH

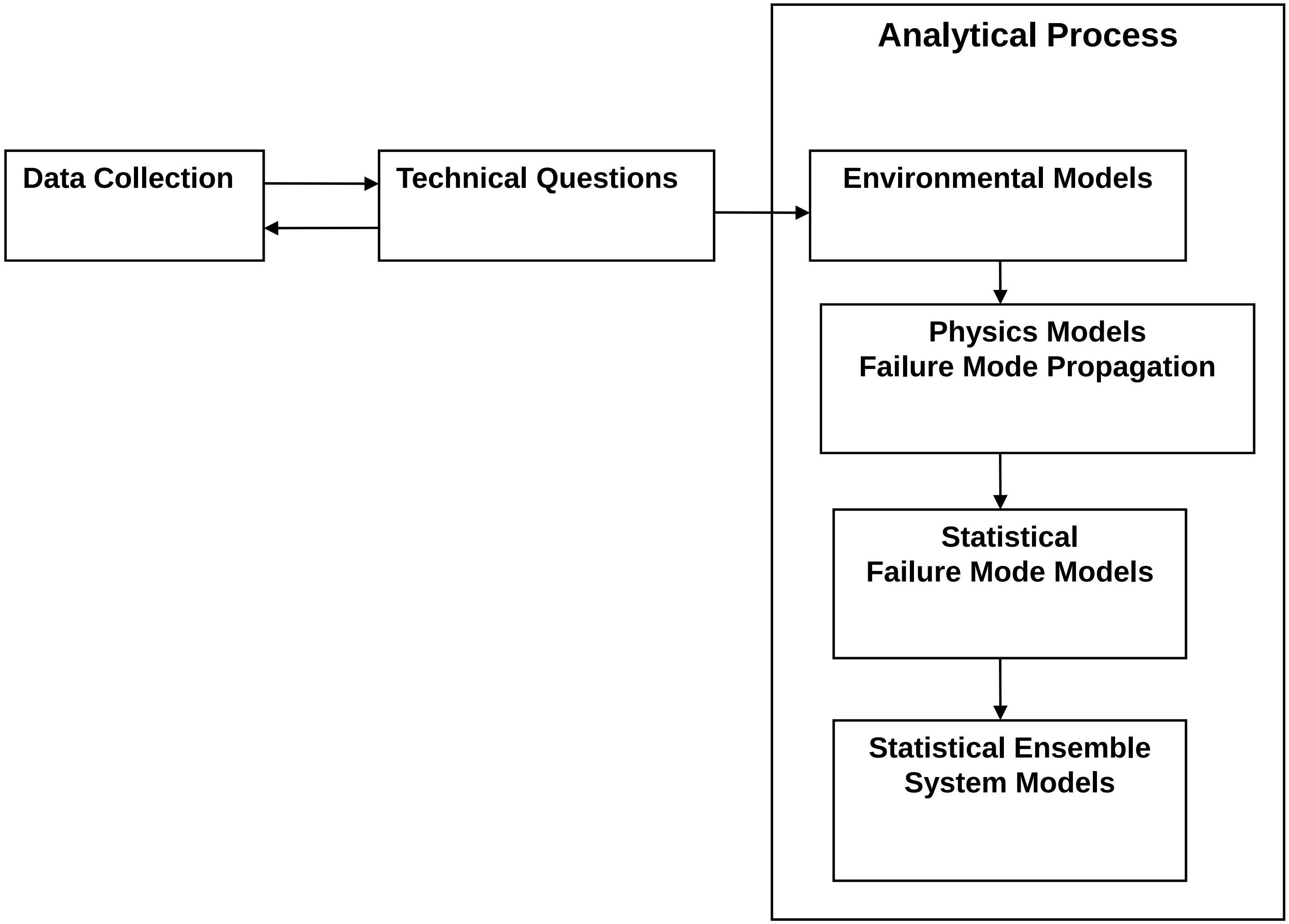
CMSA

Probabilistic Design Analysis

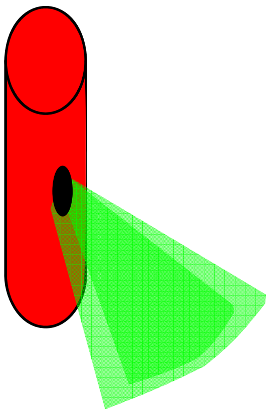
- Team objective: Analyze risks in support of Ares I Crew Launch Vehicle
  - Revolve around “what if” questions/secondary effects/conditional probabilities
  - Overall analysis procedure
    - Select scenarios to investigate
    - Develop standalone engineering simulations for each scenario
      - Designed with interoperability in mind—can identify unforseen risks
    - Analyze the simulation results
    - Validate results against existing system, if possible
    - Provide risk and timing information to several entities
      - Design engineering teams
      - Reliability and safety teams
      - Integrated aborts teams
      - Back to FMEA/Fault Tree
- Many failure simulations share certain characterics
  - Statistical analysis
  - Visualization capability
  - Geometric relationships between components



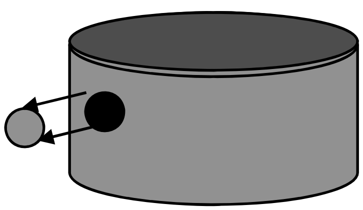
PDA Simulation Development Approach



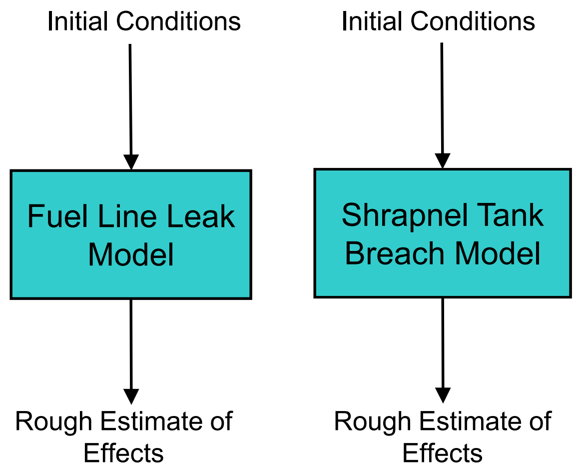
PDA: Examples of Standalone Models



Fuel Line Leak



Shrapnel Breaking Through Housed Tank Casing



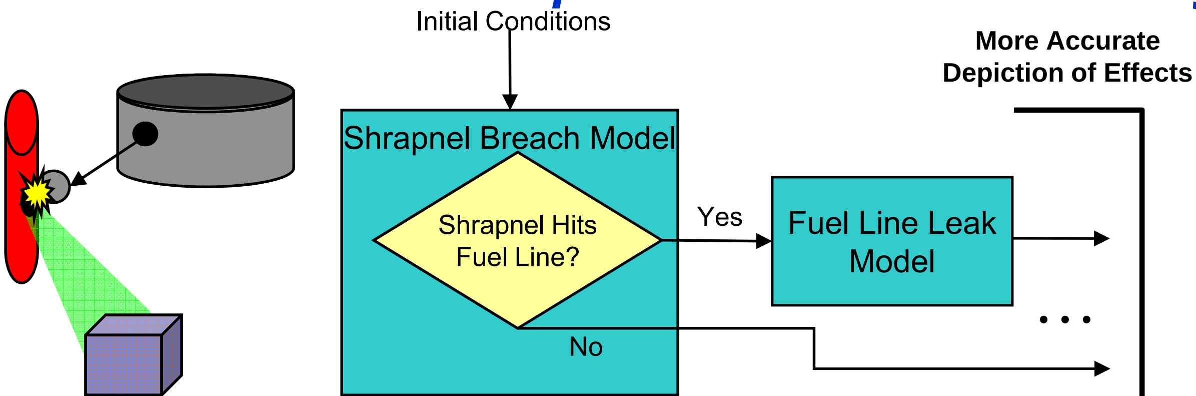
"Given that there is a fuel line leak, what are the effects?"

"Given that there is a shrapnel breach in the housed tank, what are the effects?"

