11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

MULTIDISCIPLINARY ANALYSIS OF THE DLR SPACELINER CONCEPT BY DIFFERENT OPTIMIZATION TECHNIQUES

A. Tröltzsch[1], M. Siggel[1], A. Kopp[2], T. Schwanekamp[2],

[1] German Aerospace Center (DLR), Simulation and Software Technology, Cologne, Germany,
anke.troeltzsch@dlr.de, martin.siggel@dlr.de,
[2] German Aerospace Center (DLR), Institute of Space Systems, Bremen, Germany,
alexander.kopp@dlr.de, tobias.schwanekamp@dlr.de

Key words: Multidisciplinary Analysis and Optimization, Coupled Problems, SpaceLiner, Derivative-Free Optimization.

Abstract. In engineering design, people are more and more interested in analyzing and optimizing systems of components and not only single components. We implement a multidisciplinary design optimization framework to analyze the DLR SpaceLiner design concept involving different engineering disciplines. Moreover, we present a comparison of different optimization tools regarding their ability to find a multidisciplinary feasible solution where the optimizer SOLVOPT shows to perform best on the given problem.

1 Introduction

The DLR SpaceLiner [1], depicted in Figure 1, is a concept study between aviation travel and space travel for ultra fast passenger transport. A flight from Europe to Australia would take only 90 minutes with the SpaceLiner. It combines features of conventional aircraft with systems only found in spacecraft, e.g. rocket engines and thermal protection systems.

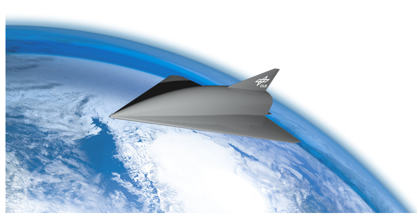


Figure 1: DLR SpaceLiner (illustration)

As one can imagine, many challenges have to be faced when designing new space transportation vehicles. In particular the induced heat during atmospheric reentry at high velocities creates new problems which do not exists in conventional aviation. So far, research has been carried out on several aspects of the SpaceLiner concept separately, including aerodynamics and the resulting heat loads, structure analysis, and cooling concepts for the reentry phase.

Here, we want to address the complete system analysis and shape optimization of the SpaceLiner, where all these different aspects have to be combined. Since each aspect is computed by a different simulation tool, these programs have to be coupled. The optimization of such scenarios is typically performed using multidisciplinary optimization (MDO) techniques. In this paper, we want to set up an integrated simulation environment for the multidisciplinary analysis and optimization of the whole system involving different disciplines. We are going to compare three different publicly available optimization codes to optimize the total space craft mass, regarding the convergence rate and in particular the quality and feasibility of the final multidisciplinary solution.

In Section 2 of this paper, the different involved disciplines together with the simulation tools are described. In Section 3, we introduce our multidisciplinary problem and which approach is applied to solve it. Numerical experiments are presented in Section 4, followed by some concluding remarks and the outlook in Section 5.

2 Involved Disciplines

During the concept design phase, complex numerical analysis methods such as CFD or FEM are too time consuming and thus unsuitable to assess the large variety of vehicle modifications. Instead, approximate engineering tools characterized by reasonable computation times and an acceptable accuracy, are the most appropriate choice. Several adequate programs are available at the DLR Space Launcher Systems Analysis group (DLR-SART) and are described below.

Due to the reasonable computation times, these programs are efficient tools for multidisciplinary spacecraft predesign. The involved disciplines are introduced in this section.

2.1 Shape parametrization and geometry generation with GGH

To be able to optimize the outer shape of the spacecraft, the shape needs to be parametrized to maintain a manageable number of design variables. Obviously, the outer configuration shape impacts many aspects of the spacecraft, including aerodynamic performance and stability, the inner spacecraft structure and thus it's structural mass.

To simulate these different aspects of the spacecraft, the vehicle surface geometry is represented by a block-wise build up structured quadrilateral panel mesh. This mesh is created by our fast and efficient program *GGH* (Grid Generator for HOTSOSE) that was implemented by DLR-SART. Even if the mesh created by GGH is simple compared to a complex CAD shape design, it is definitely appropriate enough to serve as a structural geometry input for the mass model as well as for our inner aircraft structure computation tool. An example mesh made by GGH is given in Figure 2.

