# MADO - SOFTWARE PACKAGE FOR HIGH ORDER MULTIDISCIPLINARY AIRCRAFT DESIGN AND OPTIMIZATION

**Tomasz Goetzendorf-Grabowski* , Jacek Mieloszyk* , Dawid Mieszalski****
***Warsaw University of Technology, Warsaw, Poland **MSP, Drogomyśl, Poland**

**Keywords:** *optimization, aircraft design, software*

## Abstract

*Paper presents an overview of MADO, the Multidisciplinary Aircraft Design and Optimization software package. The package is prepared to link existing and modified tools to utilize them next, in an efficient way for optimization. The tools used in the first stage of development of MADO were presented. Examples of numerical computations are shown - airfoil and the new designed UAVs.*

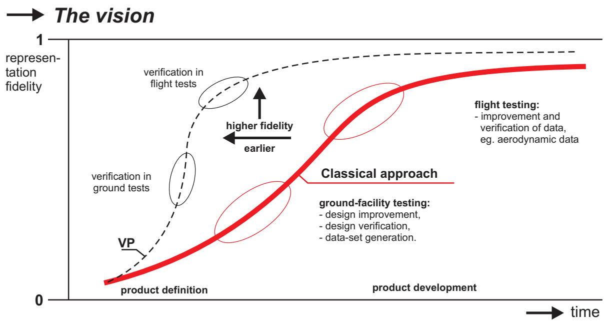


**Fig. 1** The idea of cutting time and costs by SimSAC [3]

## 1 Introduction

Design of new aircraft is always a challenge. Today's market demands highly efficient planes, fulfilling challenging missions and often contradicting expectations. Thus every project, especially as complex as new design of aircraft, is a set of compromises. Without meeting the goals from many scientific disciplines the aircraft simply won't fly. If the design requirements are very demanding it is often hard to propose sufficient project not using numerical optimization techniques [1, 2]. More information on the early stage of the project helps to reduce costs and work time. This is done by improving the process of making confident decisions, which was considered in the SimSAC[3] project (Fig.1).

One of the main achievements of SimSAC project was development of CEASIOM[4] package, that links many different tools, from geometry definition, trough aerodynamic computation, mass and inertia analysis, to stability and control issues. This system had to decrease time necessary to obtain wide knowledge about new designed aircraft. MADO goes a step beyond CEASIOM[5]. Collecting experience of SimSAC project authors present proposition to link current tools into global numerical optimization loop.

## 2 Main idea of MADO

Engineers have to deal with increasing amount of data, which forces the designers to focus on repeatable activities, not on creative work they are paid for. Most of these activities can be automated or semi automated depending on the complexity of the task and experience of the designer. Very often the disciplines (e.g. aerodynamics, stability and control, strength analysis, etc.) exist separately. The lack of interfaces between the tools, or only one way connections, cause that translating data from one solver to another consumes most of the work time. Thus even simplest optimization is not in use. Thanks to software environment integration it is possible to improve the design process. The key concept lies in connection of main software analyzers (e.g. stability analyzer, aerodynamic solver, etc.) into one structure (Fig.2). The designer can make such connections from one program to other manually, however it still could consume much time. Depending on skills and experience user can add scripts controlling data flow. The solution is developing easy to use, so-called wrappers, that link tools used in design process into globally managed system. Easy way of data transferring relieves the designer and will allow to increase his creativeness. Many starting points are possible to realize the design task, either from CAD geometry to numerical analysis, or in an opposite way. Defining the complicated nonlinear geometry of an aircraft can also be done by author's innovative interface by setting only few design parameters. This method can be treated as fast, high level geometry definition. The package is thought to extend capabilities of existing software by adding additional components - commercial or free, used in the actual institution and linking them into one suite, which can be used for optimization.

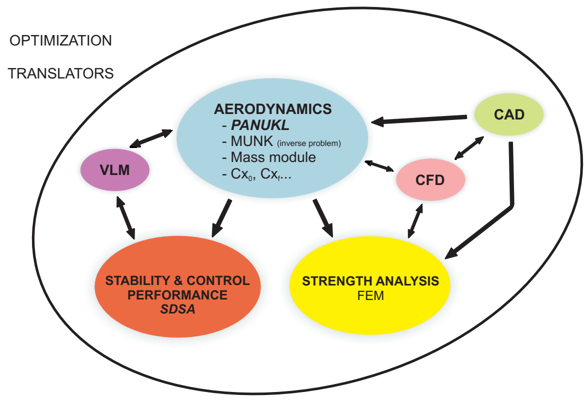


**Fig. 2** Topology of software components and connections

## 2.1 Current development state of MADO

The collected software is capable of performing the aerodynamic, structural analyzes basing on CAD geometry if desired. The package contains the component for static and dynamic stability analysis, which also allows to perform flight simulation and estimate performance [6]. In this way we can take into account many project details. All the unique features of presented package, which provide automation, allow to define optimization task in a relatively easy way. Gradient [7, 8], Monte Carlo [9], Genetic [10] and Swarming [11] optimization algorithms are currently available in the suite. Designer works with user friendly environment, spending less time on repeatable activities, without sacrificing flexibility and complexity of the task he defines. Over the time spent working with the software user can increase complexity by adding next scientific disciplines and increasing number of design variables. This procedure will lead to better optimized solutions.

