Proceedings of the 24th UK Conference of the
Association for Computational Mechanics in Engineering
31 March - 01 April 2016, Cardiff University, Cardiff

Evaluation of the Tangent Stiffness Matrix for Hyperelastic Fibres using Automatic Differentiation

***Euan Muir[1], Chris Pearce[1] and Lukasz Kaczmarczyk[1]**

[1] School of Engineering, University of Glasgow, G12 8QQ

*e.muir.1@research.gla.ac.uk

ABSTRACT

The numerical modelling of fibres undergoing finite deformations requires the implementation of a constitutive law and the solution of the governing PDE requires an accurate linearisation of the nonlinear response. In order to ensure a computationally efficient implementation and to simplify the implementation, automatic differentiation has been utilised.

Key Words: *finite element method; automatic differentiation; hyperelasticity; fibres*

1. Introduction

Fibres are present in many materials and serve to enhance the material performance in particular directions. In natural materials, this directional dependency is a result of optimisation whereby the increased strength and stiffness is only where it is needed. Synthetic materials such as fibre reinforced polymers (FRP) have been designed to deliberately exploit this benefit. This paper focuses on fibres in soft tissue but also has broader applicability. The implementation of a nonlinear material response in the finite element method requires linearisation of the constitutive relationship . This paper will describe this process for a hyperelastic fibre model subject to finite deformations using automatic differentiation.

2. Tangent Stiffness Matrix

The nonlinear system of equations are expressed simply as:

$$\mathbf{r}(x) = \mathbf{f}_{\text{ext}}(\mathbf{x}) - \mathbf{f}_{\text{int}}(\mathbf{x}) = \mathbf{0} \tag{1}$$

This is solved using a Newton-Raphson scheme. Equation 1 is expressed as a truncated Taylor series expansion, whereby the residual $\mathbf{r}$ at the next iteration is expressed as:

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \frac{\partial \mathbf{r}_i}{\partial \mathbf{x}} d\mathbf{x} \tag{2}$$

where

$$\frac{\partial \mathbf{r}_i}{\partial \mathbf{x}} = -\frac{\partial \mathbf{f}_{\text{int}}(\mathbf{x})}{\partial \mathbf{x}} = -\mathbf{K}_t = -\int_V \boldsymbol{B}^T \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}} \mathbf{B} \, dV \tag{3}$$

The derivative $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}$ is the elasticity tensor $\mathbb{C}$, the computation of which is non-trivial. In this paper we demonstrate how to compute it using automatic differentiation.

3. Constitutive Models

Soft tissues are composed of an extracelluar matrix (ECM) with collagen fibres. The ECM is represented as a neo-Hookean material [1] and the fibres from the Eberlein model [2], both being hyperelastic materials. The two material models are combined in Equation 4. Equation 5 and Equation 7 are the strain energy functions (Ψ) for the neo-Hookean and fibre materials respectively and their corresponding Second-Piola Kirchoff stress ($\boldsymbol{S}$) are in Equation 6 and Equation 9.

$$\Psi = \Psi_n + \Psi_f \tag{4}$$

$$\Psi_n = \frac{\mu}{2}(I_C - 3) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2 \tag{5}$$

where $\boldsymbol{F}$ is the gradient of deformation, J is the volumetric change, $\boldsymbol{C}$ is the right Cauchy-Green deformation tensor, $J = \det(\boldsymbol{F})$, $I_C = \operatorname{tr} \boldsymbol{C}$, and λ and μ are the Lamé coeficients.