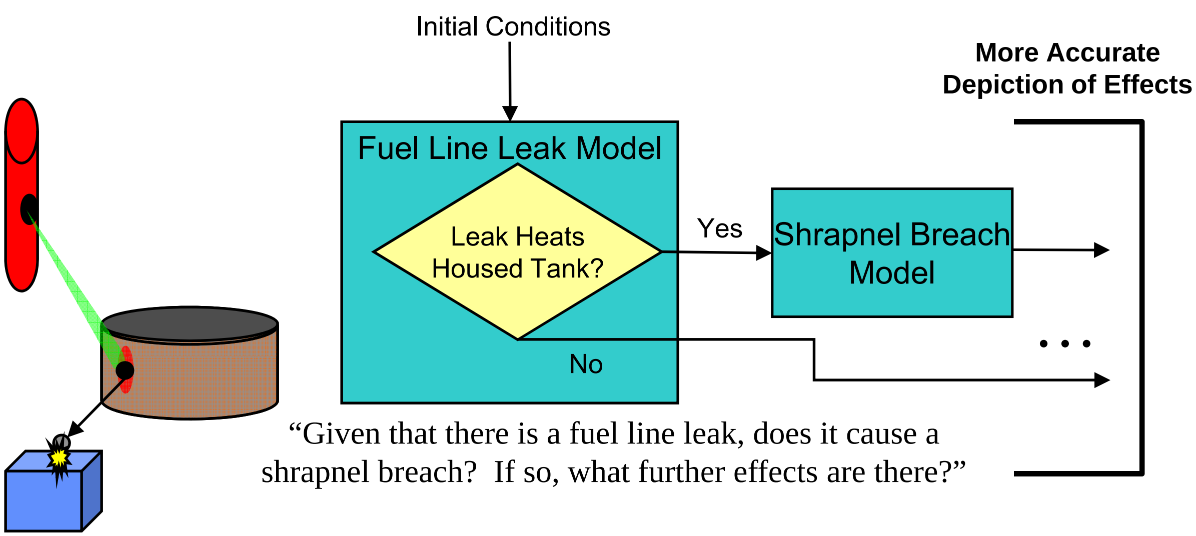
Questions Answered

- Will a fuel leak cause engine shutdown?
- Will shrapnel from the housed tank strike and damage other components?
- Will the breach relieve pressure in the casing?

PDA: Examples of Secondary Effects



"Given that there is a shrapnel breach, does it cause a fuel line leak? If so, what further effects are there?"



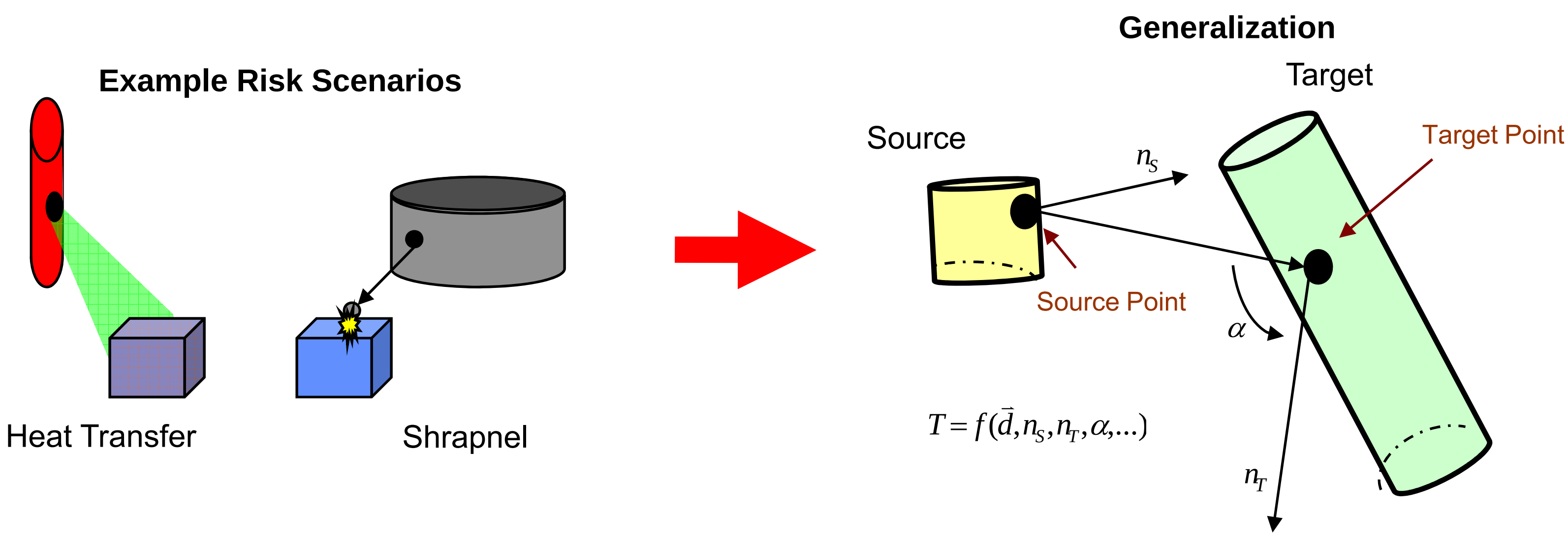
"Given that there is a fuel line leak, does it cause a shrapnel breach? If so, what further effects are there?"

Questions Answered

- Will a fuel leak cause engine shutdown?
- Will shrapnel from the housed tank strike and damage other components?
- Will the breach relieve pressure in the casing?
- Can a breach of shrapnel in the component casing cause the engine to shut down?
- …
- Can a fuel line leak cause pressure to be relieved in the housed tank?
- …
- Can certain other components be affected?
- …

Geometric Toolkit as a Common Framework

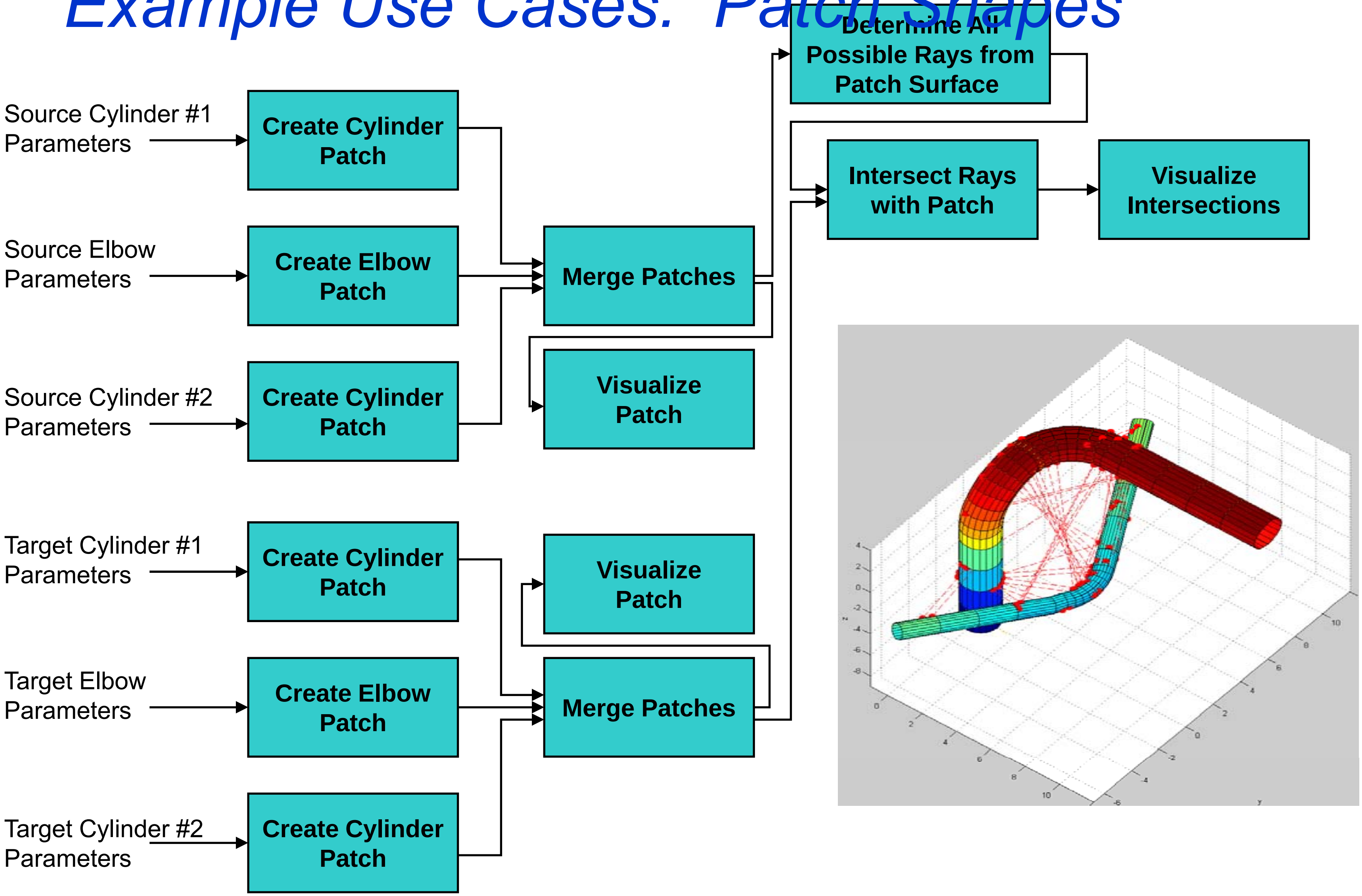
- Initiated by the PDA team's Gas Generator study, but designed for general use
- Developed in the MATLAB programming environment
- Helps evaluate potential effects from points on a "source" geometry on a "target" geometry
- Determines important geometric factors
  - Angle of incidence, normal vector, etc.
- Can import CAD data for real components