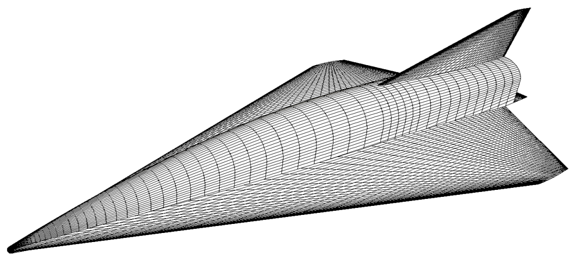


Figure 2: Example mesh of the SpaceLiner created by GGH

2.2 Aero- and aerothermodynamics with HOTSOSE

The aero-/thermodynamic performance of the configuration is mainly affected by the outer spacecraft shape and its center of gravity (CoG). The vehicle must be capable of trimming along the whole trajectory and should provide a sufficiently large glide ratio, in particular during the hypersonic descent flight.

As the main part of the descent is flown within the hypersonic regime, aerothermal aspects become a critical design factor. For hypersonic Mach numbers the surface inclination method *HOTSOSE* (HOT Second Order Shock Expansion method) was implemented at DLR to estimate the total aerodynamic coefficients as well as surface parameters and heat loads [6, 7]. The local air pressure distribution is evaluated from various surface inclination methods, depending on the geometry of the corresponding surface section. HOTSOSE also allows to approximate the influence of viscous effects either for ideal gas assumption or in case of thermodynamic equilibrium flow. The corresponding parameters such as wall temperature, heat transfer and skin friction coefficients are calculated by established engineering methods. More details about these methods and their uses can be found in [6, 7].

Even if some fundamental aerodynamic aspects such as shock-boundary layer interactions or interference drag are neglected by HOTSOSE, this method is well proven for the preliminary aircraft design phase and suitable for a variety of vehicle shapes in hypersonic flow conditions [6].

2.3 Thermal protection system and water cooling with CalCoolAid

The thermal protection system (TPS) must protect the structure, the internal systems and the passengers against the external heat loads during the descent. Due to its own mass, any TPS strongly influences the total mass of the spacecraft. This has to be considered during optimization.

We use our tool *CalCoolAid* for the preliminary estimation of the required cooling water mass. It makes the simplified assumption that the required cooling water mass flow only depends on the heat load and the specific vaporization enthalpy of water. HOTSOSE delivers the heat flux distribution over the outer spacecraft shape. For the simulation, HOTSOSE is executed twice by CalCoolAid. In the first call the vehicle surface heat fluxes and temperatures are calculated under the assumption of radiation adiabatic equilibrium. This simulation provides the surface parameters, which would occur without active cooling. CalCoolAid automatically selects the vehicle surface regions, which are above a critical temperature T_{max}. These are the critical regions, where active cooling is necessary to avoid structural damage and which therefore are relevant for the water mass estimation. However, if these regions are actively cooled down to a target temperature (T_t), the assumption of radiation adiabatic equilibrium is not valid anymore there. Therefore HOTSOSE is executed a second time under the assumption of an isothermal wall at T_t within the critical regions. The calculated heat flux is then integrated along the surface in the critical regions to achieve the total heat flux for a certain flight point. Within the optimization loop, CalCoolAid is executed subsequently for every flight point of the trajectory and the heat fluxes are then integrated over the time to get the total integrated heat load of the full flight trajectory. Dividing this heat load by the vaporization enthalpy of water then provides the total required water mass.

2.4 Mass model with STSM

The evaluation of a very detailed and accurate mass model requires profound structural analyses as well as advanced dimensioning of systems and subsystems, aspects usually not applicable during the preliminary design process. Therefore the approximation tool *STSM* (Space Transportation Systems Mass) was developed at SART for zero and first level investigation of single and multiple stage configurations. It supports the early evaluation process and conveniently delivers data required at the pre-design phase, calculated via empirical correlations which reduce the amount of necessary input to a minimum. A compilation of the empirical mass estimation methods of zero level analysis is given in [9, 10, 11, 12]. The most powerful capability of STSM is the calculation of CoG movement along the trajectory history. This is performed for the complete vehicle as well as for each stage by superposition of the CoG of all single components and systems, which are either calculated also by empirical methods or provided by the user, if known. When entering first level analysis the results can be refined by including more elaborate information from additional tools.