## 2.2 Main components of MADO

Current version of MADO contains several components, that are the standalone applications. However, they can also run in batch mode, which is very useful to automate optimization process. In the beginning the freeware (or ourself written) applications were taken into account, due to possibility of making necessary changes to run iteration without any prompting. Some commercial applications, which were available for authors were also initially tested with success (VSAERO, VLAERO by AMI Inc.).

### *2.2.1 XFOIL*

XFOIL [12] is a well known program for the design and analysis of subsonic isolated airfoils, developed by Mark Drela. Possibility to use XFOIL in batch mode allowed for easy integration of the software in the optimization loop.

### *2.2.2 PANUKL*

PANUKL is a package for aerodynamic analysis of an aircraft using low order panel method. The package was born in mid nineties and was used many times to compute aerodynamic characteristics of an aircraft including stability derivatives

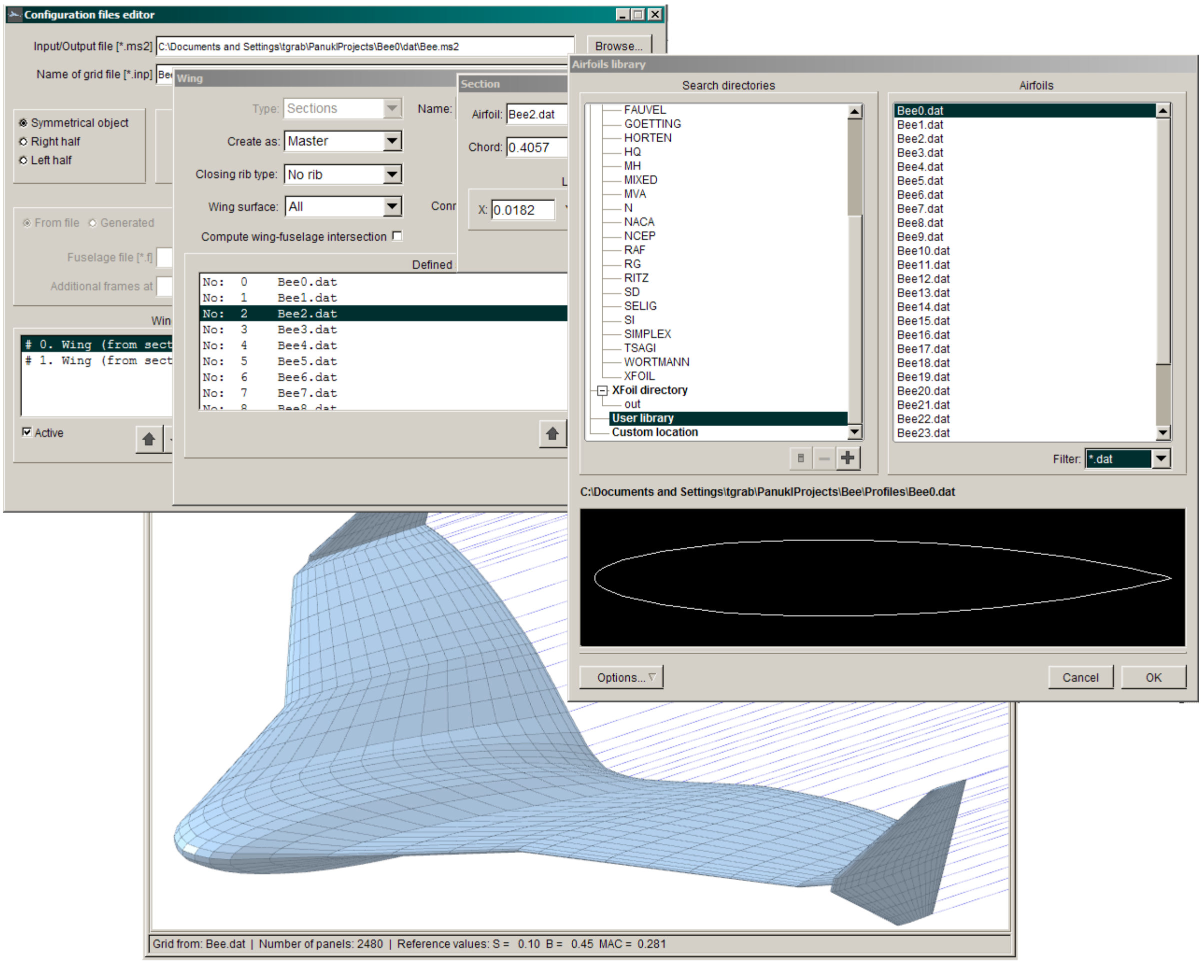


**Fig. 3** PANUKL package

[13]. The package (Fig.3) is being developed all the time, reaching it's maturity. Current version [14] contains preprocessor to define geometry in an easy way using minimum necessary parameters, vortex wake generator, solver and post-processor to view and analyze the computation results. Although pure potential methods are not able to compute all components of drag, PANUKL can estimate induced drag using Trefftz method. The components of PANUKL package can be run in batch mode, what is useful to automate computations in optimization loop.

### *2.2.3 SDSA - Simulation and Dynamic Stability Analyzer*

SDSA module was developed as the CEASIOM[5] module however it can be run as a standalone application as well. It was developed for S&C analysis and is able to compute stability characteristics using linear and nonlinear model (simulation model) as well [15].

