

# Sensitivity of solutions computed through the Asymptotic Numerical Method

Isabelle Charpentier

*Laboratoire de physique et mécanique des matériaux, UMR 7554, Île du Saulcy, 57045 Metz, cedex 1, France*

Received 18 March 2008; accepted after revision 1 September 2008

Available online 2 October 2008

Presented by Évariste Sanchez-Palencia

---

## Abstract

The Asymptotic Numerical Method (ANM) allows one to compute solution branches of sufficiently smooth non-linear PDE problems using truncated Taylor expansions. The Diamant approach of the ANM has been proposed for hiding definitively the differentiation aspects to the user. In this Note, this significant improvement in terms of genericity is exploited to compute the sensitivity of ANM solutions with respect to modelling parameters. The differentiation in the parameters is discussed at both the equation and code level to highlight the Automatic Differentiation (AD) purposes. A numerical example proves the interest of such techniques for a generic and efficient implementation of sensitivity computations. **To cite this article: I. Charpentier, C. R. Mecanique 336 (2008).**

© 2008 Académie des sciences. Published by Elsevier Masson SAS. All rights reserved.

## Résumé

**Sensibilité des solutions calculées par la Méthode Asymptotique Numérique.** La Méthode Asymptotique Numérique (MAN) permet de calculer des branches de solutions de problèmes d'EDP suffisamment réguliers à l'aide de séries de Taylor tronquées. L'approche Diamant de la MAN a été proposée pour cacher définitivement les aspects différentiation à l'utilisateur. Dans cette Note, cette amélioration significative en terme de généricité est exploitée pour calculer la sensibilité des solutions MAN par rapport aux paramètres de modélisation. La différentiation en les paramètres est discutée au niveau des équations et du code pour souligner les aspects Différentiation Automatique (DA). Un exemple numérique prouve l'intérêt de ces techniques pour l'implémentation générique et efficace de calculs de sensibilité. **Pour citer cet article : I. Charpentier, C. R. Mecanique 336 (2008).**

© 2008 Académie des sciences. Published by Elsevier Masson SAS. All rights reserved.

*Keywords:* Continuation; ANM; Diamant; Sensitivity analysis

*Mots-clés :* Continuation ; MAN ; Diamant ; Analyse de sensibilité

---

---

*E-mail address:* [isabellecharpentier@hotmail.com](mailto:isabellecharpentier@hotmail.com).

*URL:* <http://www.lpmm.fr/charpentier/Diamant>.

## Version française abrégée

Les problèmes d'EDP non-linéaires réguliers peuvent être résolus à l'aide de séries de Taylor tronquées. La Méthode Asymptotique numérique (MAN) [1–3] introduit ces séries dans le problème discret  $\mathcal{R}(u, \lambda) = 0$ , équation (1), pour construire la séquence de systèmes linéaires (4). À l'ordre  $k$ , le membre de gauche contient les termes relatifs aux  $k^{\text{ièmes}}$  coefficients de Taylor  $u_k$  et  $\lambda_k$ , dont le produit de la matrice linéaire tangente  $K_T$  par  $u_k$ . Le membre de droite  $R_k$  est constitué de formules de dérivation de haut degré desquelles ont été retirés les termes intervenant dans la construction de  $K_T$ . Dans la MAN, étapes de différentiation et résolutions de systèmes linéaires sont réalisées alternativement. Jusqu'à présent, les développements en séries requis par la MAN ont été principalement effectués manuellement.

La Différentiation Automatique (DA) [4] est un ensemble de techniques génériques. L'approche Diamant [5,6] (acronyme de *Différentiation Automatique de la Méthode Asymptotique Numérique Typée*) a été conçue pour cacher définitivement les aspects différentiation de la MAN à l'utilisateur. Dans Diamant, le fait d'initialiser  $u_k$  et  $\lambda_k$  à 0 permet de déduire le second membre  $R_k$  d'une évaluation du  $k^{\text{ième}}$  coefficient de Taylor  $\mathcal{R}_k$  de  $\mathcal{R}$  au moyen des formules de récurrence usuelles. Informatiquement, la librairie Diamant est basée sur la technique dite de « surcharge d'opérateurs » offerte par les langages orienté-objet et qui permet d'attacher les calculs de dérivées d'ordre élevé aux opérations et fonctions intrinsèques du langage de programmation. Une fois validée, cette librairie peut être utilisée avec confiance pour la résolution de problèmes non-linéaires réguliers ou régularisés. Comme montré dans [5–7], l'approche Diamant – générique, réutilisable et simple d'utilisation – préserve les avantages de la MAN en termes d'efficacité et de suivi de courbe. Une revue de problèmes résolus par la MAN est proposée dans [3], ceux-ci sont clairement résolubles par Diamant. Dans cette Note, nous présentons également l'outil de DA par « transformation de source » Tapenade [8] qui permet de générer les instructions de codes linéaires tangents à partir de codes et de directives de différentiation fournis par les utilisateurs. Une description des outils de DA existants est consultable le site Internet [autodiff.org](http://autodiff.org).

