AUTOMATED SENSITIVITY ANALYSIS IN EARLY SPACE MISSION DESIGN

Schaus, V.[(1)], Fischer, P.[(1)], Quantius, D.[(2)], Gerndt, A.[(1)]

[(1)]German Aerospace Center (DLR), Simulation and Software Technology
Lilienthalplatz 7, 38108 Braunschweig, Germany
Email: volker.schaus@dlr.de
Email: philipp.fischer@dlr.de
Email: andreas.gerndt@dlr.de

[(2)]German Aerospace Center (DLR), Institute of Space Systems
Robert-Hooke-Str. 7, 28359 Bremen, Germany
Email: dominik.quantius@dlr.de

ABSTRACT

Concurrent Engineering in space mission design features several hundred design parameters provided by more than a dozen domain experts. The quality of the design strongly depends on the individual expertise of the people in the team, their experience and collaboration. As part of the design team, a moderator guides the study. In order to bring the study forward, he must have a good overview, ask the design team the right questions and trigger the next actions while keeping the study objectives in mind. The system design evolves quickly; major changes to the baseline are frequent and have to be taken into account. In this dynamic process it is desirable to quickly identify all the effects of a design change throughout the whole system. The paper presents an automated sensitivity analysis approach to evaluate the current design. This analysis can be used directly during the study to identify design drivers and relationships between individual design parameters. The sensitivity data helps to focus on the right components or subsystems and to guide the design process in a well-founded way. The paper explains the implementation of the automated sensitivity analysis algorithm based on a shared system data model, which is capable of holding all the necessary parameters and equations. The implementation in the DLR software Virtual Satellite and the interaction with the user is discussed. A comprehensible example of laying out the fuel tank of a spacecraft shows how the sensitivity analysis can be interpreted and how it can help to improve the design quality. The same example is later used to point out the local relevance of the sensitivity results. In this case, one design parameter is successively changed by three orders of magnitude. Looking at the sensitivity values in each step, the impact of one parameter is completely shifted to another parameter. This effect is the result of the associated dimension of the design parameters. A second implementation calculating dimensionless sensitivity coefficients is then used to create quantifiable results that allow for comparison between the design parameters. In the concluding section on future work we discuss how the automated sensitivity analysis can be used during an actual Concurrent Engineering study with several hundred design parameters.

INTRODUCTION

Concurrent Engineering (CE) has been used for space mission design since more than 12 years. Over the years, more and more design facilities were inaugurated in Europe and in the whole world. In recent years, one tendency was to develop software data models that represent the system design hierarchy, parameters, calculations and the complex interconnections and dependencies. These underlying models are the foundation for applying Model-based Systems Engineering (MBSE) methods to the design process. The German Aerospace Center (DLR) has its own Concurrent Engineering Facility (CEF) at the Institute for Space Systems in Bremen. In parallel to the normal operation of carrying out design sessions, the DLR Institute of Simulation and Software Technology developed an integrated design environment for spacecrafts called Virtual Satellite [1] in collaboration with the colleagues in Bremen. This software was successfully introduced during a study in December 2011 and will now be more and more becoming a standard tool of the CEF [2].

The software Virtual Satellite is written in Java™ and based on the Eclipse Rich Client Platform. The data model was defined with the Eclipse Modeling Framework and is capable of holding all design information of the early design phases. The domain experts in the CEF use the software to enter their individual design parameters, equations or more complex Excel sheets which can then be committed to a central repository server. In return, they can request an update from the server that downloads the latest design information from other disciplines to the local working copy.

The availability of such a digital master model is the prerequisite for further analysis. In this context, the term analysis has a very broad meaning. It can range from simple charts that visualize static parameter values to formal concept analysis used in knowledge management tools or probabilistic methods. This paper focusses on sensitivity analysis which is doing variations of input parameters and is looking at the effect on the unknown variables of the system. Sensitivity analysis is used in various different fields such as Finance, Chemistry and Physics. Of course Engineering is no exception, in particular it is also applied in the domain of aerospace engineering, e.g. in conceptual airship design [3].

Virtual Satellite already incorporates views with charts and diagrams, such as a mass overview or a time-dependent parameter history [2]. The sensitivity analysis is, so to say, the next step towards more complex analysis methods of a design.

The sensitivity analysis approaches can be categorized in global and local methods. Local methods are more simple and therefore widely used. Most often one uses the so–called one factor at a time (OAT) approach where one input parameter is changed at a time and the behavior of the model system is analyzed and interpreted. This can for example be done by calculating the partial derivatives in a direct way [4]. The local methods have the drawback that they don't look at the interactions between the parameters. This is where the global methods come into play. They tackle the whole problem and in result can also give information about the interaction between the various input parameters. Typically these methods use a large number of input parameter sets generated with Monte Carlo methods. The parameter interaction can be displayed with response surface plots.