SDSA uses the same Six DoF mathematical nonlinear model of the aircraft motion for all functions. For the eigenvalue analysis, the model is linearized numerically around the equilibrium (trim) point. Eigenvalues and eigenvectors analysis allow automatic recognition of the typical modes of motion and their parameters. The flight simulation module can be used to perform test flights and record flight parameters in real-time. The recorded data can be used for identification of the typical modes of motions and their parameters (period, damping coefficient, phase shift). The stability analysis results can be assessed basing on CS/FAR, ICAO, and MIL requirements.

As a module of MADO, it can receive all the necessary data (aerodynamics, mass, inertia),

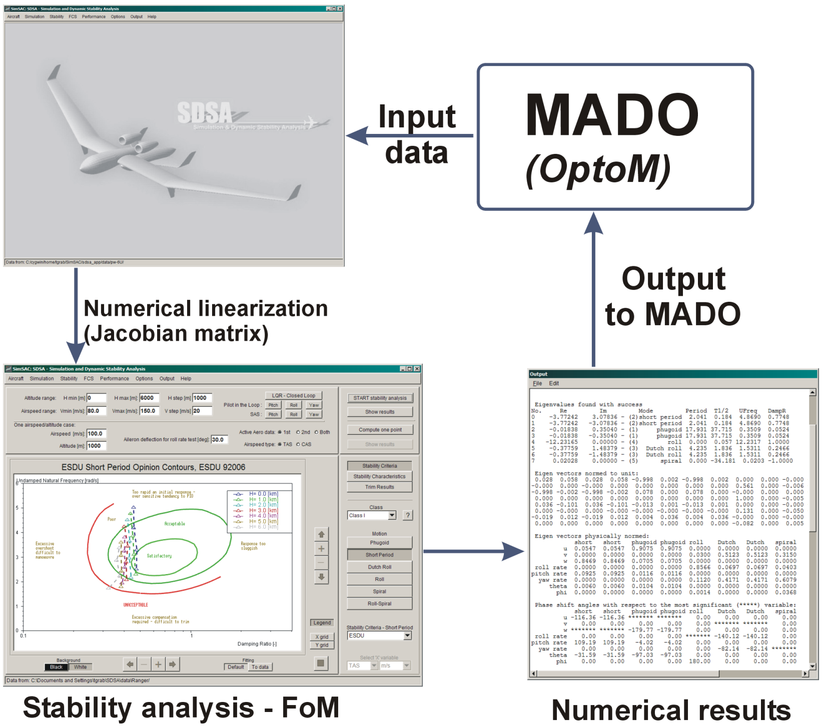


**Fig. 4** SDSA and its functionality in optimization loop

when available, without special prompting. The necessary data can be delivered to SDSA as an XML file or as a set of plain text files. The second option is useful e.g. for experimental data. The data set contains aerodynamic coefficients or/and stability derivatives tables, mass and inertia data, propulsion data, control derivatives and reference dimensions. The control and propulsion data can be completed and edited using special options of SDSA. SDSA accepts aerodynamic data as tables of stability derivatives as function of angle of attack and Mach number. SDSA also accepts a multidimensional array of force and moment coefficients versus six state parameters (angle of attack, Mach number, sideslip angle and rotational velocity components). A similar array is defined for control derivatives and stability derivatives versus selected accelerations (i.e. alpha dot derivatives). All aerodynamic data (derivatives) can be reviewed and are checked by comparison with typical values.