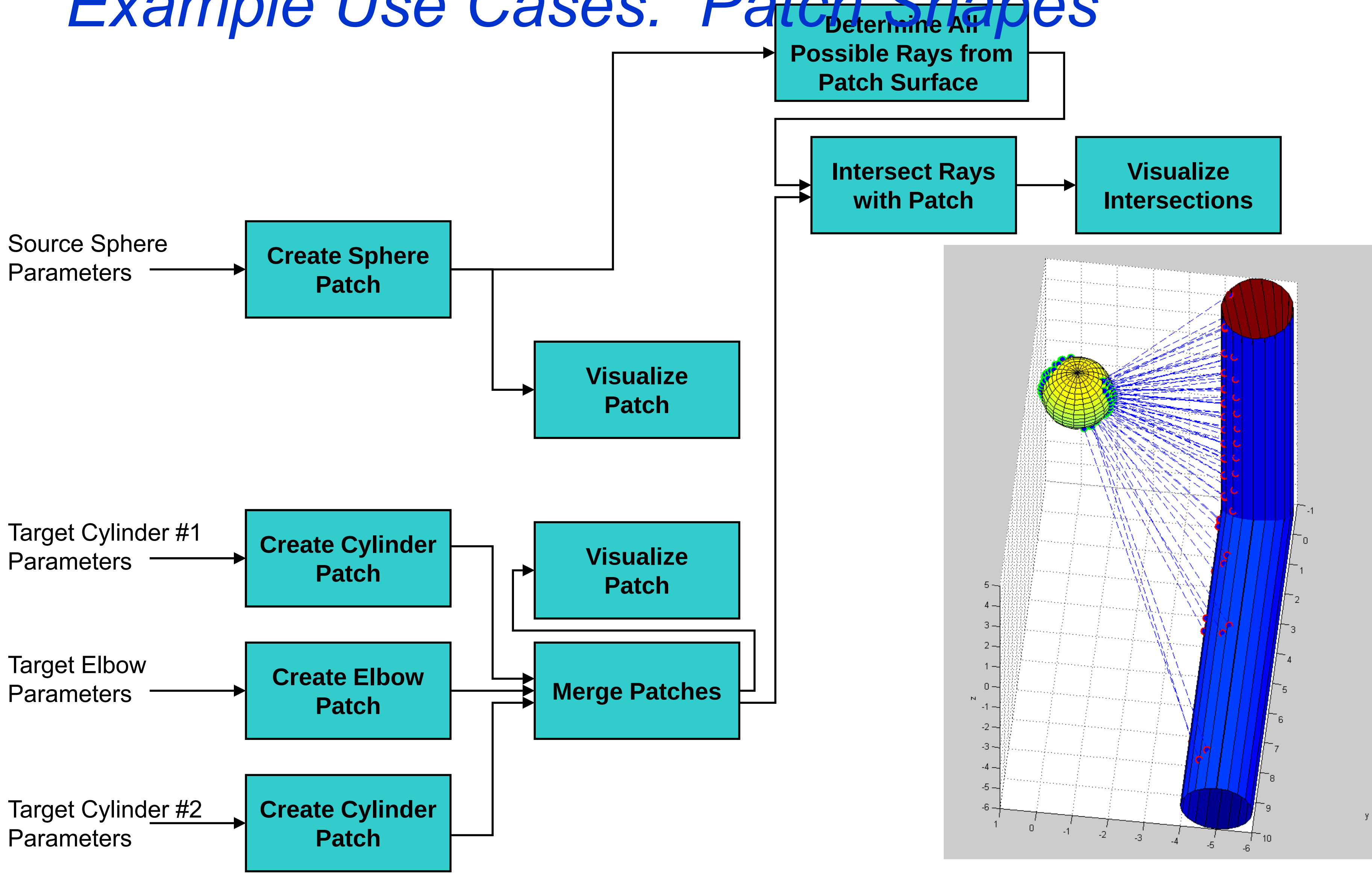
Composability in the Toolkit

- Toolkit contains several specialized tools
  - Analytical Tools: Work with mathematically-defined shapes
  - Patch Tools: Work with MATLAB “Patches”
- Designed to be configurable to suit a variety of applications
  - Inspired by Unix design philosophy
  - Tools have well-defined inputs and outputs
  - Can be strung together in a toolchain
  - Some tools provide redundant functionality, but at a different level
- Modularity of software design aids designing toward composability
  - Breaking large, complicated packages down into smaller, simple ones
  - The whole is greater than the sum of its parts
    - Individual parts are more specialized
    - Small pieces can be connected in a variety of ways
    - Versatility and reliability are increased naturally

