

Forward Automatic Differentiation with an application to solid mechanics

Automatic Differentiation (AD) is an ensemble of techniques that allows for the numerical evaluation of the derivatives of a function, expressed in a computer programming language, with the same accuracy of the function itself. AD techniques rely on the assumption that mathematical expressions are always decomposed into a sequence of elementary sub-expressions by computers. Therefore, if the analytical derivatives of the elementary operators and functions are known, it is possible to evaluate the derivatives of complex expressions, with respect to the given independent variables, by operating on the partial results. As a result, AD differs from finite differences, because it does not approximate the continuous derivatives with discrete differences, thus it does not suffer from truncation error. Also, in comparison with symbolic differentiation, AD does not suffer from expression swelling, due to the limitations of the simplification algorithms, which induces both on memory and CPU costs.

This seminar will focus on forward automatic differentiation, a flavor of AD that can be easily implemented in the programming languages that allow user defined data type and the overloading of existing operators and functions to work with the user introduced types. Together with the theoretical basis of AD, a specific implementation in the Julia programming language will be presented, along with the application of AD techniques to the solution of a selection of solid mechanics problems.

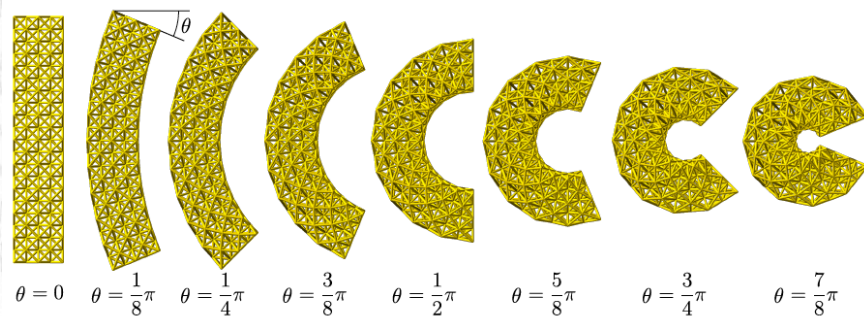


Figure 1: Deformation of a lattice structure under large displacements. The rotation on the top and the bottom planes have equal amplitude and opposite sign. Deformed shapes for different rotation angles.

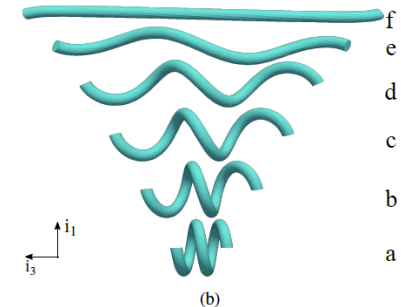
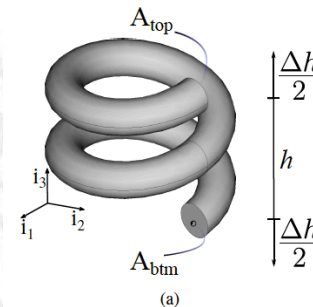


Figure 2: Deformation of a continuum under finite material and geometrical nonlinearities. (a) Domain's geometry, A_{top} and A_{btm} , are the top and the bottom end cross sections of the helix, respectively. (c) Deformed configurations at different stages of the simulation.

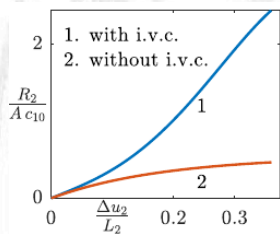
Dott. Andrea Vigliotti
Innovative Materials Laboratory
Italian Aerospace Research Centre, Capua (Italy)

January 20th, 14:30 (sharp)
Aula MS1, DICAr
Via Ferrata, 3 – Pavia

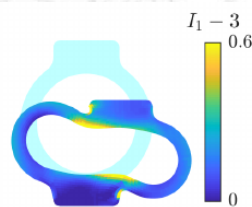
Implementing a Forward Automatic Differentiation system for Solid Mechanics in Julia*

This tutorial will illustrate the structure of a general finite element program for the solution of solid mechanics problems based on AD in Julia. The tutorial will be hands-on and will cover the following points:

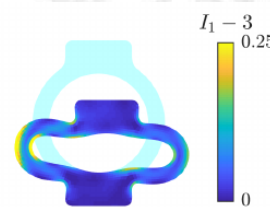
- Brief introduction to Julia, download and installation of the core language
- The Julia type system and operator overloading
- A simple implementation of a second order forward AD system
- The structure of a general purpose Finite Element program implemented through AD: elements, materials, solver
- Examples covering structural elements, 2D and 3D continuum domain, non-trivial internal constraints



(a)



(b)



(c)

Axi-symmetric problem, simulation results. (a) Normalized force displacement plot, Line 1 is the response with internal volume constraint (i.v.c.), line 2 is the response without i.v.c. . (b) Deformed cross section with i.v.c., and (c) without i.v.c. at maximum deformation.

* From Wikipedia: *Julia is a high-level, high-performance, dynamic programming language. While it is a general purpose language and can be used to write any application, many of its features are well-suited for high-performance numerical analysis and computational science.* More about Julia at <https://julialang.org/>

Dott. Andrea Vigliotti
Innovative Materials Laboratory
Italian Aerospace Research Centre, Capua (Italy)

January 20th, 15:15 (sharp)
Aula MS1, DICAr
Via Ferrata, 3 – Pavia