SDSA, running in batch mode, can deliver necessary output data for optimization procedure without any prompting and the optimization process managed by OptoM can run completely in an automatic way.

### *2.2.4 OptoM*

OptoM (Fig.5) is meant to be easy to use tool for all purpose optimization. Core of the application has incorporated many different optimization algorithms, with possibility of adjusting their settings in graphical user interface. Graphical interface also provides some settings for output format, input and output files paths, flags to use constrains, optimization parameters check, fast update of configuration files, error messages and so on. One of the very useful features is "Tester" option for single objective function analyze, which helps to test and debug optimization task defined by the user. Currently available family of optimization algorithms are: Gradient [7, 8], Monte Carlo [9], Genetic [10] and Swarming [11] methods. All the optimization algorithms can act in significantly different manner depending on the settings. User of the application has to know only the basics of the optimization algorithms. The main work for him is to properly define the optimization task in a dynamically linked library,

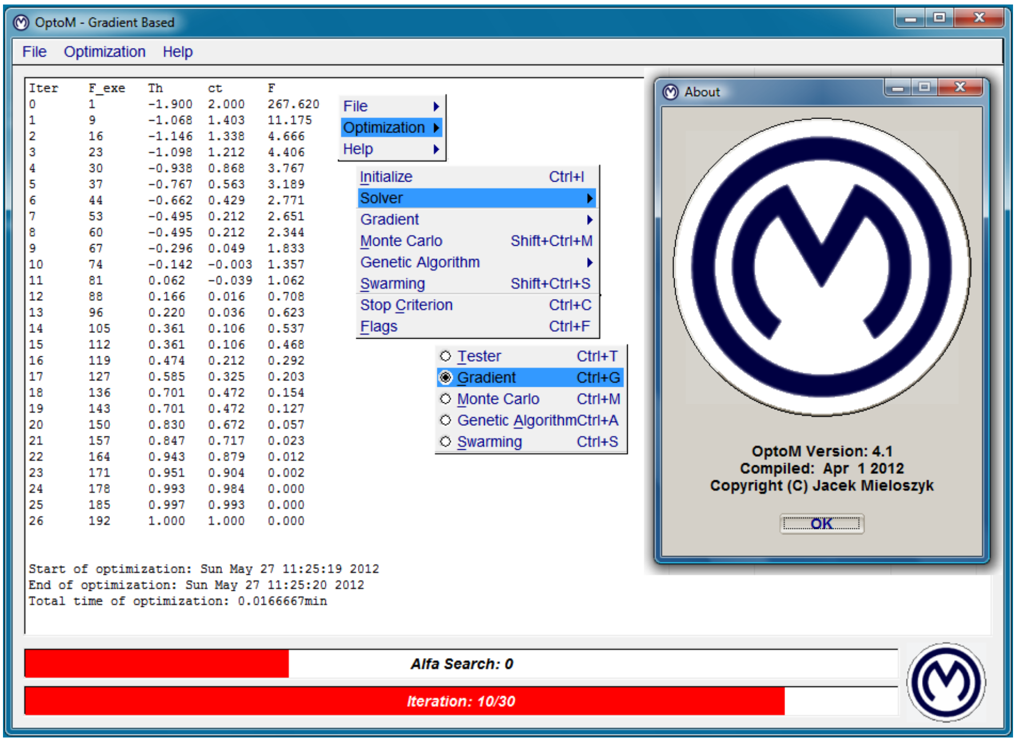


**Fig. 5** OptoM screen shot

which is described in more detail in the next chapter. User is also equipped with tools to help him in the definition of the optimization task, like: definition of quadratic penalty function, matrix operations, functions calling external analysis software and many more.

### *2.2.5 Dynamic linked libraries*

Optimization software is part which connects all the MADO tools in one comprehensive aircraft analyze. OptoM has flexible structure which is shown on Fig.6. User has majority of optimization algorithms he can choose from, which are incorporated in the main part of the OptoM program. All settings of the optimization algorithms are done in OptoM GUI. Change from one optimization method to another is done by single switch. Providing that the user has previously adjusted settings of the particular optimization algorithm, or agrees for the default settings, optimization process can start immediately with the new algorithm. Optimization task is defined in dynamic library which is linked with the OptoM. Such solution allows designer to focus on definition of his problem to solve without need of deep understanding of how the optimization algorithms work. Only basic knowledge about optimization is needed to adjust the settings if necessary. Ability to define directly optimization problem in the dynamic library provides flexibility and unbounded possibilities for the user. Optimization task can be entirely defined in the dynamic library, but nothing stands against using external analysis software. Any analysis software, which can take input parameters, scripts and commands can be incorporated Fig.6. Only minor difficulty is to provide appropriate input for the analysis software and read in results from the analysis during optimization.