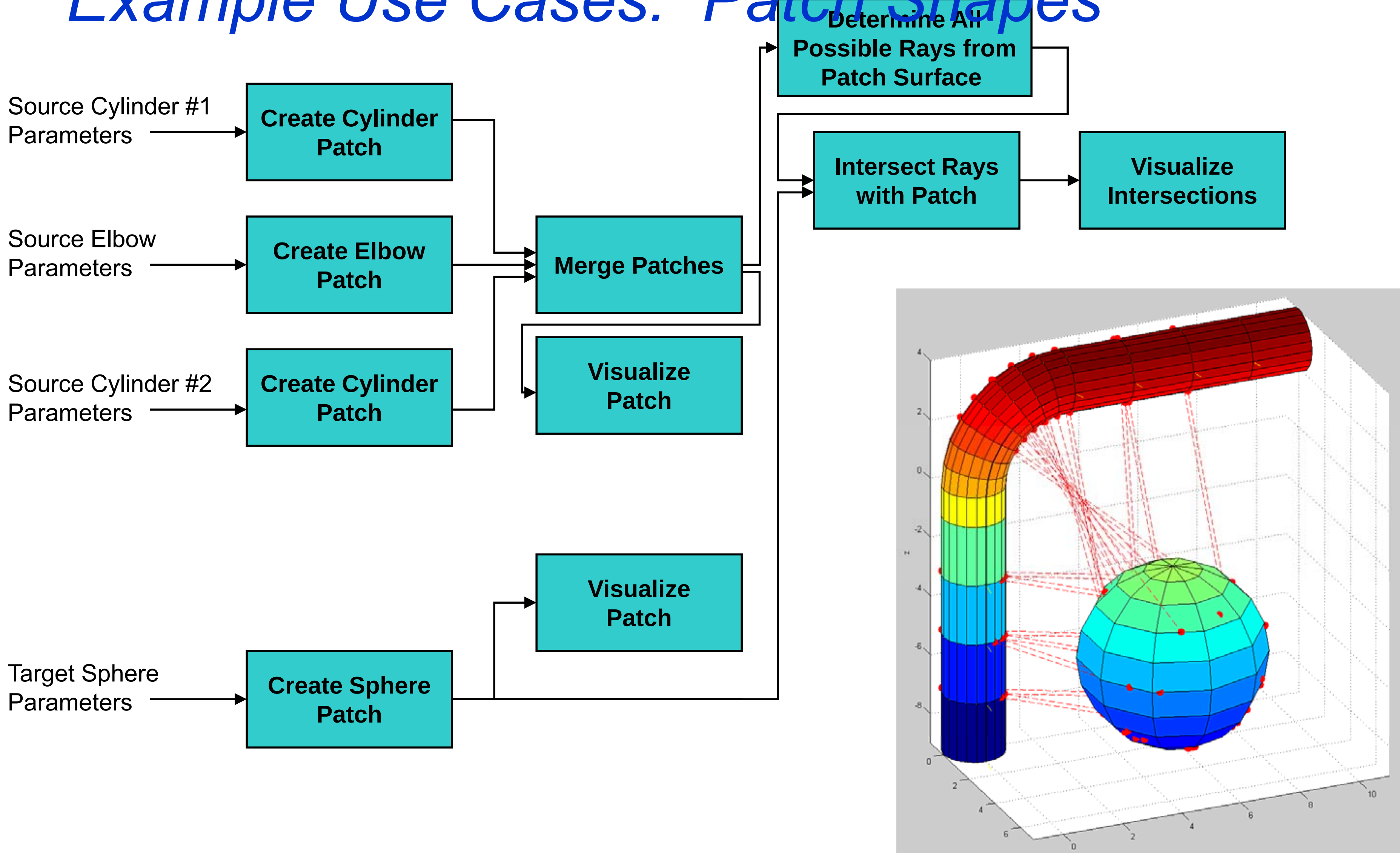
Example Use Cases: Patch Shapes



Example Use Cases: Patch Shapes

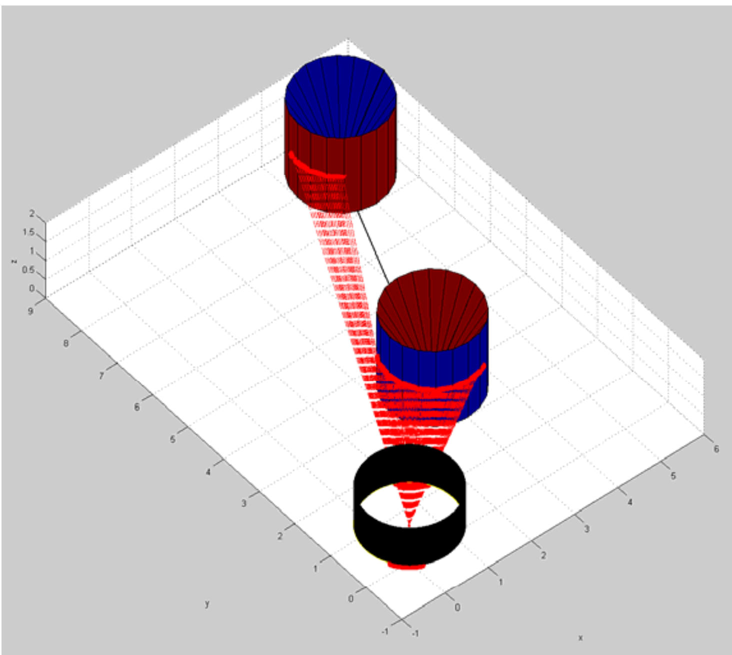


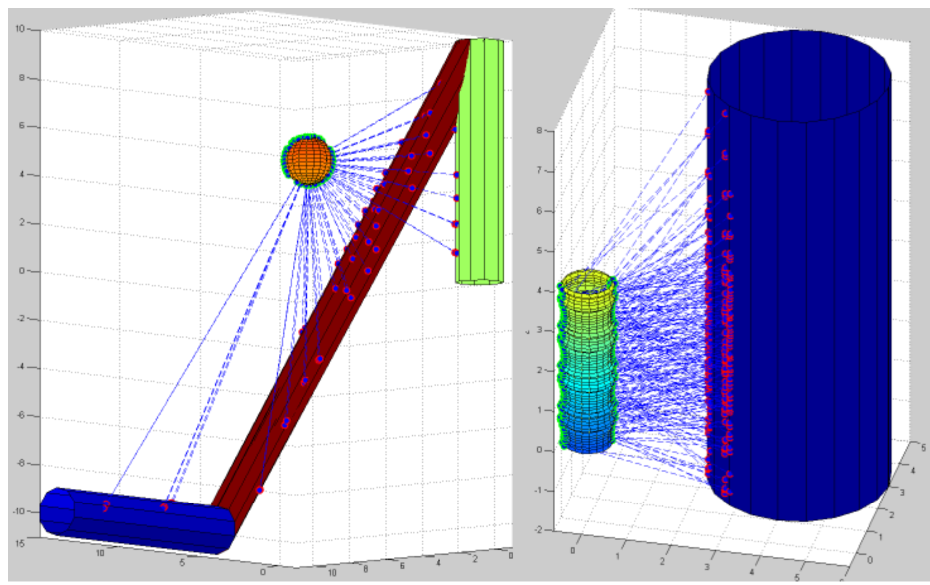
Example Use Cases: Patch Shapes



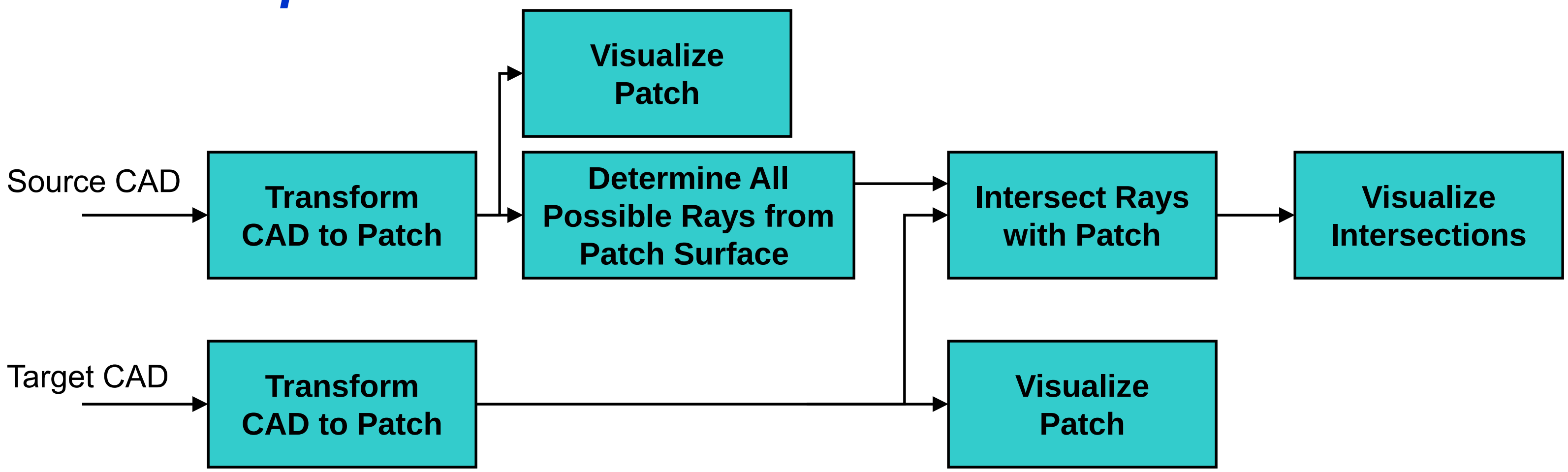
Example Use Cases: Patch Shapes

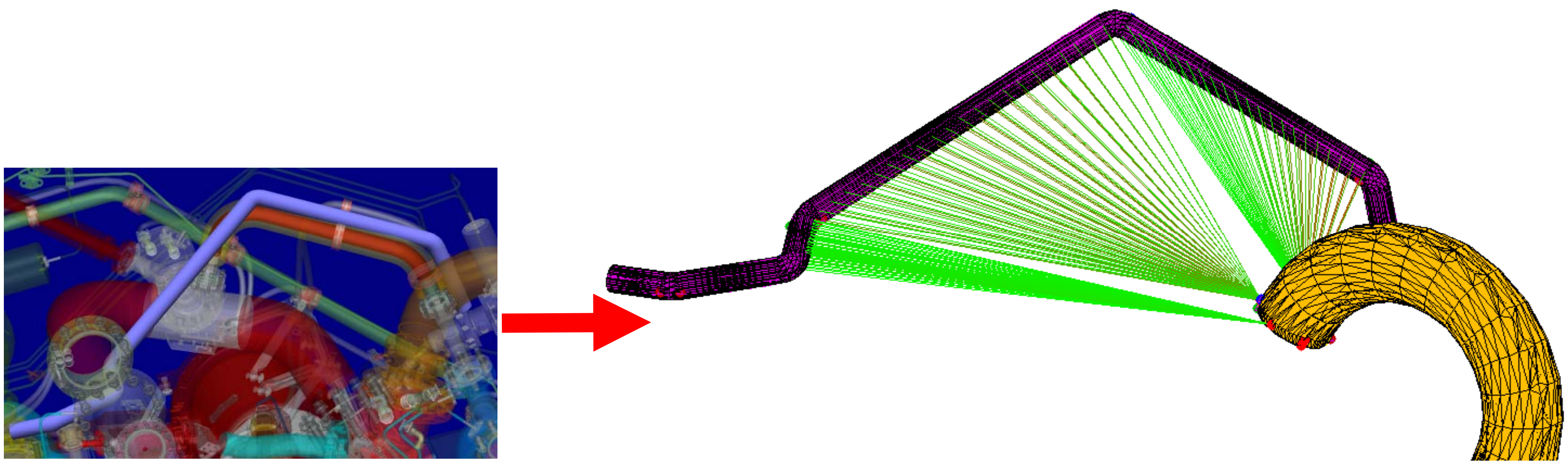
Other configurations are possible with creative combinations of Geometric Toolkit shape creation tools