2.5 Structural sizing with HySAP

HySAP is an ANSYS based structural analysis program developed by the DLR-SART. Its main task is to perform rapid parametric structural analysis on a preliminary design level for almost arbitrary vehicle configurations, with comparatively low modelling and calculation times.

HySAP combines preprocessor and sizing routines with the ANSYS Mechanical APDL environment. The Fortran based tool HySAP serves as a preprocessor to create an APDL input file for ANSYS. This file contains the commands for geometry generation, loads application, meshing, solution and post-processing as well as iteration step and load case information. The structural sizing is performed by a separate self-developed sizing tool, which is called by ANSYS after determining a solution. The following data are delivered by ANSYS to the sizing tool: calculated stresses for the complete vehicle structure, geometry and structural design of the individual structural members, wall thicknesses, and material data and masses. The sizer validates the structure against several strength and stability failure modes and adapts wall thicknesses as necessary. ANSYS then restarts the modeling and computation process with the adapted wall thicknesses. This procedure is repeated several times and for different load cases until convergence of the structural mass has been reached. More detailed program descriptions and results of application cases have been published in [3] and [4], and will be complemented by more recent developments in [5].

HySAP is connected to the other system analysis tools already mentioned in order to receive input data from the particular disciplines. Aerodynamic pressure distributions as well as the surface mesh are provided by HOTSOSE and GGH respectively. Aeroelasticity is not considered in the program loop since the deflections of wings and fuselages for hypersonic vehicle concepts are usually low. Thus, no back-coupling from the structure to the aerodynamics discipline is implemented.

In ANSYS, the complete vehicle geometry is modeled with multi-layer shell elements, whereas different stiffening concepts are available. An arbitrary number of load cases can be processed successively by HySAP that include aerodynamic pressures, tank static and hydrostatic pressures, accelerations that yield inertia loads, and user-defined point loads or moments. The structure is divided in optimization components. Each of these components will be sized individually and assigned a uniform wall thickness. The design criteria for the structural sizing are various buckling and stability failure modes as well as von Mises stresses. The buckling and stability sizing is done via standard handbook methods. No FE buckling analysis is done.

3 The multidisciplinary optimization problem

So far, research has been carried out on the mentioned disciplines separately. The parameters of each major discipline have been optimized on its own with only little consideration of the other disciplines.

In this paper, the goal is to find the optimal preliminary SpaceLiner design considering all the mentioned disciplines together in one single optimization problem. As the considered disciplines together build a coupled system (see Figure 3), multidisciplinary optimization techniques have to be applied.

In our case, the system is coupled with one back coupling. When the inner spacecraft structure is optimized the CoG is modified as a result. This CoG shift influences the

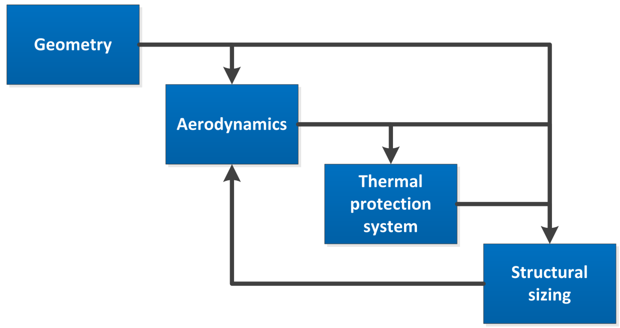


Figure 3: The coupled system of the disciplines for SpaceLiner flight analysis and optimization

aerodynamics of the SpaceLiner on one hand. On the other hand, the structure tool gets aerodynamic pressure distributions from the aerodynamics tool which influences in turn the structure computation and thus also the resulting CoG again. More formally speaking, values which are output to one discipline and input to another discipline are called coupling variables.

3.1 The problem setting

The trajectory is fix and the configuration of the SpaceLiner is optimized for one flight point during the gliding flight at a height of $h = 46\,km$ at an approximate speed of $M = 19.8$ as this point represents the aerodynamics during a major part of the trajectory.

The design variables of the optimization problem are shape parameters of the fuselage and the wing. The length of the fuselage is separated into a nose, a center and a tail part. The nose radius and the vertical shift of the nose are also varied. As parts of the wing, the sweep angle, the chord length of the root and the tip of the wing are adjusted. Furthermore, the angle of attack is varied for the given flight point.