$$\boldsymbol{S}_n = \mu(\boldsymbol{I} - \boldsymbol{C}^{-1}) + \lambda(\ln J)\boldsymbol{C}^{-1} \tag{6}$$

$$\Psi_f = \sum_{\alpha=1}^{2} \frac{k_1}{2k_2}\{[\exp[k_2(\bar{I}_\alpha^* - 1)^2] - 1]\} \tag{7}$$

$$\bar{I}_\alpha^* = \bar{\boldsymbol{C}} : \boldsymbol{A}_\alpha, \quad \alpha = 1, 2, \qquad \bar{\boldsymbol{C}} = J^{-2/3}\boldsymbol{C}, \qquad \boldsymbol{A}_\alpha = \boldsymbol{a}_\alpha \otimes \boldsymbol{a}_\alpha, \qquad \mathbb{P} = \boldsymbol{I} - \frac{1}{3}\boldsymbol{C}^{-1} \otimes \boldsymbol{C} \tag{8}$$

$$\boldsymbol{S}_f = J^{-2/3}\mathbb{P} : \hat{\boldsymbol{S}}_f \tag{9}$$

$$\hat{\boldsymbol{S}}_f = \sum_{\alpha=1}^{2} 2k_1\{\exp[k_2(\bar{I}_\alpha^* - 1)^2](\bar{I}_\alpha^* - 1)\boldsymbol{A}_\alpha\} \tag{10}$$

4. Elasticity Tensor

The constitutive equations are expressed in terms of the Second-Piola Kirchoff stress **S**; this can be pushed forward to obtain the First-Piola Kirchoff stress **P** = **FS**. Thus, the elasticity tensor can be calculated from $\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}$. To illustrate that this derivative is not trivial to obtain, the elasticity tensor for the fibre model is shown here: Equation 11 to Equation 13 [2].

$$\mathbb{C}_f = 2\frac{\partial \boldsymbol{S}_F}{\partial \boldsymbol{C}} = \mathbb{P} : \hat{\mathbb{C}}_f : \mathbb{P}^T + \frac{2}{3}\operatorname{tr}(J^{-2/3}\hat{\boldsymbol{S}}_f)\tilde{\mathbb{P}} - \frac{2}{3}(\boldsymbol{C}^{-1} \otimes \hat{\boldsymbol{S}}_f + \hat{\boldsymbol{S}}_f \otimes \boldsymbol{C}^{-1}), \tag{11}$$

$$\hat{\mathbb{C}}_f = 2J^{-4/3}\frac{\partial \hat{\boldsymbol{S}}_f}{\partial \hat{\boldsymbol{C}}} = \sum_{\alpha=1}^{2} \delta_\alpha \boldsymbol{A} \otimes \boldsymbol{A}, \tag{12}$$

$$\delta_\alpha = 4J^{-4/3}k_1[2k_2(\bar{I}_\alpha^* - 1)^2 + 1]\exp[k_2(\bar{I}_\alpha^* - 1)^2], \quad \alpha = 1, 2 \tag{13}$$

Alternatively, automatic differentiation can be used to calculate the elasticity tensor, thereby removing the need for it to be undertaken explicitly. In automatic differentiation the derivatives of a function can be evaluated numerically through the recursive application of the chain rule, exploiting the ability to express the function as basic arithmetic operators and functions.

5. Automatic Differentiation by OverLoading in C++ (ADOL-C)

ADOL-C is a open-source C++ library that applies the principles of automatic differentiation [3]. The independent variables subject to differentiation are defined using a special type $adouble$ and all variables which depend on the independent variable must also be defined using this type. Constants are considered passive and can be defined using standard types such as $double$. The function description needs to be recorded to a tape for the differentiation process and so requires active sections to be set by `trace_on` and `trace_off`. To illustrate the process a piece of pseudo code is shown in Algorithm 1 that calculates the elasticity tensor for the Eberlein fibre model.

Algorithm 1 ADOL-C Fibres Constitutive Model

$double\ k_1, k_2$ ▷ passive variables
$adouble\ F, \bar{\boldsymbol{C}}, J, \boldsymbol{A}_\alpha, \boldsymbol{V}_{f1}, \boldsymbol{V}_{f2}$ ▷ active variables

$double\ x$ = values
$double\ V_{f1i}, V_{f2i}$ = values ▷ local fibres direction

TAPE(ON)
$\boldsymbol{F} <<= x$ ▷ independent variables
$\boldsymbol{V_{f1}} <<= V_{f1i}$ ▷ independent variables
$\boldsymbol{V_{f2}} <<= V_{f2i}$ ▷ independent variables