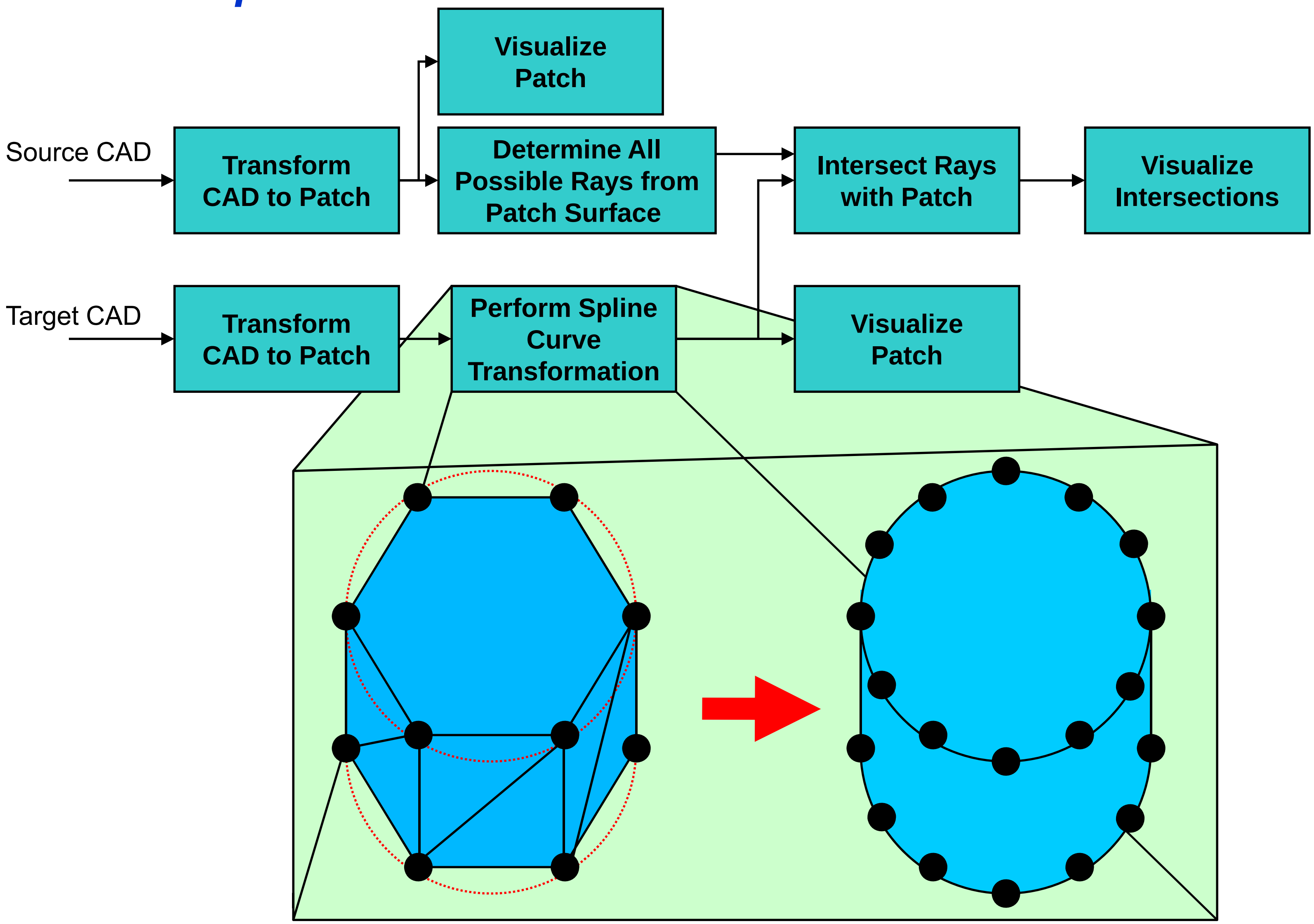


Example Use Cases: Patch from CAD

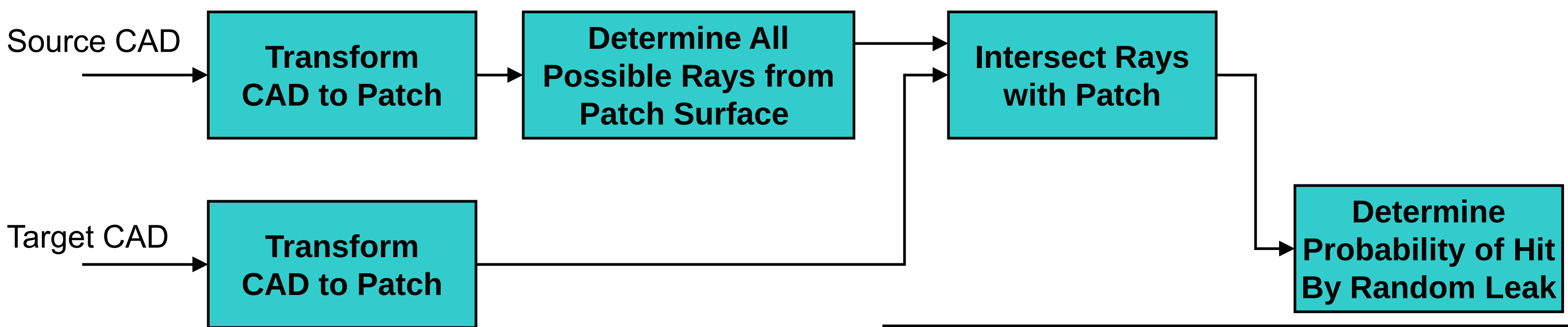


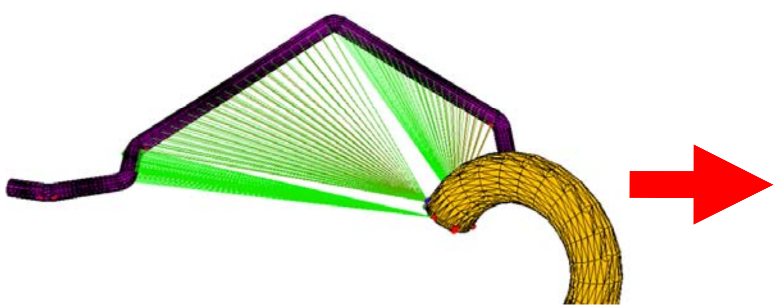


Example Use Cases: Patch from CAD



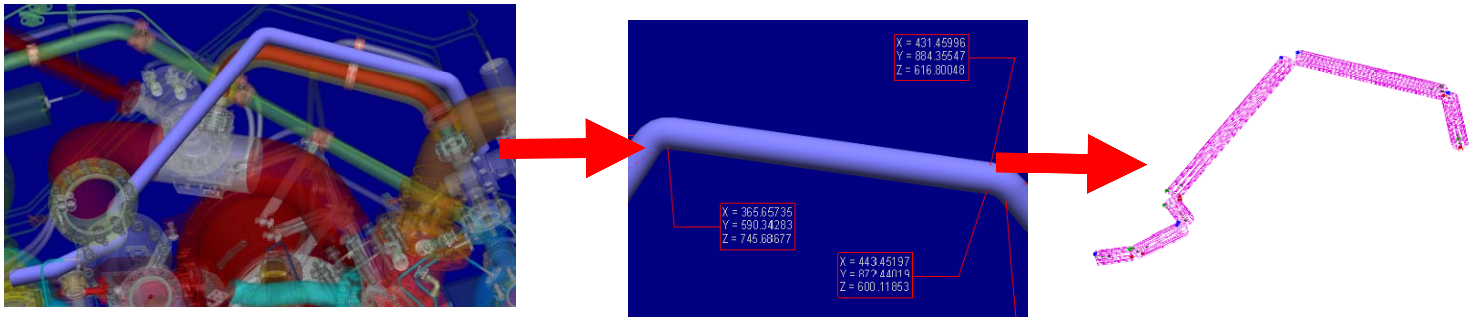
Example Use Cases: Patch from CAD

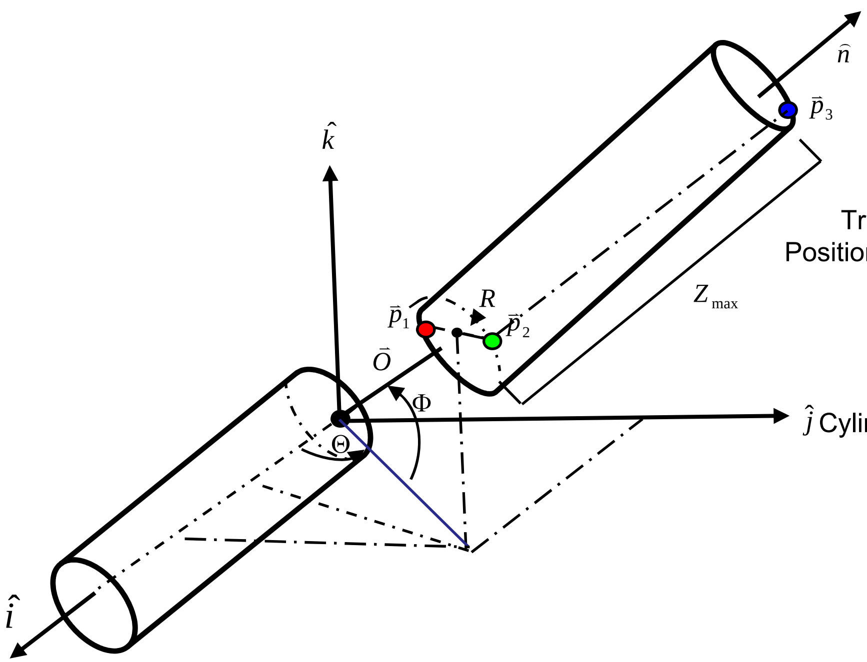




Point	Hit
P_1	1
P_2	1
P_3	0
P_4	1
P_5	0
P_6	1
P_7	1
…	…
Probability	**0.573**

Analytical Toolkit’s Cylinder Definition





Cylinder Definitions

$$Z_{\max} = |\vec{p}_3 - \vec{p}_2| \qquad R = \frac{|\vec{p}_2 - \vec{p}_1|}{2}$$