This paper focusses on a local sensitivity analysis approach and presents two implementations. The first one uses simply the normalized sensitivity index. In the second approach the sensitivity values are multiplied with a fraction of the current input value to output value, which makes the local derivative dimensionless.

Another method which is also frequently used in the area of design is the so-called Design Structure Matrix (DSM). The DSM analyses the dependencies and interconnections of a complex system design. It can help to identify the necessary process, e.g. which tasks must be run in parallel because they are related. Or one might have the opposite case where the next design step depends on data of another domain which postulates a sequential procedure/approach. Generally DSM analysis focuses more on process, planning and organizational aspects of the design. The journal paper [5] gives a good overview of the different categories of DSMs. Especially the parameter-based techniques result in similar table views than the sensitivity matrix. However, DSM is another research field and there is no implementation in our software yet. Therefore, a comparison between sensitivity analysis and DSM cannot be conducted at this point in time. Nevertheless this is an interesting question and will be discussed in the future work section.

IMPLEMENTATION AND INTEGRATION IN THE SOFTWARE VIRTUAL SATELLITE

The software Virtual Satellite is developed as rich client application based on the Eclipse Framework. It is a modular software project that is easily extensible through the plug-ins concept. One core element is the internal data model of the software. This is the basis for model-based systems engineering. It acts as common system data model and is the single-point-of-truth for the experts. The relevant part of the data model for this paper is depicted in figure 1. It is a very much simplified class diagram just showing the four essential classes and their relationships. At the top, there is the *SystemComponent* class. First of all, it is important to understand that the system design is composed of several *SystemComponents* on different hierarchical levels. Typically, in space mission design these decomposition levels are system, sub-system, element, equipment and sub-equipment [6]. In the Virtual Satellite software however, these levels are all derived from one super class called *SystemComponent*. Every *SystemComponent* can contain several *Parameters* and *Balancings*. *Parameters* are linked to the *Value* class that actually represents the defined value. Every equation of the design is programmed as *Balancing* class and can be instantiated in the running application. In essence, a system design consists of many *SystemComponents* that have *Parameters* and *Balancings*. The domain experts are responsible for defining their parameters and equations and therefore defining a feasible working point as solution to the design problem. Looking at it from a system point of view, the design is a collection of input parameters or boundary constraints and a set of equations that calculate the unknowns of the system. For more information on the internal data model and the comparison to the technical memorandum [6] please refer to the data exchange publication [7].

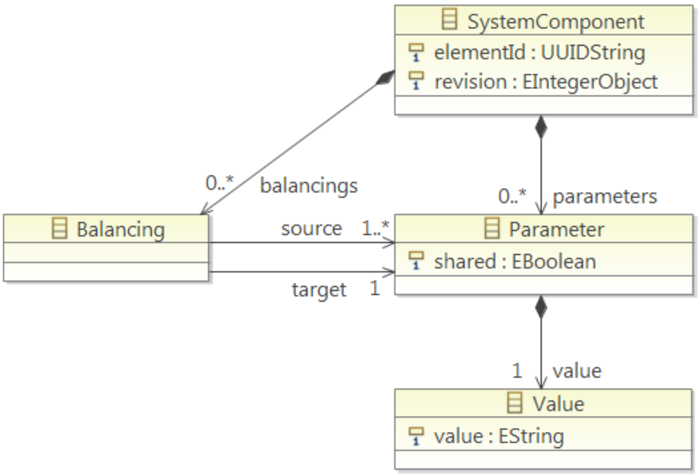


Fig. 1. Simplified class diagram of the internal data model explaining how parameters, balancing equations and system components are handled in the software Virtual Satellite.

In order to calculate the results of all the balancing equations, the software has an integrated solver. This solver first gathers all the balancing equations and builds up the solutions path which is the sequence of equations that need to be solved one after the other to solve the equation system. Of course this only works if there are no cycles in the design where the output of one equation is the input for another an vice-versa. Currently such cycles have to be avoided or they are manually monitored. Such feedbacks result in typical optimization problems which are currently not tackled by the software. It is important to mention that this calculation engine was not implemented specifically for the sensitivity analysis. It was already present in the software to support the CE sessions. The sensitivity analysis just uses the solver after each variation of an input parameter to calculate the effect on the output parameters.