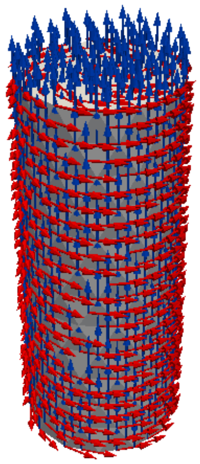
calculate: $\boldsymbol{C}, J, \bar{\boldsymbol{C}}, \boldsymbol{A_\alpha}, \bar{I}_\alpha$
$\boldsymbol{S}$ = fibresFunction($\boldsymbol{C}, J, \bar{\boldsymbol{C}}, \boldsymbol{A_\alpha}, \bar{I}_\alpha$)

$\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}$
TAPE(OFF)

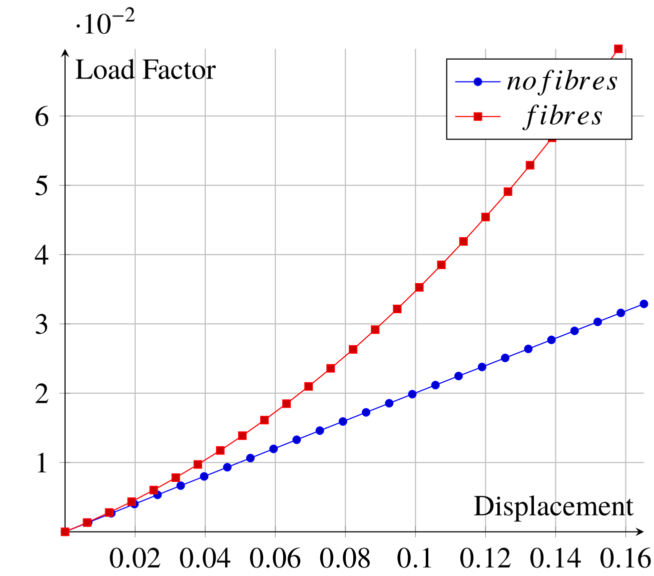
$Jacobian(\boldsymbol{P}, \boldsymbol{F}, [x, V_{f1i}, V_{f2i}])$ ▷ $\dfrac{\partial \boldsymbol{P}}{\partial \boldsymbol{F}}$

6. FE Implementation Example

ADOL-C is integrated into our open source finite element software package MoFEM [4]. To demonstrate ADOL-C working it was used in the calculation of the elasticity tensor for the combined Eberlein Fibre and neo-Hookean material models; the fibre directions applied were not constant and so were considered active variables by ADOL-C. The physical problem was an uni-axially loaded cylinder with fibres wrapped around the circumference and the length as shown in Figure 1a. To demonstrate the influence of the fibres the analysis was repeated with no fibres, also shown in Figure 1b.



(a) Mesh and Fibres



(b) Load Displacement Graph

Figure 1: Uni-axial Load

7. Conclusion

Calculating the elasticity tensor to form the tangent stiffness matrix from the constitutive equations, can be a very involved process. The use of automatic differentiation can remove this step and thereby speeds up the implementation of constitutive models and removes errors.

References

[1] Bonet, J., Wood, R. Nonlinear Continuum Mechanics for Finite Element Analysis. Second. Cambridge Books Online. Cambridge University Press. ISBN: 9780511755446. 2008.

[2] Eberlein, R., Holzapfel, G., Schulze-Bauer, C. An Anisotropic Model for Annulus Tissue and Enhanced Finite Element Analyses of Intact Lumbar Disc Bodies. In: Computer Methods in Biomechanics and Biomedical Engineering 4.3, pp. 209-299, 2001.

[3] Walther, A., Griewank, A: Getting started with ADOL-C. In U. Naumann und O. Schenk, Combinatorial Scientific Computing, Chapman-Hall CRC Computational Science, pp. 181-202, 2012.

[4] MoFEM, Mesh Orientated Finite Element Method http://bitbucket.org/likask/mofem-joseph