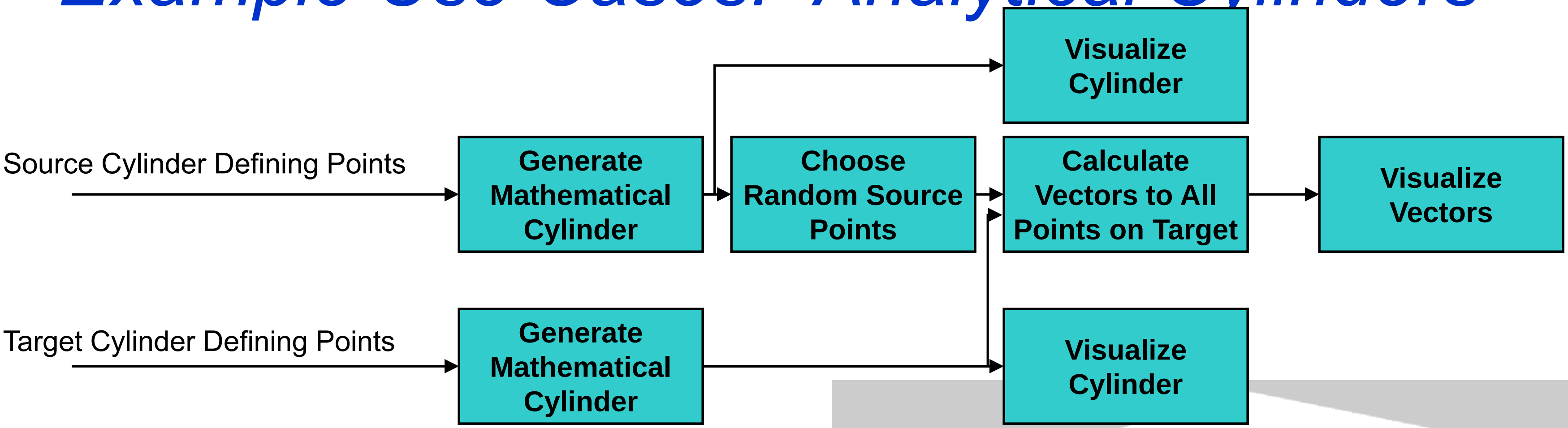
Translation and Position Parameters

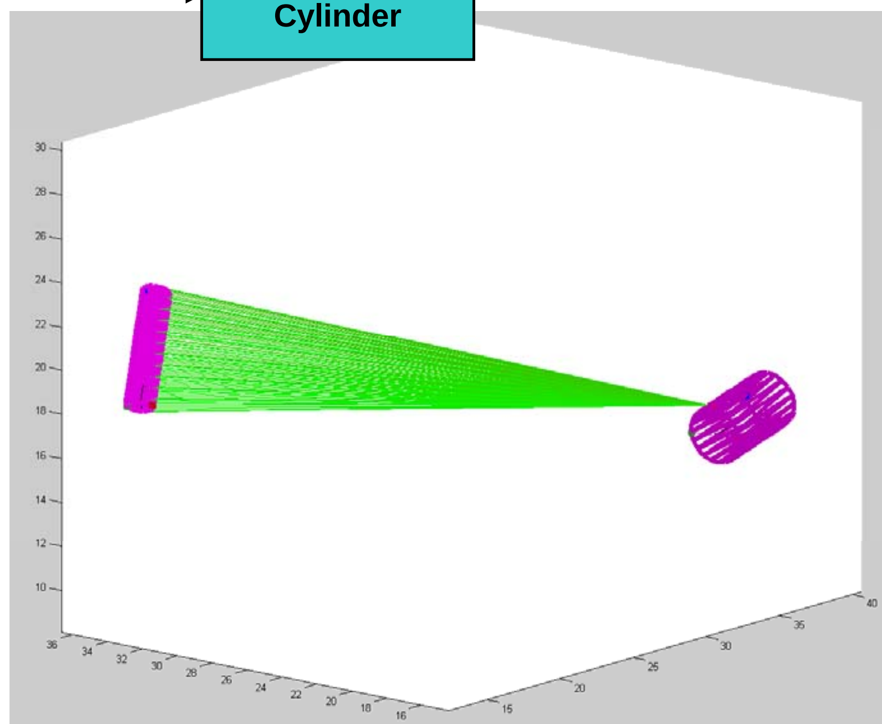
$$\hat{n} = \frac{\vec{p}_3 - \vec{p}_2}{|\vec{p}_3 - \vec{p}_2|} \qquad \vec{O} = \frac{\vec{p}_2 + \vec{p}_1}{2}$$

Cylinder Rotation Parameters

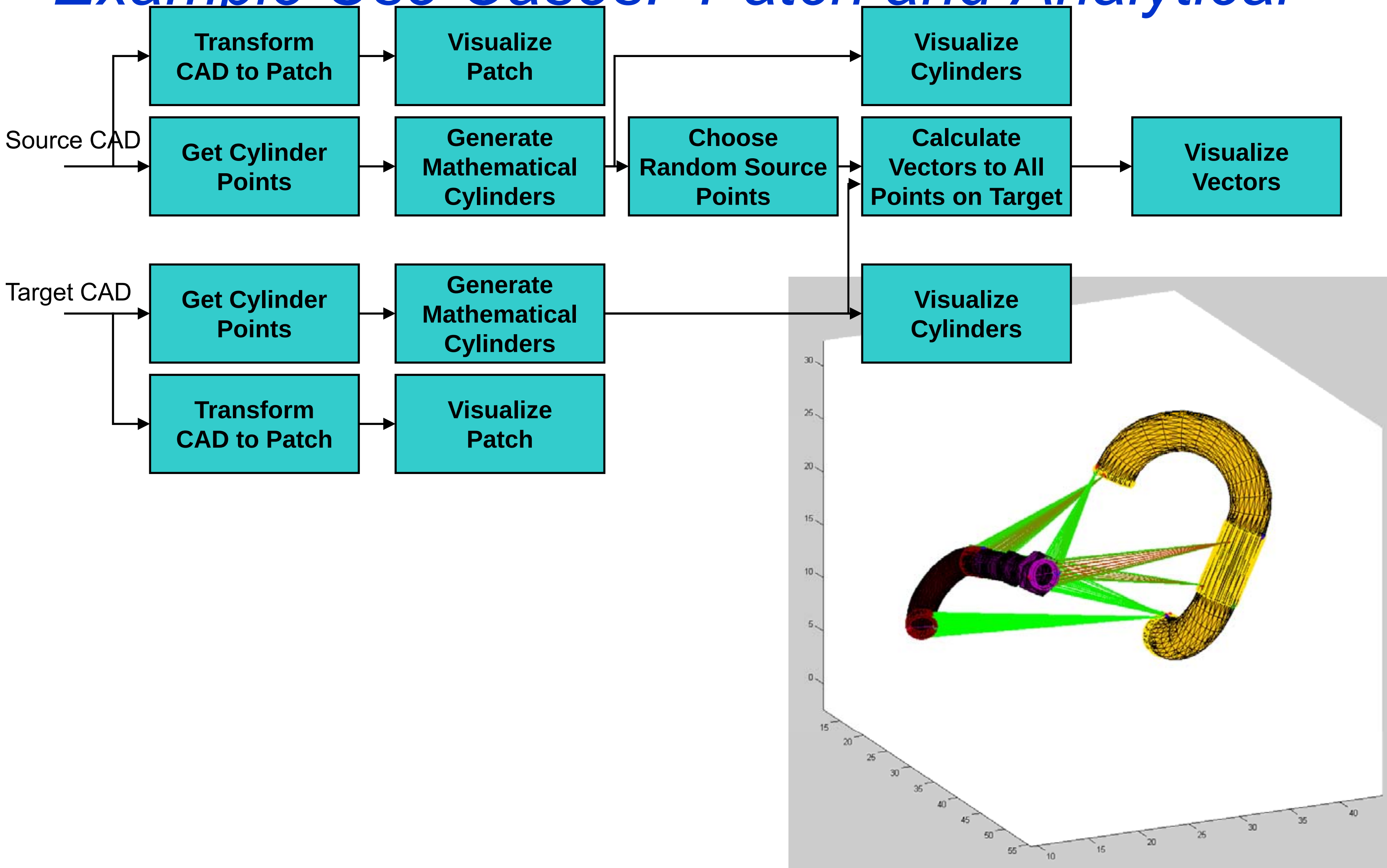
$$\Theta = \cos^{-1}\left(\frac{n_x}{\sqrt{n_x^2 + n_y^2}}\right) \qquad \Phi = \frac{\pi}{2} - \cos^{-1}\left(\frac{n_z}{\sqrt{n_x^2 + n_y^2 + n_z^2}}\right)$$