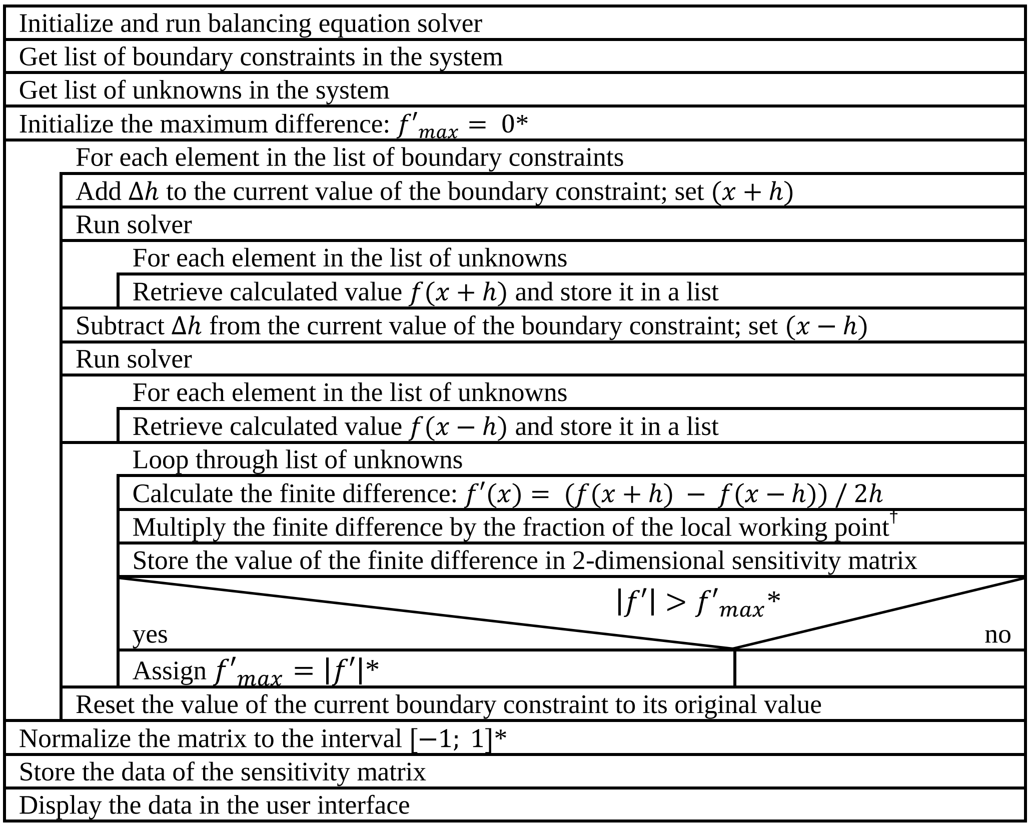
Two ways of calculating the sensitivity index are currently implemented. The first one uses the partial derivative at the working point defined as

$$SI = \left(\frac{dy}{dx}\right)_0. \qquad (1)$$

The second implementation multiplies (1) with a fraction of input to output value at the current working point. This makes the sensitivity coefficients dimensionless and thus comparable among each other [8].

$$SI_D = \frac{x_0}{y_0}\left(\frac{dy}{dx}\right)_0 \qquad (2)$$

The algorithm to calculate the sensitivity coefficients is explained in detail in Fig. 2. At first the solver is initialized and solves the equation system for the first time. This first run is needed to see if the equation system can generally be solved and if there are no cyclic dependencies in the equations system. The next three steps are also initializing steps. The lists of boundary constraints and the list of unknowns in the equation system can be defined by first collecting all the input and output parameters of all equations in the model and then using set theory to decide whether a parameter is considered as boundary constraint or as unknown. Finally, a variable is initialized, that will later hold the maximum gradient of the system. Please note the little star symbol (*) in the fourth line. This indicates that this step is optional whether the user wants the results normalized or not.



```
Initialize and run balancing equation solver
Get list of boundary constraints in the system
Get list of unknowns in the system
Initialize the maximum difference: $f'_{max} = 0$*
For each element in the list of boundary constraints
    Add $\Delta h$ to the current value of the boundary constraint; set $(x + h)$
    Run solver
    For each element in the list of unknowns
        Retrieve calculated value $f(x + h)$ and store it in a list
    Subtract $\Delta h$ from the current value of the boundary constraint; set $(x - h)$
    Run solver
    For each element in the list of unknowns
        Retrieve calculated value $f(x - h)$ and store it in a list
    Loop through list of unknowns
        Calculate the finite difference: $f'(x) = (f(x + h) - f(x - h)) / 2h$
        Multiply the finite difference by the fraction of the local working point†
        Store the value of the finite difference in 2-dimensional sensitivity matrix
        $|f'| > f'_{max}$*
            yes: Assign $f'_{max} = |f'|$*
            no
    Reset the value of the current boundary constraint to its original value
Normalize the matrix to the interval $[-1; 1]$*
Store the data of the sensitivity matrix
Display the data in the user interface
```

Fig. 2. Structogram of the algorithm of the sensitivity analysis

Now, the algorithms runs one loop over all boundary constraint parameters and successively adds and subtracts a small Δh to the current boundary constraint parameter, runs the solver and stores the values of the unknowns. The upper value $f(x + h)$ and the lower value $f(x - h)$ is then used to calculate the symmetric finite difference $f'(x)$. All these local derivatives around the working point are stored in a two-dimensional array. One additional step is needed when the user wants the dimensionless calculation of the sensitivity coefficients (indicated by the cross symbol †). All little conditional block identifies the steepest gradient and saves it. This is later used to normalize the sensitivity matrix to an interval between -1 and 1. The normalization does not alter the results, it just puts the outcome in a well-defined interval, so that it is easy for the user to browse through the sensitivity values and identify the maximum values. Again these steps are marked with the star symbol (*) meaning they are only required if the user wants normalized values or not. Finally, the sensitivity matrix is stored in a supporting data structure that makes it easier to display it in the graphical user interface.