## 3 Numerical examples

This chapter shows two examples of numerical optimization, which utilizes MADO concept. First example concerns of optimization of classical airfoil. The second example concerns MAV optimization.

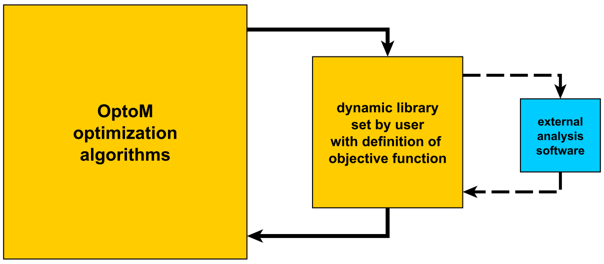


**Fig. 6** OptoM structure

## 3.1 Airfoil

In this example two dimensional airfoil is optimized. Objective function was maximization of lift coefficient for prescribed starting geometry and defined flow conditions. The optimized airfoil was NACA 23012, which had fixed angle of attack equal to 3 deg. The airfoil aerodynamic analyze was done by Xfoil code. It was assumed that the airfoil is optimized for small UAV and Reynolds number was set to Re=200000 and Mach number Ma=0. Design variables were: maximum thickness of the airfoil, maximum camber of the airfoil, position of the maximum thickness and position of the maximum camber. Changes of the geometry shape were easily made by using Xfoil tools for airfoil geometry modification. Optimization software always minimizes objective function, to maximize airfoil lift coefficient the objective function was mathematically defined as in equation (1).

$$F_{OBJ} = 1/C_L \tag{1}$$

For the optimization Monte Carlo optimization algorithm was used. The algorithm was set for ten iterations with fifty objective function analysis during every iteration. No additional constrains were defined. Convergence of the objective function (maximized lift coefficient) shows Fig.7. History of changing design parameters is shown on Fig.8. Initial and optimized geometry is shown on Fig.9.

After ten iterations airfoil lift coefficient for fixed angle of attack improved by $dC_L \sim = 0.1$. In the case of the considered airfoil most influence on the objective function had parameters of maximum thickness and position of maximum camber and moderate influence of position of maximum thickness. Definition of the objective function and commands flow to make analysis with Xfoil software took less than one hundred lines of C++ code written in the OptoM shared library. Time of optimization was about fifteen minutes.