Example Use Cases: Analytical Cylinders





Example Use Cases: Patch and Analytical

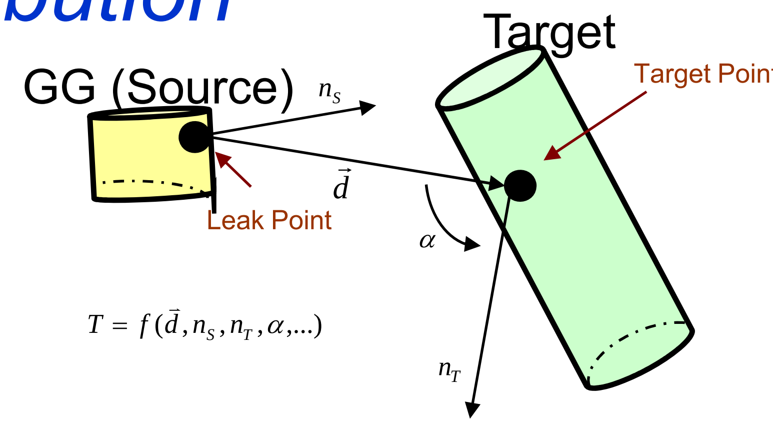


The Gas Generator Study

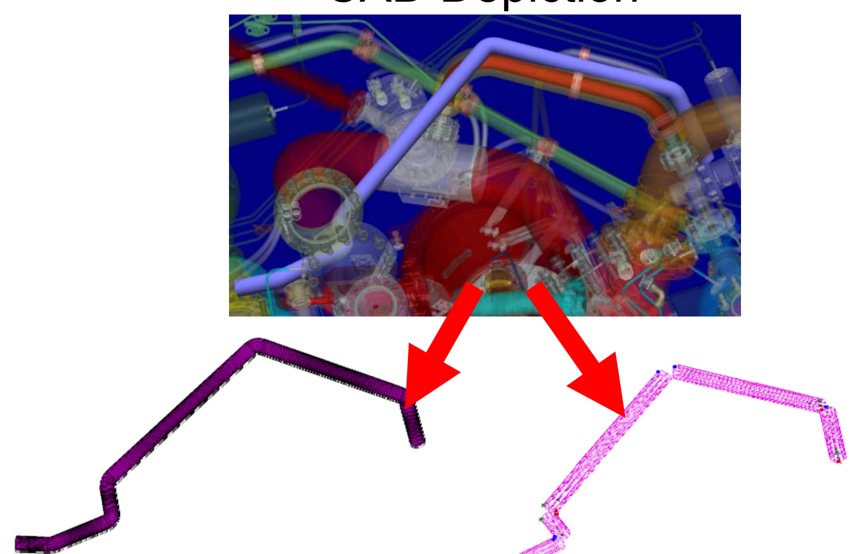
- One of several studies undertaken by the PDA team
- Wanted to answer several questions
  - Which other J-2X engine components are in the vicinity of the Gas Generator's leak point?
    - Can look at CAD depiction
  - Which of these would be most likely to fail from secondary effects?
    - Observe quantity of heat transfer from leak, material properties
  - Which scenarios could potentially lead to a failure in the Gas Generator?
  - What other secondary effects could occur from a Gas Generator leak (e.g. thrust/engine performance)?

Geometric Toolkit's Contribution

- Heat transfer from Gas Generator is dependent upon the spatial relationship between it and a given "target" component
- Many J-2X engine components are composed of pipes (cylinders and elbows)
  - Can model potential leak points on GG using cylindrical coordinate points
  - Can cover most of target areas with mathematically-represented cylinders
  - Can easily derive spatial relationships between points on two cylinders
- Points on existing CAD drawings can define mathematical shapes
- Other tools exist in the Geometric Toolkit to handle more complex geometries



CAD Depiction



MATLAB Patch Series of Cylinders

Geometry and Heat Transfer Logic

GEOMETRY

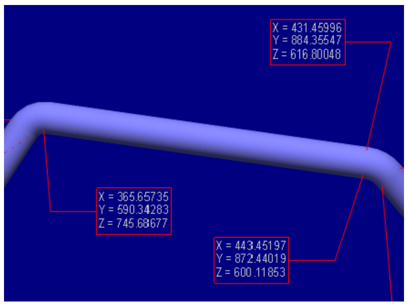
- Partition source into cylinders
- For each target…
  - Partition target into cylinders
  - For each cylinder on the source…
    - For each target cylinder…
      - For every point on the source cylinder and target cylinder…
        - Determine whether the source point can “hit” target point
        - Calculate and record geometric parameters in a matrix

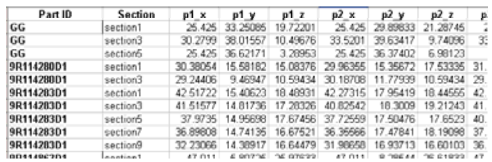
HEAT TRANSFER

- Choose a set of random columns (e.g. points on GG)
  - For each row (all points on all targets)…
    - Pass associated parameters to thermal model
    - Run thermal model and collect transient heat data at each point
  - For each target…
    - Run surface heat transfer FEM
- “Roll up” results statistically

Geometry and Heat Transfer Logic

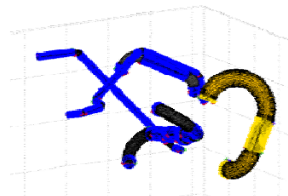
Find and record points in CAD

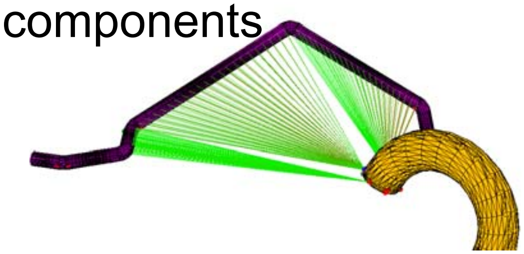


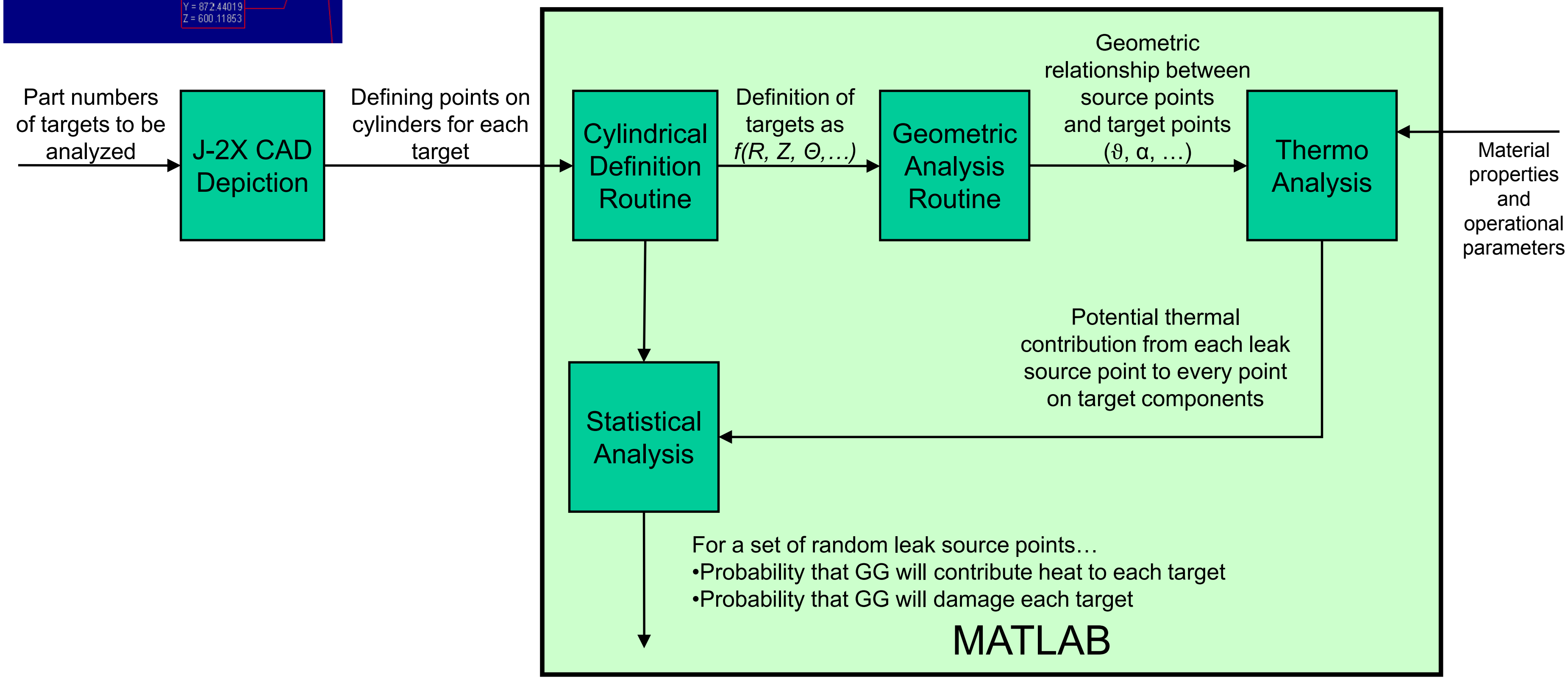


Part ID	Section	p1_x	p1_y	p1_z	p2_x	p2_y	p2_z	p
GG	section1	25.425	33.25085	19.72201	25.425	29.89833	21.28745	2
GG	section3	30.2799	38.01557	10.49676	33.5201	39.63417	9.74096	33
GG	section5	25.425	36.62171	3.28953	25.425	36.37402	6.98123	
9R114280D1	section1	30.38054	15.58182	15.08376	29.96355	15.35672	17.53335	31.
9R114280D1	section3	29.24406	9.46947	10.59434	30.18708	11.77939	10.59434	29.
9R114283D1	section1	42.51722	15.40623	18.48931	42.27315	17.95419	18.44555	42.
9R114283D1	section3	41.51577	14.81736	17.28326	40.82542	18.3009	19.21243	41.
9R114283D1	section5	37.9735	14.95698	17.67456	37.72559	17.50476	17.6523	40.
9R114283D1	section7	36.89808	14.74135	16.67521	36.35566	17.47841	18.19098	37.
9R114283D1	section9	32.23066	14.38917	16.64479	31.98658	16.93713	16.60103	36.