La sensibilité aux paramètres de modélisation  $p$  (matériau, régularisation, forme, ...) des branches de solutions obtenues par la MAN est calculée en différentiant en  $p$  la séquence de problèmes linéaires (4). Comme montré dans la Section 3, les systèmes résultants (5) suivent le même patron que précédemment. La matrice  $K_T$  intervient à la fois dans le calcul d'une branche  $(u, \lambda)$  et de sa sensibilité  $(u^d, \lambda^d)$  en  $p$ . Par la suite, les variables différenciées en  $p$  sont signalées par un exposant  $d$ , imitant ainsi le suffixe ajouté par Tapenade aux variables dérivées en mode linéaire tangent. Le membre de gauche contient notamment la série  $R_k^d = \frac{\partial R_k}{\partial p}$  et le terme  $K_T^d u_k = \frac{\partial K_T}{\partial p} u_k$ , tous deux peuvent être obtenus par DA. Nous exploitons la structure particulière de ces équations pour mettre en place une stratégie de DA adaptée à la différentiation du solveur MAN. Dans un premier temps, la différentiation en  $p$  est réalisée en utilisant Tapenade sur le code informatique de  $\mathcal{R}$ . Dans un second temps, Diamant est appliqué au code linéaire tangent généré par Tapenade. Héritant des atouts de Diamant, cette stratégie de DA est aisément applicable à tout code éléments finis discrétisant un problème non-linéaire suffisamment régulier. Elle appliquée sur un problème académique [9] en Section 5. Les sensibilités numériques calculées par le code « Tapenade, puis Diamant » et celles produites par un code MAN écrit à la main puis différencié en  $p$  par Tapenade sont identiques à la précision machine près. A l'exécution, le rapport (égal à 2.8) entre le code « Tapenade, puis Diamant » (avec sensibilité) et le code MAN « Diamant » (sans sensibilité) est inférieur à la borne théorique de 5 classique en différentiation linéaire tangente [10,11] ce qui prouve l'efficacité de notre DA. Un exemple de calcul de sensibilité implémenté à l'aide de la librairie Diamant [5] est librement téléchargeable depuis la page Internet de l'auteur.

## 1. Introduction

The Asymptotic Numerical Method (ANM) [1–3] uses truncated Taylor expansions for the solution of non-linear smooth (or regularised) PDE problems. Their introduction into the PDE problem allows for the construction of a finite sequence of linear systems, one per order of differentiation, depending on the same tangent linear matrix and some higher order differentiated terms. These differentiation stages may be hidden to the user by means of Automatic Differentiation (AD) techniques [4]. In AD, any differentiable discrete model  $\phi$  is viewed as a sequence of elementary operations. The differentiation of  $\phi$  is then performed applying the chain rule to this sequence. Nowadays, powerful automatic differentiation softwares (based on either “source transformation” or “operator overloading”) are available for the differentiation of computer codes written in Fortran, C or Matlab. A review for these tools is presented on

the Internet site `autodiff.org`. The AD-based Diamant approach (as a French acronym for *Différentiation Automatique de la Méthode Asymptotique Numérique Typée*) [5] relies on an operator overloading library devoted to the propagation of Taylor coefficients by means of classical higher-order differentiation formulae. Once validated, this library may be used with confidence on any PDE problem that can be solved through the ANM. As shown in [5–7], the Diamant algorithmic pattern for the series computation is generic, reusable and of easy management. In this Note, we exploit the Diamant abilities to perform sensitivity analysis of the ANM-computed solutions with respect to modelling parameters (material, regularisation, shape, . . .). As the ANM involves high level mathematical operations (linear solver and ANM series computations), this additional differentiation is discussed at both the equation and code level to highlight the AD strategy.

The layout of the Note is as follows. Section 2 briefly presents the ANM, the ensuing linear systems and the implementation with Diamant. The sensitivity of ANM-computed solutions and its AD implementation are described in Section 3. Numerical results presented in Section 4 prove the efficiency, the genericity and flexibility of Diamant for such computations. Some perspectives are proposed as a conclusion.