The objective of the formulated optimization problem is to locate the design with the minimal mass including liquids subject to lower bounds on glide ratio and lift coefficient, an upper bound on the pitching moment and to several geometrical constraints which restrict the total length of the vehicle and the trailing edge position of the wing.

3.2 The multidisciplinary approach

Several approaches have been developed to solve multidisciplinary optimization problems. Among others, Simultaneous Analysis and Design (SAND) [13], Individual Design Feasible (IDF) [14], Multidisciplinary Feasible (MDF) [14], Concurrent Subspace Optimization (CSSO) [15] and Collaborative Optimization (CO) [16] are well-known methods.

We decided to use the “sequential Individual Design Feasible” (sIDF) approach which is a version of IDF. In IDF, the coupling variables are added to the set of design variables to decouple the discipline analyses so that they no longer rely on each other for their coupling variable input. To ensure a multidisciplinary feasible solution at the optimum, one additional feasibility constraint is added to the optimization problem for each coupling variable. These constraints ensure that at the optimum, the estimate of the coupling variables matches the actual coupling variables computed by each discipline. With f denoting the objective function and c denoting the constraint functions, IDF can be stated as,

$$\begin{aligned} \min \quad & f(z, x, y^t) \\ \text{w.r.t.} \quad & z, x, y^t \\ \text{s.t.} \quad & c(z, x, y(x, y^t, z)) \leq 0 \\ & y_i^t - y_i(x, y_j^t, z) = 0, \end{aligned} \tag{1}$$

where y^t represents the coupling variables estimates (or targets) provided by the optimizer, y_i are the coupling variable outputs of discipline i given the estimate of the non-local coupling variables y_j^t from discipline j. Furthermore, x represents the set of local design variables, which are only involved in one discipline and z are the global design variables which are involved in more than one discipline.

This architecture enables the discipline analyses to be performed in parallel, since the coupling between the disciplines is resolved by the coupling variable copies and consistency constraints. The advantage of IDF compared to other approaches is that the IDF problem formulation is very compact and requires minimal modification to existing discipline analyses. Nevertheless, it is not recommended for problems with a large number of coupling variables. The sIDF approach, as depicted in Figure 4, is a version of IDF where

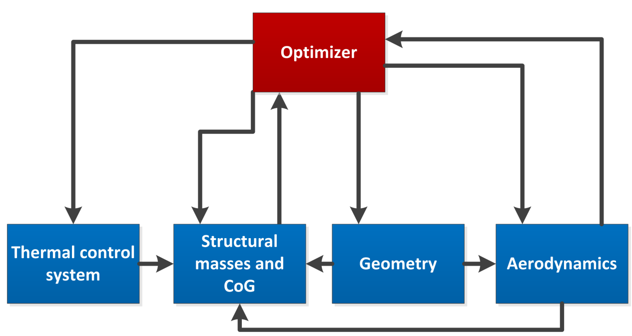


Figure 4: Scheme of the sIDF approach with the involved disciplines

outputs of one discipline which are connected as inputs to only one other discipline are directly tied with the receiving discipline. In this case, coupling through the optimizer is avoided. This technique reduces the number of coupling variables and constraints which would be required in the pure IDF approach. On the other hand, the evaluation of the discipline outputs can not be fully performed in parallel anymore.

4 Numerical experiments

In this section, we briefly describe the framework of our problem implementation and present the applied optimizers together with the results obtained in our experiments.

4.1 The integration environment

To be able to efficiently analyze and possibly optimize the overall system, the described software and simulation tools have been integrated as a process chain inside the Remote Component Environment (RCE) [2]. RCE is an open source distributed workflow-driven integration platform with a graphical user interface developed at the DLR. The implemented process chain of our multidisciplinary problem is depicted in Figure 5. The engineering simulation tools and scripts, described above in Section 2, are integrated as components and displayed each as a box. The boxes are connected with arrows to visualize the data flow between the tools. In the "Optimizer" component, different optimization