EXAMPLE - DIMENSIONING THE TANK OF A SPACECRAFT

This section introduces an example to better explain the sensitivity analysis and its purpose. Given that a spacecraft needs a propulsion system for orbit maneuvers in space, a typical recurring design task is the dimensioning of the fuel tank. This task already involves three disciplines, namely mission analysis, propulsion and structure. At the beginning mission analysis usually provides a required change of velocity Δv over the complete mission duration including orbit maneuvers, orbit maintenance and disposal of the spacecraft (if needed). The propulsion expert then calculates the required propellant m_p through Ziolkowski's equation (3) considering the in-orbit mass of the spacecraft m_0 and the exhaust velocity of the nozzle c.

$$m_p = m_0 \left(1 - \frac{1}{e^{\frac{\Delta v}{c}}}\right) \tag{3}$$

The next step is to calculate the required volume V_t of the tank. In case the propellant is a liquid the volume V_t can be simply calculated by dividing the mass of the propellant m_p through the density ρ_p of the propellant:

$$V_t = \frac{m_p}{\rho_p} \tag{4}$$

The volume is then forwarded to the next domain expert which is structure. The structure engineer is responsible for selecting the geometry of the tank. Generally, the mass of the tank m_t can be calculated with

$$m_t = \rho_t A \delta \tag{5}$$

where ρ_t is the density of the material, A is the surface area, δ is the thickness of the skin of the tank. Assuming that the tank is spherical, the thickness δ can now be replaced with a rewritten version of Barlow's formula [9] $\sigma_t = \frac{p\,r}{2\,\delta}$.

$$\delta = \frac{p\,r}{2\,\sigma_t} \tag{6}$$

In (6), p is the pressure inside the tank and r is the radius of the sphere. σ_t is the tangential tension in the skin of the tank which must not exceed the maximum allowed tension of the chosen material for the tank.

Selecting a sphere as shape for the tank also defines the necessary equation for the surface area

$$A = 4\pi r^2 \tag{7}$$

Putting (6) and (7) into (5) gives:

$$m_t = \rho_t\, 4\pi r^2\, \frac{p\,r}{2\,\sigma_t} = \rho_t\, 4\pi r^3\, \frac{p}{2\,\sigma_t} \tag{8}$$

Finally r^3 can be replaced with the volume of a sphere, which is equals to the volume of the tank V_t.

$$\mathrm{V} = \frac{4}{3}\pi r^3 = V_t \tag{9}$$

Then (8) results in:

$$m_t = \frac{3}{2}\rho_t V_t \frac{p}{\sigma_t} \tag{10}$$

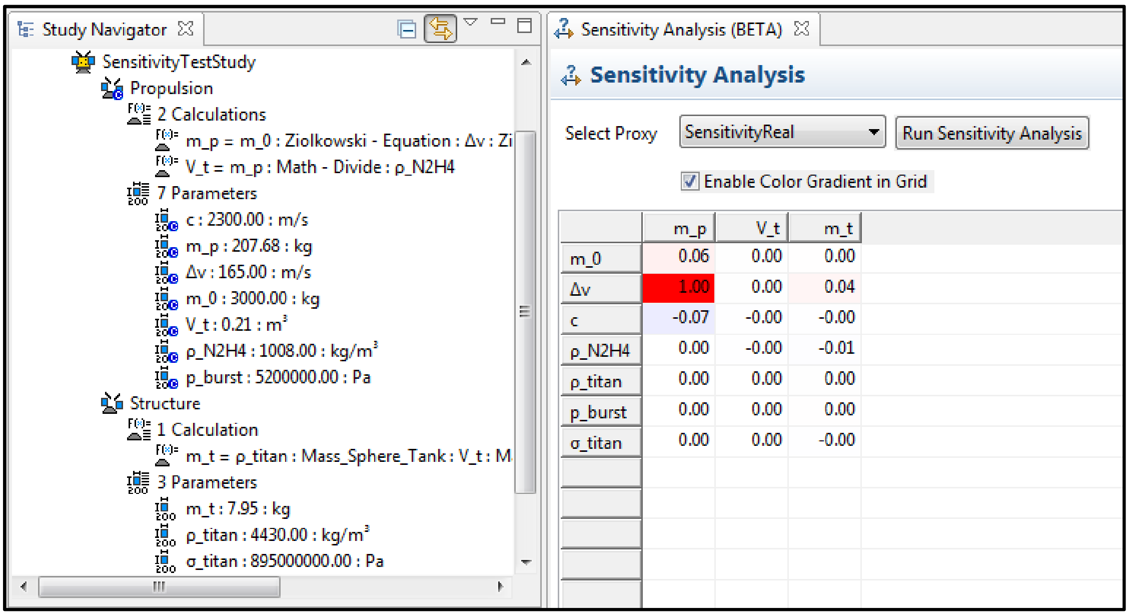
In this design example we can identify three parameters that have to be calculated. So to speak these are the unknowns in this equation system. The remaining seven parameters are boundary constraints. These are set to the following values:

Table 1. Summary of selected and calculated values of the tank dimensioning example; these values define the working point on which the sensitivity analysis will be carried out.

boundary constraints	$m_0 \ = 3000 \ kg$	in-orbit mass of the UV telescope mission AEGIS [2]
	$\Delta v \ = 165 \ m/s$	required delta v for the AEGIS mission [2]
	$c \ = 2300 \ m/s$	exhaust velocity [10]
	$\rho_p \ = 1008 \ kg/m^3$	density of hydrazine [10]
	$\rho_t \ = 4430 \ kg/m^3$	density of the titan alloy [11] used as material for the reference tank [12]
	$\sigma_t \ = 895 \cdot 10^6 \ N/m^2$	tensile strength of the titan alloy [11]
	$p \ = 52 \ bar = 52 \cdot 10^5 \ N/m^2$	burst pressure of the tank [12]
unknowns	$m_p \ = 207.68 \ kg$	mass of the required propellant
	$V_t \ = 0.206 \ m^3$	volume of the needed tank
	$m_t \ = 7.95 \ kg$	mass of the needed tank

INTERPRETATION OF THE RESULTS OF THE SENSITIVITY ANANLYSIS

The sensitivity analysis can be triggered from the graphical user interface of the software Virtual Satellite. It provides a separate view with all the necessary function to run the analysis. Fig. 3 shows a screenshot of the sensitivity view together with the study navigator on the left hand side which provides an overview of the study, the subsystems, the parameters and the calculations. The workflow is as follows. At first all the data needs to be entered in the system, all parameters need to be defined and all the calculations need to be set up according to the example given in the previous section. The values shown in the study navigator are the same ones than in Table 1, which means that the correct working point is entered and the sensitivity analysis can now be run.



```
SensitivityTestStudy
  Propulsion
    2 Calculations
      m_p = m_0 : Ziolkowski - Equation : Δv : Zi
      V_t = m_p : Math - Divide : ρ_N2H4
    7 Parameters
      c : 2300.00 : m/s
      m_p : 207.68 : kg
      Δv : 165.00 : m/s
      m_0 : 3000.00 : kg
      V_t : 0.21 : m³
      ρ_N2H4 : 1008.00 : kg/m³
      p_burst : 5200000.00 : Pa
  Structure
    1 Calculation
      m_t = ρ_titan : Mass_Sphere_Tank : V_t : M
    3 Parameters
      m_t : 7.95 : kg
      ρ_titan : 4430.00 : kg/m³
      σ_titan : 895000000.00 : Pa
```

Select Proxy: SensitivityReal

☑ Enable Color Gradient in Grid

	m_p	V_t	m_t
m_0	0.06	0.00	0.00
Δv	1.00	0.00	0.04
c	-0.07	-0.00	-0.00
ρ_N2H4	0.00	-0.00	-0.01
ρ_titan	0.00	0.00	0.00
p_burst	0.00	0.00	0.00
σ_titan	0.00	0.00	-0.00

Fig. 3. Screenshot of the study navigator and the sensitivity analysis view inside the software Virtual Satellite. Once the working point is entered, the sensitivity analysis can be run. Once the algorithm is finished, the grid on the right hand side is filled with the normalized results of the sensitivity analysis. The color coding ranging from a red (+1) to a blue (-1) helps to quickly interpret the correlations in the data.