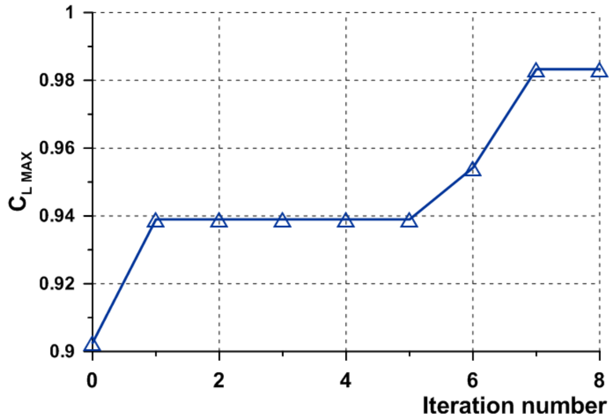


**Fig. 7** Airfoil objective function - maximization of lift coefficient

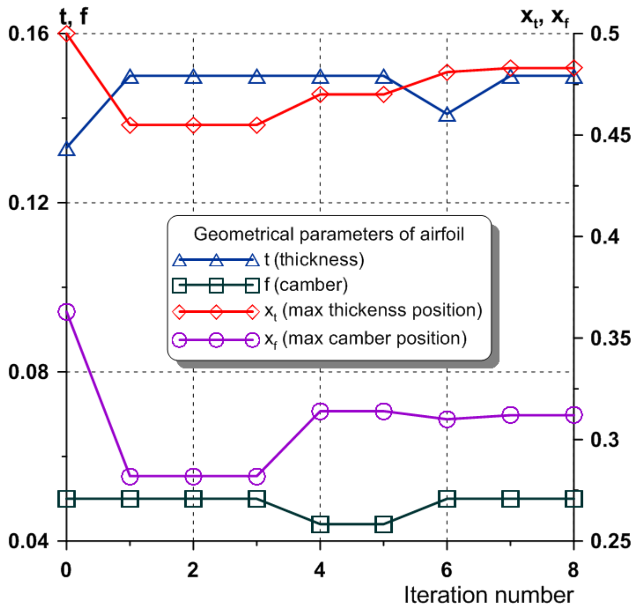


**Fig. 8** History of changing design parameters

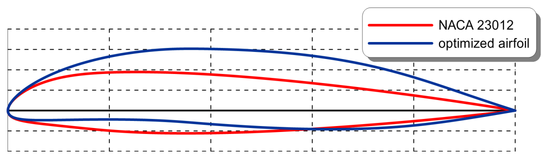


**Fig. 9** Airfoil before and after optimization

### 3.2 Bee - mini UAV

In the second example minimization of the aerodynamic drag was optimized for constant lift coefficient, with additional constrains put on equilibrium of forces in the aerodynamic z axis and predefined static stability margin. Constrains were realized by quadratic penalty function. Achieved values of penalties from the crossed constrains were very big so the objective function was scaled by a factor of one hundred, which is defined by equation (2).

$$F_{objective} = 100C_D + P_1 + P_2 \tag{2}$$

where:

$$P_1 = 0.5C_1^2/\mu$$
$$C_1 = mg - 0.5\rho V^2 SC_L$$
$$P_2 = 0.5C_2^2/\mu$$
$$C_2 = -0.1 - dC_m/dC_L$$

Analysis were done utilizing non-viscous panel code PANUKL[14]. Design variables were: angle of attack, length of the tip chord, three parameters, which controlled nonlinear wing twist distribution defined as a fourth order polynomial center of gravity and four variables defining wing tip cut and filet in the middle part of the wing Fig.10.

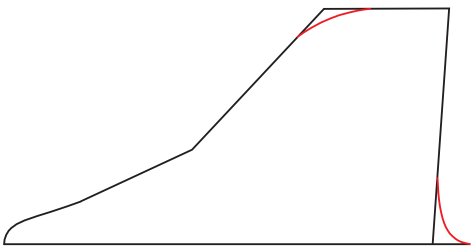

**Fig. 10** Bee wing planform

The first optimization results showed that the optimization task was over constrained. Solution satisfied stability constrains by varying angle of attack, but the geometry parameters didn't change significantly maintaining the old geometry. After this experience variable of position of center of gravity was added. This way geometry could vary still satisfying the constrains.

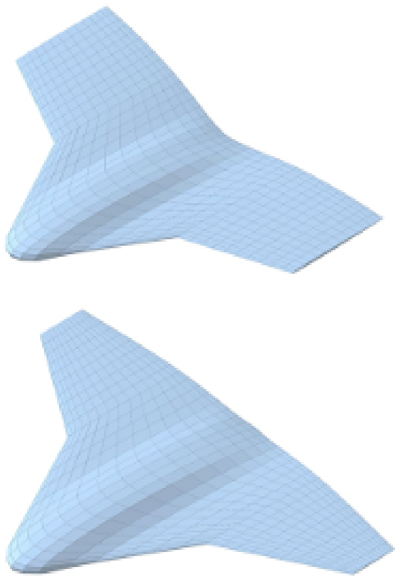

**Fig. 11** Example of competitive optimization solutions

For optimization genetic and gradient algorithms were used, both giving corresponding results. Genetic algorithm was very robust and always gave solutions. Contrary to that experience gradient algorithm needed quite some time to be set to start the solution to converge, but after it was done it converged much faster than genetic algorithm. The best type of gradient solver was second order Newton method. Although this algorithm needed second order derivatives, its estimation of search direction was so fine, that its efficiency overtook Steepest Descent, Conjugate Gradient and Quasi Newton gradient methods. Interestingly genetic algorithm, which can theoretically lead to random solutions, showed that two competitive solutions are possible, with the current optimization task definition, with completely different geometry (Fig.11).

The obtained solution, from the first optimization tests, revealed a question: are the configurations dynamically stable in all modes?

The basic tests of dynamic stability were performed and satisfying results were obtained for the first configuration, with the aft wing sweep (Fig.12). In the next step SDSA is going to run in

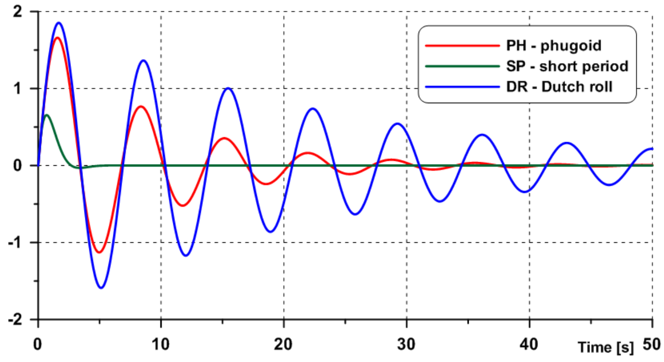


**Fig. 12** Basic modes of motion for analyzed UAV

the optimization loop to satisfy the stability criteria "on-line". In the final version authors forecast, that SDSA will be able to modify the decision variables or be a part of constraints system.