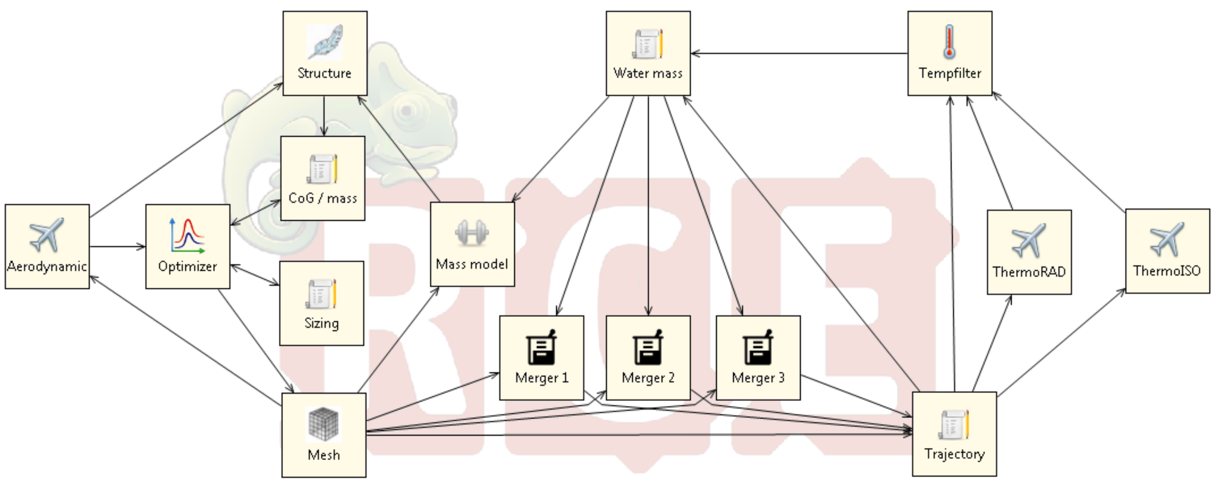


Figure 5: Integrated process chain in Remote Component Environment (RCE)

algorithms can be chosen and the process chain can be started.

4.2 The solvers

Most of the integrated engineering simulation tools do not provide derivatives of the objective and constraint functions such that only optimization methods which do approximate, or which do not need derivatives at all, can be applied in our case. The described optimization problem is solved using several publicly available software codes which we compare in terms of quality of the solution and in terms of number of function evaluations needed for convergence. For our comparison, we consider only local optimization solvers as we expect to find an acceptable multidisciplinary solution in the vicinity of the given SpaceLiner reference configuration which we got from each discipline. Furthermore, global optimization solvers are generally interested to explore the whole design space what is known to require an exhaustive number of function evaluations. This is prohibitive in engineering design optimization. Therefore, our solvers to test are COBYLA [17] and APPS [21] from the DAKOTA [18] optimization suite and SOLVOPT [20] from the Python Optimization Package pyOpt [19].

COBYLA and APPS implement derivative-free optimization algorithms, thus they do not need derivatives at all. COBYLA implements a sequential trust-region algorithm that employs linear approximations to the objective and constraint functions, where the approximations are formed by linear interpolation at $n+1$ points in the space of the variables and tries to maintain a regular-shaped simplex over iterations. APPS is a pattern search method which generally walk through the search domain according to a defined stencil of search directions. It applies a fully asynchronous technique in that the search along each offset direction continues without waiting for searches along other directions to finish.

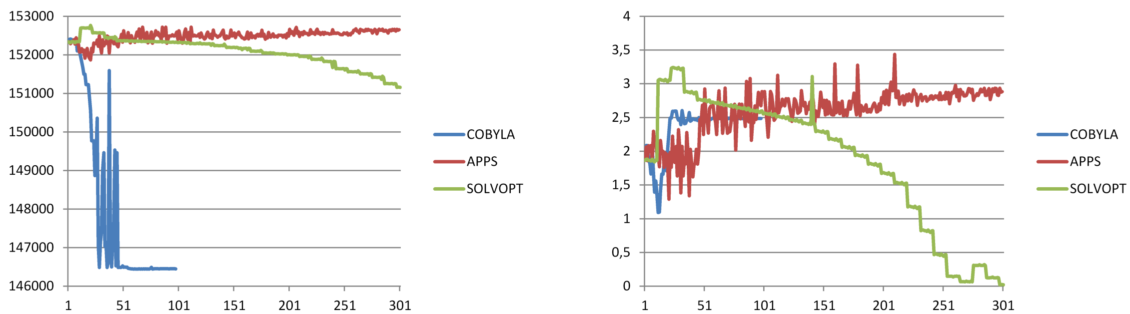
SOLVOPT is a gradient-based method where the gradients are approximated by finite differences due to the lack of analytical gradients in our context. SOLVOPT is a modified version of Shor's r-algorithm with space dilation to find a local minimum of nonlinear and nonsmooth problems. The algorithm handles constraints using an exact penalization method.