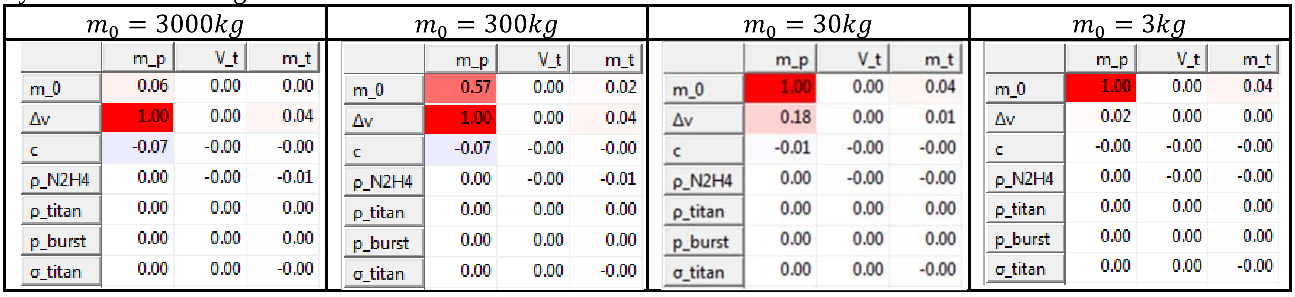
In the sensitivity view on the right hand side of Fig. 3, the only thing that needs to be done is selecting the study which should be analyzed in the drop-down box. After that the user simply can click on the ‘Run Sensitivity Analysis’-button and the grid below is filled with the normalized results of the sensitivity analysis. The rows of the grid are boundary constraints of the equation system, the columns are the unknowns. Again, this corresponds to the example of the previous section. The view also has a little check box that toggles a color coding of the grid. If this is turned on the largest value (+1) is highlighted in red and the lowest values (-1) is highlighted in blue. All values in between are gradually faded with white color.

Looking at the result, eye-catching is the strong connection between required Δv and the mass of the propellant m_p. This simply means that the derivative of $\frac{\partial m_p}{\partial \Delta v}$ has the largest value. The sign of the derivate is positive which means that an increase of Δv will also increase the mass of the propellant. And this is clearly what Physics tells us. Also looking at the relation of the mass of the tank to Δv, it is not as strong as the relation to the mass of the propellant, but still, an increase of the Δv also means increase of the mass of the tank since we have to carry more propellant. And this sounds logical as well. The grid also contains many values which are zero or very close to zero. This either means that there is no effect at all between the two values or it is very little compared to the rest of the sensitivity values. E.g. the material property of titan σ_t has no relation to the mass of the propellant m_p. When looking back to the equations of the example, it's easy to understand this. σ_t only appears in the last equation, when the mass of the propellant is already calculated. So the selection of the material will never influence the mass of the propellant. This derivate will always be zero. The case is a little bit different for the relation between the material constant σ_t and the mass of the tank m_t. Here, the entry in the sensitivity matrix also states zero, but is the minus sign indicates that the value is slightly below zero. Yet again, the explanation can be found by looking at the example. If a stronger material is chosen for the tank, the tensile strength σ_t will increase, which, in result, makes the tank lighter. The mass of the tank decreases and this is indicated by the minus sign.

The sensitivity analysis is only valid locally, very closely around the working point. If a parameters is changed it might strongly affect the sensitivity values. A little example shown in Table 2 gives an overview of the change in the sensitivity matrix when the total mass of the satellite m_0 is changed by three orders of magnitude from 3000kg down to 3kg. All the other boundary constraint parameters are held constant. At the original working point with $m_0 = 3000kg$, the Δv was the design driver and had the strongest effect on the mass of the propellant. It means that changing the Δv by one meter per second has a much stronger effect than changing the mass m_0 by one kilogram. Or looking at it the other way round, the mass m_0 needs to be changed by $\Delta m = 1/0.06\ kg = 16.67kg$ to have the same effect on the mass of the propellant. Whether it is technically easier to reduce the mass of a 3 tons spacecraft by 16 kilograms or to optimize the orbit and reduce the Δv by one meter per second is out of question here. This needs to be answered by the design team, but for the team leader of a CE study this is important information in order to drive the study. The team leader can point out this result in a joint session and then trigger the right domain experts to further enhance their designs, e.g. being precise in margin assumptions.

Now, looking at the second column of Table 2, the sensitivities of m_0 and Δv in relation to m_p are both strong. They have the same order of magnitude which in this particular case means changing the mass m_0 by two kilograms has a similar effect on the mass of the propellant m_p than changing Δv by one meter per second. As the system mass decreases further to 30kg and then finally to just 3kg, the stronger is the influence on the mass of the propellant.

Table 2. Different sensitivity results at different working points; the total mass of the spacecraft is decreased by three orders of magnitude.