## 2. Automatic differentiation of the Typed Asymptotic Numerical Method

Let  $A(p)$  be a square  $N \times N$  matrix depending on parameters  $p$ . Let  $v$  be any non-linear analytical function. The problem we consider in the Note is:

$$\mathcal{R}(u(a, p), \lambda(a, p)) = A(p)u(a, p) + \lambda(a, p)(v \circ u)(a, p) = 0 \quad (1)$$

where the unknown vector  $u(a, p) \in \mathbb{R}^N$  and the unknown parameter  $\lambda(a, p) \in \mathbb{R}$  depend on the arc-length parameter  $a$  satisfying:

$$a = \langle u(a, p) - u_0(p), u_1(p) \rangle + (\lambda(a, p) - \lambda_0(p))\lambda_1(p) \quad (2)$$

In the sequel, all the functions are assumed to be analytical in  $a$  and differentiable in  $p$ . Variables

$$u_k(p) = \frac{1}{k!} \frac{\partial^k u}{\partial a^k}(0, p), \quad \lambda_k(p) = \frac{1}{k!} \frac{\partial^k \lambda}{\partial a^k}(0, p) \quad \text{and} \quad V_k(p) = \frac{\partial^k (v \circ u)}{\partial a^k}(0, p)$$

are the Taylor coefficients at order  $k$  of functions  $u(a, p)$ ,  $\lambda(a, p)$  and  $V = (v \circ u)(a, p)$  evaluated at point  $a = 0$ . Truncated Taylor expansions

$$\sum_{k=0}^K a^k u_k(p), \quad \sum_{k=0}^K a^k \lambda_k(p) \quad \text{and} \quad \sum_{k=0}^K a^k V_k(p)$$

are approximations of  $u(a, p)$ ,  $\lambda(a, p)$  and  $V(a, p)$  in the vicinity of  $a = 0$ . Using the Diamant approach [5], the ANM may be discussed in a generic fashion since Taylor coefficients  $V_k(p)$  may be computed automatically as presented below. Introducing these series into (1)–(2) leads to the  $K$  linear systems:

$$A(p)u_k(p) + \sum_{l=0}^k \lambda_{k-l}(p)V_l(p) = 0, \quad \langle u_1(p), u_k(p) \rangle + \lambda_1(p)\lambda_k(p) = \delta_{1k}, \quad k \geq 1 \quad (3)$$

where the Kronecker symbol  $\delta_{1k}$  is equal to 1 when  $k = 1$  and 0 in the other cases. Highlighting the tangent linear matrix  $K_T(p)$  verifying  $K_T(p)w = A(p)w + \lambda_0(p)v_1(p)w$  ( $\forall w$ ) in system (3) yields:

$$K_T(p)u_k(p) + \lambda_k(p)V_0(p) = R_k(p), \quad \langle u_1(p), u_k(p) \rangle + \lambda_1(p)\lambda_k(p) = \delta_{1k}, \quad k \geq 1 \quad (4)$$

In formula (4), right-hand side terms  $R_k(p)$  contain higher-order differentiation formulae that miss terms in  $u_k(p)$  and  $\lambda_k(p)$  already taken into account in the left-hand side terms. As shown in [6], the right hand-side term  $R_k(p)$  is equal to the Taylor coefficient of function  $\mathcal{R}$  at order  $k$  calculated using  $u_k(p)$  and  $\lambda_k(p)$  set to 0 and previously computed Taylor coefficients  $u_l(p)$  and  $\lambda_l(p)$  ( $l = 0, \dots, k-1$ ). With such choices, terms  $R_k(p)$  may be thus obtained applying higher-order AD techniques to  $\mathcal{R}$ . Similar arguments allow for a low-cost construction of the sparse Jacobian  $K_T$  from first order Taylor coefficients of  $\mathcal{R}$  (repeated evaluations in well-chosen directions).

From a computer point of view, the AD Diamant tool is designed to hide the differentiation stages to the user (Diamant library) while being efficient in ANM series computations (Diamant algorithmic pattern). To that end,

Diamant relies on the concept of Operator Overloading (OO), provided by object-oriented programming, as the vehicle for attaching higher order derivative computations to the arithmetic operators and intrinsic functions. The Diamant library contains a type `Creal` devoted to the storage of the Taylor coefficients of a scalar variable in an array of dimension  $K + 1$  and redefines the classical arithmetic operations and intrinsic functions for `Creal` variables. For instance, the higher order differentiation of the multiplication statement  $r = x * y$  between two variables of type `Creal` is implemented using the classical Leibniz formula. The multiplication between a scalar variable and a `Creal` variable has to be defined too. The new type and definitions are gathered in Fortran modules that form the OO library.