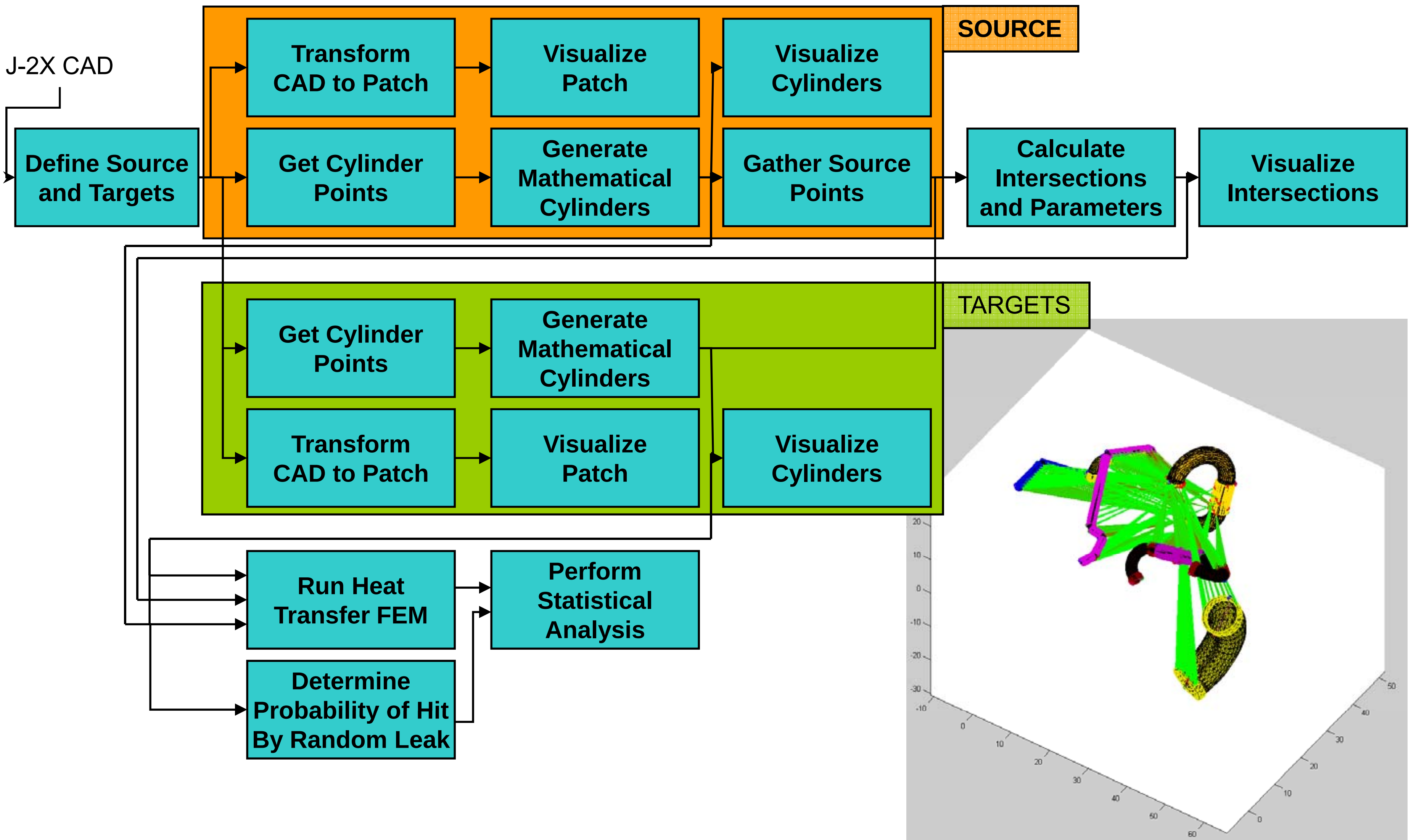
Determine relationships between components



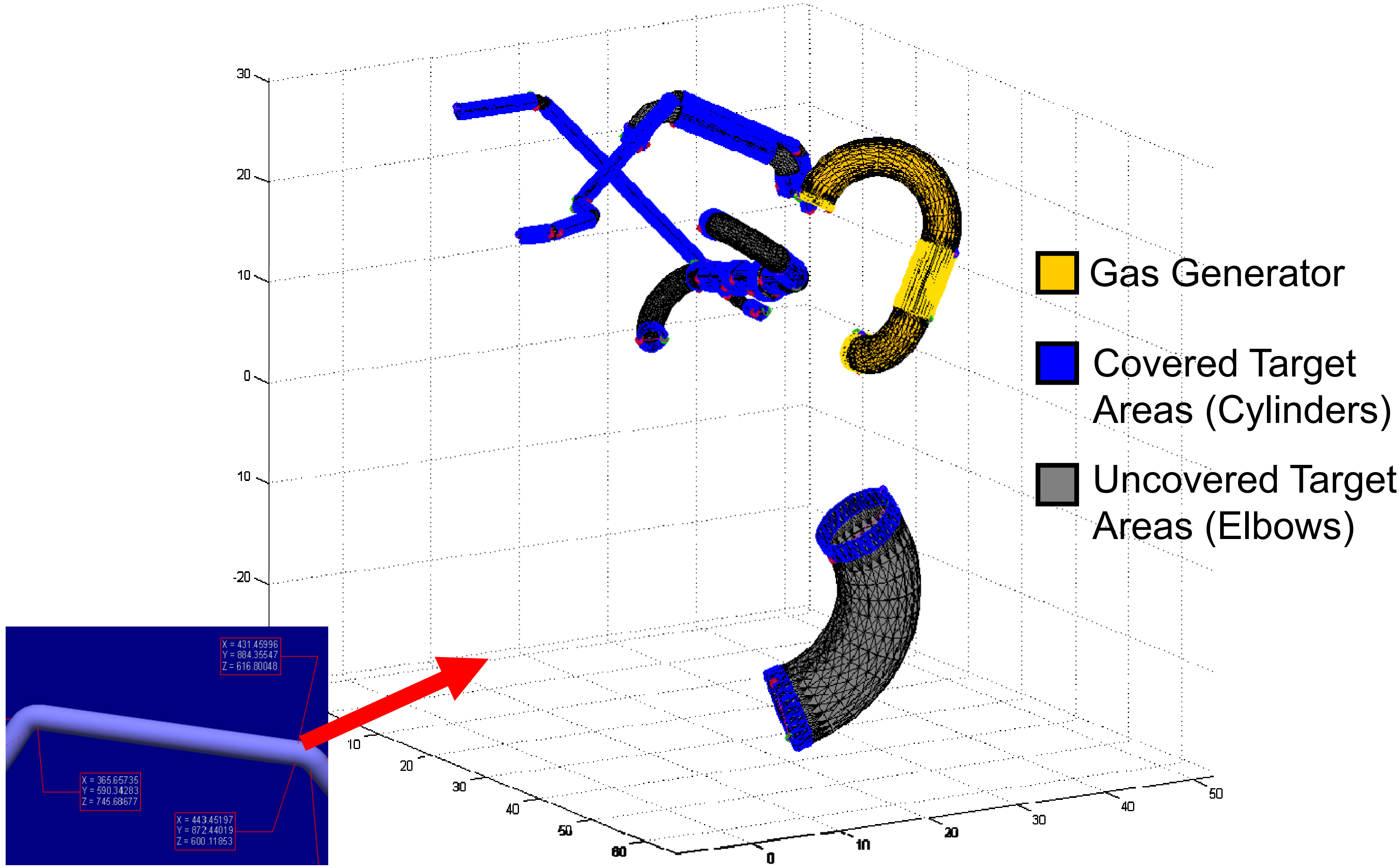




Use Case for GG Problem



Face Validation of Geometric Toolkit



Summary of Study

- Modularity of software design can facilitate model composability
- Many varied problems have certain aspects in common
- The Systems Engineering “V” applies

Contact

- **Gregory Reed**
  UAH Center for Modeling, Simulation, and Analysis
  reedgs@uah.edu

- **Thomas Campbell**
  PeopleTec
  twcjr@comcast.net