4.3 Results

Finally, we would like to know which of the described solvers is able to solve our multidisciplinary optimization problem best. As we terminate the optimization process due to limited time resources after at most 300 function evaluations, each optimizer produces another intermediate solution of the problem. This is quite natural as the computation of the search step and the handling of the constraints is different for each optimizer. But on the other hand, it can also show whether or not some solver is suitable to be applied in such a framework.

The numerical results can be seen as histories in Figures 6(a) and 6(b) where all evaluated function values are displayed. Each of the number of function evaluations, depicted on the lower axis, include one evaluation of the objective function and one evaluation of each constraint function since these values are computed in the same run of the implemented simulation environment. We present our results in terms of the objective function (mass of the space craft in kg) on the left and in terms of a cumulated constraint violation function $\psi(x_k)$ on the right. The values of $\psi(x_k)$ are computed a posteriori for each new configuration x_k suggested by the respective optimizer during the optimization process as follows

$$\psi(x_k) = \sum_{i=1}^{m} \phi(c_i(x_k))$$



(a) Objective function (b) Cumulated constraint violation

Figure 6: Comparison of COBYLA, APPS and SOLVOPT

with

$$\phi(c_i(x_k)) = \begin{cases} 0, & \text{if} \quad c_i(x_k) < 0, \\ c_i(x_k) & \text{if} \quad c_i(x_k) \geq 0. \end{cases}$$

Please note that the values of this constraint violation function are computed to be able to display a fair comparison in Figure 6(b), but do not necessarily coincide with the values, the different algorithms are guided through their individual optimization runs.

At first glance, one could assume from the objective function reduction on the left figure that COBYLA is the best optimizer out of the three tested. It shows a very fast descent of the objective function and converges early. But on the right figure, one can see that the initial configuration with the given design variables and the initial coupling variable estimate for CoG is not feasible and that COBYLA does not satisfy these constraints at termination. Instead, we see that SOLVOPT is able to find a feasible solution inside the given time frame and that it is at the same time able to reduce the space craft mass for more than 1000 kg. The optimal solution has a significantly smaller nose radius what causes higher temperatures and more water is needed for cooling this region but it has positive effects on the aerodynamic glide ratio. The optimizer APPS is not able to provide a feasible solution nor a reduced objective in the given time frame.

5 Conclusions and outlook

We presented the multidisciplinary analysis and optimization problem of the DLR SpaceLiner design concept involving the disciplines geometry, aerodynamics, thermal protection and structure. The goal was to find a feasible and possibly optimized solution of the whole system starting from the best solution of each single discipline. For this reason, we implemented the given simulation tools in the integration framework RCE using the sIDF approach.

In our experiments, we applied the optimization codes COBYLA, SOLVOPT and APPS which implement a trust-region method, a line-search method and a direct pattern-search method, respectively, to the posed multidisciplinary optimization problem. Our numerical experiments have shown that the optimization algorithm COBYLA has a clear advantage in terms of convergence speed over the other tested solvers but it is not able to find a feasible solution and stops prematurely. At the end, SOLVOPT has shown the best performance out of the three tested codes. It was successful to find a feasible and optimized solution for our problem in the given time frame. Whether this advantageous behavior comes from the line-search approach, the constraint handling by an exact penalization strategy, the application of a gradient-based aproach or due to other reasons will be the topic of further investigations.

Concerning the engineering simulation tools of the involved disciplines, several modifications and enhancement are planned in the near future. E.g., a panel code derived from the NASA program PanAir is planned to be connected to HySAP as well, in order to provide low speed pressure distributions.

Furthermore, to enhance the presented multidisciplinary analysis framework, we envisage to integrate a tool for a more reliable analysis of the passive thermal protection system of the SpaceLiner and a tool for the exact estimation of tank masses.