$m_0 = 3000kg$				$m_0 = 300kg$				$m_0 = 30kg$				$m_0 = 3kg$			
	m_p	V_t	m_t		m_p	V_t	m_t		m_p	V_t	m_t		m_p	V_t	m_t
m_0	0.06	0.00	0.00	m_0	0.57	0.00	0.02	m_0	1.00	0.00	0.04	m_0	1.00	0.00	0.04
Δv	1.00	0.00	0.04	Δv	1.00	0.00	0.04	Δv	0.18	0.00	0.01	Δv	0.02	0.00	0.00
c	-0.07	-0.00	-0.00	c	-0.07	-0.00	-0.00	c	-0.01	-0.00	-0.00	c	-0.00	-0.00	-0.00
ρ_N2H4	0.00	-0.00	-0.01	ρ_N2H4	0.00	-0.00	-0.01	ρ_N2H4	0.00	-0.00	-0.00	ρ_N2H4	0.00	-0.00	-0.00
ρ_titan	0.00	0.00	0.00	ρ_titan	0.00	0.00	0.00	ρ_titan	0.00	0.00	0.00	ρ_titan	0.00	0.00	0.00
p_burst	0.00	0.00	0.00	p_burst	0.00	0.00	0.00	p_burst	0.00	0.00	0.00	p_burst	0.00	0.00	0.00
σ_titan	0.00	0.00	-0.00	σ_titan	0.00	0.00	-0.00	σ_titan	0.00	0.00	-0.00	σ_titan	0.00	0.00	-0.00

This example shows that the validity of the sensitivity analysis has to be considered always locally around the current design point. However, the dimension of the parameters still affects the results. Looking at the example again, when putting (3) into (4) we get

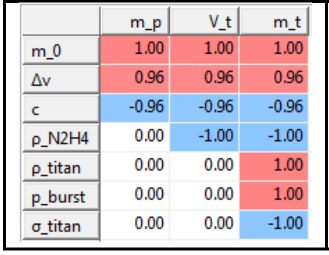
$$V_t = \frac{m_0}{\rho_p}\left(1 - \frac{1}{e^{\frac{\Delta v}{c}}}\right). \tag{11}$$

Equation (11) indicates a proportional parameter interaction between m_0 and V_t, which is not visible in Table 2. The respective sensitivity coefficient is always displayed as zero. In relation to the largest values in the matrix the coefficient for m_0 to V_t is simply too small. The effect of the dimension can be eliminated when using the second implementation of the sensitivity coefficients with (2). Table 3 shows the dimensionless coefficients again for a total mass of the spacecraft of $3000kg$ and $3kg$. The values are the same in both cases. Also all the intermediate steps for m_0 show the same values. Again the color coding from blue to red is used to highlight how large the values are.

The interpretation of these results is different now. The proportional effect between m_0 and V_t is now given by the value 1.0. This means that if m_0 is changed by 10%, the volume of the tank V_t also changes by 10%. So, in contrast to the first implementation where one looks at one unit changes, here we use percent changes to analyze the results.

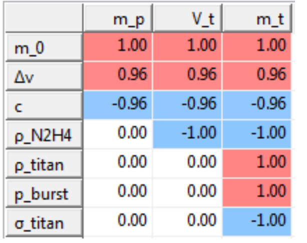
Table 3. Dimensionless sensitivity coefficients calculated with (2) cancel the effect of the dimensions of the parameters and gives the same results when m_0 is decreased.

$m_0 = 3000kg$



	m_p	V_t	m_t
m_0	1.00	1.00	1.00
Δv	0.96	0.96	0.96
c	-0.96	-0.96	-0.96
ρ_N2H4	0.00	-1.00	-1.00
ρ_titan	0.00	0.00	1.00
p_burst	0.00	0.00	1.00
σ_titan	0.00	0.00	-1.00

$m_0 = 3kg$



	m_p	V_t	m_t
m_0	1.00	1.00	1.00
Δv	0.96	0.96	0.96
c	-0.96	-0.96	-0.96
ρ_N2H4	0.00	-1.00	-1.00
ρ_titan	0.00	0.00	1.00
p_burst	0.00	0.00	1.00
σ_titan	0.00	0.00	-1.00

CONCLUSION

The work presented in this paper describes the implementation of a local, direct sensitivity analysis in the software Virtual Satellite. The algorithm and the options for the sensitivity analysis are well integrated in a graphical user interface. This smooth integration makes it possible to run a sensitivity analysis during a CE study. The result of this analysis can be used directly by the team leader to guide the design process. We present a typical recurring task in spacecraft design, the dimensioning of the fuel tank as little example with three unknown variables and seven boundary constraints. The sensitivities indexes are interpreted to show the reader what kind of information can be derived from such an analysis. We present two methods of calculating the sensitivity indexes. The first one simply uses the partial difference and is therefore affected by the dimensions of the design parameters. This enables the interpretation of what happens when a one unit change is applied. The second method divides the derivatives by a fraction of the input to output value at the current working point which makes the index dimensionless. Both implementations are useful depending on what one is looking for. It is necessary to look closely at how things are modeled to understand the behavior of the modeled system and interpret the result of the sensitivity analysis correctly.