For the final optimization, which included more design parameters and geometry details, Swarming optimization algorithm was used. This algorithm was set for ten iterations with fifty objective function analysis during every iteration. History of converging objective function is shown on Fig.13. It shows how values of the objective function for the best and the worst individual in the swarm are getting closer. It can be seen, that already in the third iteration objective function, indicating best individual in swarm, merely changes. Performance of the optimization algorithm is very good, what can be observed on the Fig.13 with logarithmic objective function axis.

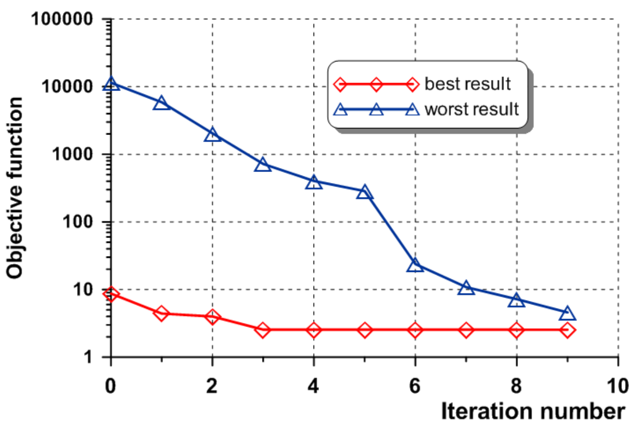


**Fig. 13** Objective function versus iteration number

On Fig.14 penalty functions values are shown. After the third iteration they are close to zero, which means that both constrains are satisfied with great accuracy.

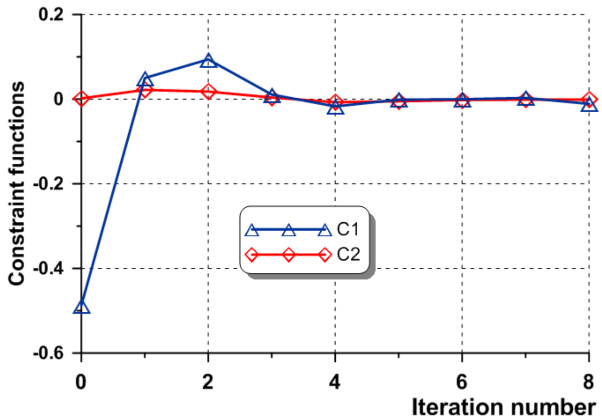


**Fig. 14** Penalty functions values versus iteration number

Final geometry with very smooth pressure distribution after optimization shows Fig.15. In the case of the MAV shared library was quite big, but most of the code defined highly nonlinear geometry of the flying object. Pure definition of the objective function and commands flow to execute external aerodynamic analysis took less than one hundred fifty lines of code. This example shows that defining optimization task takes less time, but ability to define the task in the shared dynamic library gives endless flexibility and allows for completion of very demanding tasks.

## 4 Concluded remarks and further steps

The experience collected so far in the first examples is very promising. Most of the current MADO tools have user friendly graphical interface with novel functionality, which makes it possible to define complicated problems for analyze with few variables. Great concern is put on easy data exchange between MADO tools, which allows to concentrate on the very design. Although software maintenance becomes very easy, experienced users will still be able to define complicated analyze and optimization tasks without limitations. MADO is meant to be so flexible, that its working capabilities will grow with the knowledge of the user, no matter on what stage