REFERENCES

[1] Sippel M. Introducing the SpaceLiner vision. In *7th International Symposium on Launcher Technologies, Barcelona, Spain*, 2007.

[2] Seider D., Fischer P. M., Litz M., Schreiber A. and Gerndt A. Open source software framework for applications in aeronautics and space. In *IEEE Aerospace Conference, Mountain View, MT, USA*, 2012.

[3] Kopp A. Parametric Studies for the Structural Pre-Design of Hypersonic Aerospace Vehicles. In *12th European Conference on Spacecraft Structures, Materials and Environmental Testing, Noordwijk, The Netherlands*, 2012.

[4] Kopp A., Garbers N., Jarlas R., Rabia H. Parametric Structural Analysis for the SpaceLiner. In *18th AIAA/3AF International Space Planes and Hypersonic Systems and Technologies Conference, Tours, France*, 2012.

[5] Kopp A., Garbers N. Structural and TPS Trade-Off Studies and Design Options for the Hypersonic Transport System SpaceLiner. In *European Conference on Spacecraft Structures, Materials and Environmental Testing, Braunschweig, Germany*, 2014.

[6] Reisch U., Anseaum Y. Validation of the Approximate Calculation Procedure HOTSOSE for Aerodynamic and Thermal Loads in Hypersonic Flow with Existing Experimental and Numerical Results. In *DLR-Forschungsbericht, 98-23*, 1998.

[7] Streit T., Martin S., Eggers T. Approximate Heat Transfer Methods for Hypersonic Flow in Comparison with Results Provided by Numerical Navier-Stokes Solution. In *DLR-Forschungsbericht, 94-36*, 1994.

[8] Schwanekamp T. Beschreibung des Netzgenerators GGH. *DLR internal report, SART TN-008/2011, Bremen, Germany*, 2011.

[9] Wolf D.M. PEMDIT Ein Computerprogramm zur Massenabschtzung und geometrischen Auslegung von Transportsystemen, Teil 1: Massenmodelle. *DLR internal report 30106-89/1*, 1989.

[10] Merkl A. Mass Model for TSTO Vehicle Concepts. *FSS-MAN-MM-3212-001, Issue 1*, 1995.

[11] Hirsch G. N. EDIN0613P Weight Estimating Program. *NASA CR-151898*, 1976.

[12] Glatt C.R. WAATS A Computer Program for Weights Analysis of Advanced Transportation Systems. *NASA CR-2420*, 1974.

[13] Haftka R. T. Simultaneous Analysis and Design. *AIAA Journal, Vol. 23, No. 7, pp. 10991103*, 1985.

[14] Cramer E. J., Dennis Jr J. E., Frank P. D., Lewis R. M., and Shubin G. R. Problem Formulation for Multidisciplinary Optimization. *SIAM Journal on Optimization, Vol. 4, No. 4, pp. 754776*, 1994.

[15] Sobieszczanski-Sobieski J. Optimization by Decomposition: a Step from Hierarchic to Non-Hierarchic Systems. *Tech. Rep. September, NASA Langley Research Center, Hampton, VA*, 1988.

[16] Braun R. D. Collaborative Optimization: An Architecture for Large-Scale Distributed Design. *Ph.D. thesis, Stanford University, Stanford, CA 94305*, 1996.

[17] Powell M.J.D. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis, Kluwer Academic, Dordrecht, , pp 5167*, 1994.

[18] Adams B.M., Bauman L.E., Bohnhoff W.J., Dalbey K.R., Ebeida M.S., Eddy J.P., Eldred M.S., Hough P.D., Hu K.T., Jakeman J.D., Swiler L.P., and Vigil D.M. DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 5.4 User's Manual. *Sandia Technical Report SAND2010-2183*, December 2009. Updated April 2013.

[19] Perez R.E., Jansen P.W., and Martins J.R.R.A. pyOpt: A Python-Based Object-Oriented Framework for Nonlinear Constrained Optimization, *Structures and Multidisciplinary Optimization, 45(1):101-118*, 2012.

[20] Kuntsevich A., Kappel F. SolvOpt manual: The solver for local nonlinear optimization problems. *Tech. rep., Institute for Mathematics, Karl Franzens University of Graz, Autria*, 1997.

[21] Gray G.A. and Kolda T.G. Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization. *ACM Transactions on Mathematical Software, 32(3):485507*, 2006.