FUTURE WORK

This paper just describes a first attempt towards using sensitivity analysis in CE studies. The presented example with in total ten parameters is small compared to an actual CE study that can have several hundred design parameters. One thing that needs to be done is to run this kind of analysis on a real design data set from an actual CE study. This is not as straight forward as one might imagine. Currently the sensitivity analysis only works when all the equations are modeled as Balancing Equations as defined by the Virtual Satellite data model. However, during the studies, many domain experts use spreadsheets for their calculations. An embedded interface to these spreadsheets is available in the software, so inputs can be taken from the data model to a spreadsheet cell and outputs can be fed back into the model a specific cell. But the spreadsheets are considered as external tools and black boxes. Their calculation cannot be triggered automatically inside the software and therefore the necessary calculations for the derivative cannot be computed by the solver. This task requires a larger refactoring of the source code of the software and is not yet completed.

Further, when the user deals with a couple of hundreds parameters, the dimensions of the matrix containing all the sensitivity values will grow large as well. We need to provide smart displaying and filtering techniques to allow intuitive exploration of the matrix. The ideas here are that maybe the single sensitivity indices can be combined to one index per domain. This would create a view that first shows inter-domain correlations. The user can then interactively browse through the in a top down way to reach the values of interest. Filtering the data can also help reducing the data. Assuming that within one domain the behavior of the model is well understood, we could filter out all partial derivatives inside one domain and only focus on the derivatives of parameters from different domains.

Another limitation of the current implementation is that the solver needs an equation system which has one unique solution. Feedbacks in the design, where one parameter is used as input and output at the same time are prohibited. Currently the user has to manually break such design cycles. In Concurrent Engineering this is typically done by neglecting the feedback or by applying a margin. This can also be seen in our propulsion tank example, because the mass of the propellant also contributes to the total mass of the system, which is used in the rocket equation to calculate the mass of the propellant. In our example we neglect this feedback because the mass of the propellant is small compared to the total mass of the system. Designs with feedback cycles are common and usually they are the input for many optimization problems. The limitation on the solvability of the equations system is not specific to the sensitivity analysis. It just means that if there is a design cycle the model cannot be calculated and in result the sensitivity analysis cannot be conducted, because it needs the solver to calculate the perturbations of the solution of the underlying problem.

As another means of analysis, we are thinking of extending the Virtual Satellite software with an implementation of a design structure matrix. Usually this kind of analysis is identifying activities and tasks which are dependent on each other. This information can be used to guide the study of course. However, in the design structure matrix community, there is also the method of a so-called parameter-based design structure matrix. This method looks a physical design parameters and the resulting view is therefore similar to the matrix view of the sensitivity values. Comparing these two methods seems to be an interesting task for the future.

REFERENCES

[1] V. Schaus et al., “Concurrent Engineering Software Development at German Aerospace Center – Status and Outlook –“, *4th International Workshop on System & Concurrent Engineering for Space Applications (SECESA)*, October 2010

[2] M. Deshmukh, V. Schaus, P. M. Fischer, D. Quantius, V. Maiwald, A. Gerndt, “Decision Support Tool for Concurrent Engineering in Space Mission Design,” *19th ISPE International Conference on Concurrent Engineering*, Trier, Germany, September 2012

[3] S. Rudolph, M. Bölling, “Constraint-based conceptual design and automated sensitivity analysis for airship concept studies”, *Aerospace Science and Technology*, vol. 8, pp. 333 – 345, 2004

[4] A. Saltelli, S. Tarantola, F. Campolongo, “Sensitivity Analysis as an Ingredient of Modeling”, *Statistical Science*, vol. 15, no. 4, pp. 377/395, 2000

[5] T. R. Browning, “Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions”, *IEEE Transactions on Engineering Management*, vol. 48, no. 3, August 2001

[6] European Cooperation for Space Standardization (ECSS), *Space Engineering – Engineering design model exchange (CDF)*, doc. ref. ECSS-E-TM-E-10-25A, October 2010

[7] P. Fischer, V. Schaus and A. Gerndt, “Design Model Data Exchange Between Concurrent Engineering Facilities by Means of Model Transformation”, *13th NASA-ESA Workshop on Product Data Exchange*, Cypress, California, USA, 2011

[8] G. A. Freeze, M. Reeves, “A deterministic sensitivity analysis method to aid conceptual model development”, *Calibration and Reliability in Groundwater Modeling, Proceedings of the ModelCARE 96 Conference*, Golden. Colorado, USA, September 1996

[9] G. Knappstein, *Statik, insbesondere Schnittprinzip*, 3rd ed., Deutsch Harri Gmbh, 2007

[10] W. Ley, K. Wittmann, W. Hallmann, *Handbook of space technology*, Aerospace Series, Wiley, 2009

[11] Datasheet of the titan alloy Ti6Al4V: http://valbruna.de/dokumente/datenblaetter/datenblatt_gr5.pdf, last accessed: 13/04/2012

[12] Datasheet of Astrium’s EPDM – bladder tank BT 01/0: http://cs.astrium.eads.net/sp/brochures/propellant-tanks/BT%200-01.pdf, last accessed: 13/04/2012