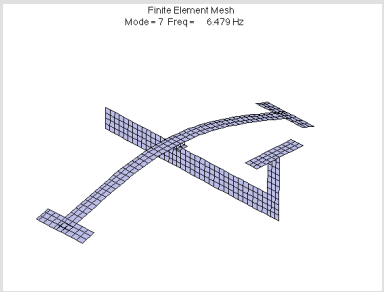
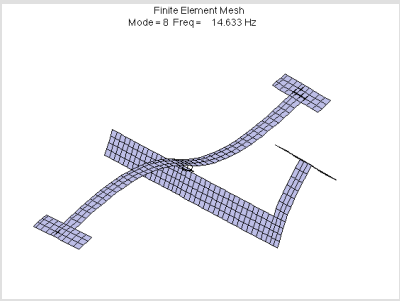
Automatic differentiation of a structural analysis code (OpenNastran)
Application to the aeroelastic optimization of a flight demonstrator

Master level (6 months from MARCH 2017)

Advisor Joan Mas Colomer, PhD ONERA/ISAE-SUPAERO Prof Joseph Morlier, ISAE-SUPAERO

Context : In mathematics and computer algebra, automatic differentiation (AD) [1], [2], also called algorithmic differentiation or computational differentiation, is a set of techniques to numerically evaluate the derivative of a function specified by a computer program. AD exploits the fact that every computer program, no matter how complicated, executes a sequence of elementary arithmetic operations (addition, subtraction, multiplication, division, etc.) and elementary functions (exp, log, sin, cos, etc.). By applying the chain rule repeatedly to these operations, derivatives of arbitrary order can be computed automatically, accurately to working precision, and using at most a small constant factor more arithmetic operations than the original program.





Within the context of structural optimization or multidisciplinary optimization problems that involve the use of a structural code, it is often necessary to compute the gradients of the objective function or the constraints with respect to the design variables whenever gradient-based optimization methods are used. In the case of multidisciplinary optimization, it is often required to execute different computational codes, which makes it makes it not possible to obtain the gradient of these functions by differentiating a single code only. In this particular case, automatic differentiation may be used to obtain certain partial derivatives for the computation of the coupled gradient.
The recent publication by NASA of the source code of a NASTRAN version [6], one of the most used structural analysis codes among the aerospace industry and research fields, represents an opportunity of application of automatic differentiation to structural optimization. There exist several ways of applying automatic differentiation: source code transformation and operator overloading. The first form, source code transformation (method used by TAPENADE [3] and OpenAD [4], for example), consists in replacing the source code for a function by an automatically generated source code that includes statements for calculating the derivatives interleaved with the original instructions.

The internship will be paid (monthly bonus: € 550) and carried out in the laboratories of the ISAE-SUPAERO in the joint research unit with ONERA (ECR MDO). Background:

- Computer sciences – Applied Mathematics

References

[1] "Automatic Differentiation." Wikipedia. https://en.wikipedia.org/wiki/Automatic_differentiation.
[2] Martins, Joaquim R. R. A., and John T. Hwang. "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models." AIAA Journal 51, no. 11 (2013): 2582-599. doi:10.2514/1.j052184.
[3] Hascoet, Laurent, and Valérie Pascual. "The Tapenade Automatic Differentiation Tool." ACM Transactions on Mathematical Software 39, no. 3 (2013): 1-43. doi:10.1145/2450153.2450158.
[4] Utke, J. "OpenAD : Algorithm Implementation User Guide." 2004. doi:10.2172/834715.
[5] Martins, Joaquim R. R. A., Peter Sturdza, and Juan J. Alonso. "The Complex-step Derivative Approximation." ACM Transactions on Mathematical Software 29, no. 3 (2003): 245-62. doi:10.1145/838250.838251.
[6] Nasa. "NASTRAN-95." GitHub. 2015. https://github.com/nasa/NASTRAN-95.