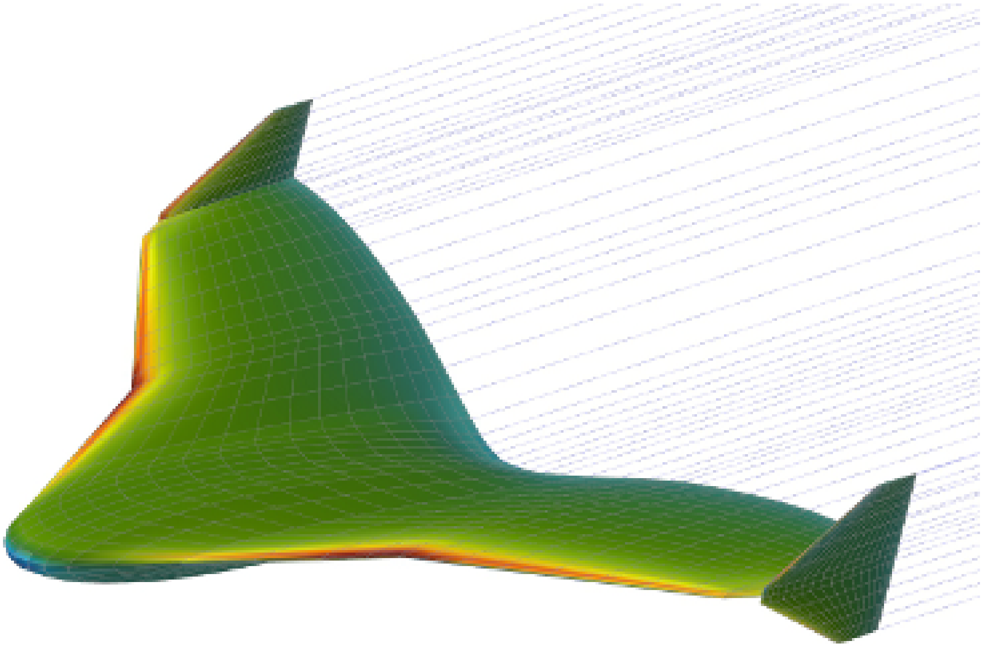

**Fig. 15** Pressure distribution for the final configuration

of the "know how" he is. In the optimization software any desired optimization task can be realized. The optimization task can be defined entirely in the shared library, or incorporate any external analysis software available with batch mode capabilities. Length of the shared library code, with analysis, optimization flow control, objective function, constrains and so on, in most cases will not even exceed 200 lines. Compiling only the shared library, containing only the optimization task and user equipped with additional features to help him define the task enables for very effective work. Future of the MADO package seems to be bright. The software is constantly developed and improved to make it even more computationally powerful, intuitive and efficient to work with.

## References


[1] Mieloszyk J., "Handling optimization problems on an example of micro UAV", 3rd CEAS Air&Space Conference, Venice 2011

[2] Stalewski W., Żółtak J., "Multi-objective and Multidisciplinary Optimization of Wing for Small Aircraft", 3rd CEAS Air&Space Conference, Venice 2011

[3] SimSAC - "Simulating Aircraft Stability And Control Characteristics for Use in Conceptual Design", Technical Annex to Contract No. FP6-030838, July 17, 2006

[4] von Kaenel R., Rizzi A., Oppelstrup J., Goetzendorf-Grabowski T., Ghoreyshi M., Cavagna L., and Berard A. CEASIOM: Simulating Stability & Control with CFD/CSM in Aircraft Conceptual Design, Paper 061, 26th Int'l Congress of the Aeronautical Sciences, Anchorage, Alaska, Sept 2008.

[5] CEASIOM - Computerised Environment for Aircraft Synthesis and Integrated Optimisation Methods, http://www.ceasiom.com/

[6] SDSA - Simulation and Dynamic Stability analysis, Software package, Warsaw University of Technology, http://www.meil.pw.edu.pl/add/ADD/Teaching/Software/SDSA

[7] Nocedal J., Wright S.J. "Numerical Optimization" Springer, Series in Operations Research, 1999

[8] Vanderplaats G.N.: Numerical Optimization Techniques For Engineering Design, McGraw Hill, 1984

[9] Dickman B.H. and Gilman M.J., Monte Carlo optimization, Journal of Optimization Theory and Applications, Volume 60, Number 1 (1989), pp 149-157

[10] Raymer D.P. "Enhancing Aircraft Conceptual Design Using Multidisciplinary Optimization" Doctoral Thesis, 2002

[11] Li W., Huyse L. and Padula S., Robust airfoil optimization to achieve drag reduction over a range of Mach numbers, Structural and Multidisciplinary Optimization, Volume 24, Number 1 (2002), pp 38-50, Springer-Varlag

[12] XFOIL - program for the design and analysis of subsonic isolated airfoils, http://web.mit.edu/drela/Public/web/xfoil/

[13] Goetzendorf-Grabowski T., Goraj Z., Calculations of stability derivatives of an aircraft using panel methods, Transactions of the Institute of Aviation, 2/1996 (145), Warsaw (in Polish)

[14] PANUKL potential solver, Software package, Warsaw University of Technology, http://www.meil.pw.edu.pl/add/ADD/Teaching/Software/PANUKL

[15] Goetzendorf-Grabowski T., Mieszalski D., Marcinkiewicz E., Stability analysis using SDSA tool, Progress in Aerospace Sciences, Volume 47, Issue 8, November 2011, Pages 636-646


## Copyright Statement