In practice, the user “includes” the OO library in its source code by means of appropriate Fortran 90 `USE` statements. He then indicates which variable requires a differentiation changing its type to the `Creal` type. This user-task, not automated in operator overloading AD tools, may be easily tackled since the compiler detects any mismatch of types if the assignment sign `=` is not overloaded. AD is performed at compile time. When the compiler encounters a `*` operation, it takes into account the type of its operands to decide which multiplication (with a higher order differentiation if at least one `Creal` variable is involved) will be performed during the execution of the code. The generic Diamant algorithmic pattern presented in the former Note [5] is an implementation of the ANM solver taking advantages of AD computed derivatives. Once validated, Diamant may be used with confidence for the solution, through the ANM, of any non-linear PDE problem.

As discussed with details in [5–7], Diamant presents the advantages of the ANM while performing the differentiation stage in an automatic fashion. The reader is referred to [3] for a review of non-linear PDE problems that were solved through the ANM. Obviously their solution may be achieved using Diamant too.

### 3. Sensitivity computations with Diamant

We are interested in the computation of the sensitivities  $\frac{\partial u}{\partial p}(0, p)$  and  $\frac{\partial \lambda}{\partial p}(0, p)$  of variables  $u(0, p)$  and  $\lambda(0, p)$  approximated by means of the ANM. Consequently, we approach these two sensitivities as truncated series. We denote by

$$u_k^d(p) = \frac{\partial^{k+1} u}{\partial^k a \partial p}(0, p) \quad \text{and} \quad \lambda_k^d(p) = \frac{\partial^{k+1} \lambda}{\partial^k a \partial p}(0, p)$$

their Taylor coefficients. Hereafter the partial derivative in  $p$  of any variable  $x$  is denoted  $x^d$  to mimic the suffix convention adopted by the AD tool Tapenade [8].

The differentiation of systems (4) with respect to the parameter  $p$  yields  $K$  systems of equations:

$$\begin{cases} K_T(p)u_k^d(p) + \lambda_k^d(p)V_0(p) = R_k^d(p) - K_T^d(p)u_k(p) - \lambda_k V_0^d(p) \\ \langle u_1(p), u_k^d(p) \rangle + \lambda_1(p)\lambda_k^d(p) = -\langle u_1^d(p), u_k(p) \rangle - \lambda_1^d(p)\lambda_k(p), \quad k \geq 1 \end{cases} \quad (5)$$

As  $u_k(p)$  and  $\lambda_k(p)$  are solutions of (3), the unknowns of (5) are  $u_k^d(p)$  and  $\lambda_k^d(p)$  only. Mixed derivatives appearing in the former system may be (i) fully handwritten, (ii) obtained in a semi-automatic fashion applying an AD tool to a previously hand-coded ANM computer program (black-box approach), or (iii) fully automated through a combination of AD tools. Hereafter we discuss the third option noticing that, in AD, peculiar arguments may be used to differentiate codes involving high-level mathematical operations (such as linear solvers) in a more efficient manner. In the present case, some knowledge is introduced in the differentiation taking into account Eqs. (5). The implementation is performed combining the source transformation engine Tapenade [8] with the operator overloading library Diamant.

Given a computer code (assumed to implement a differentiable model), the tangent linear mode of differentiation allows to get a linear code that enables the propagation of one direction of perturbation. This kind of differentiation, generally used for a sensitivity analysis or an evaluation of the Jacobian in a peculiar direction, is proposed by all the existing AD tools. Among the available softwares, we choose Tapenade since it is freely available for research purposes and may be applied to general Fortran codes. Some extension to C codes, under development, is available too. In practice, the user provides the source code, the name of the top routine to be differentiated, a set of “independent” input variables (we differentiate with respect to them) and the mode of differentiation. Given these data, Tapenade generates the statements of a linear code. In our case,<sup>1</sup> the top routine **R** with arguments  $(u, \lambda, p)$  implements equa-

<sup>1</sup> Tapenade may also differentiate a hand-coded ANM program.

tion (1), the parameter  $p$  is the independent variable, and we differentiate in tangent (linear) mode. The resulting source code is  $\mathbf{R}_d(u, \lambda, p, ud, \lambda d, pd)$  (with the Tapenade convention) that evaluates both  $\mathcal{R}(u, \lambda, p)$  and its partial derivative with respect to  $p$  in the direction  $p^d$ .

The tangent linear code  $\mathbf{R}_d$  generated by Tapenade may be differentiated with respect to the arc-length parameter  $a$  by means of the operator loading library Diamant. In that case, a few modifications of Diamant algorithm pattern [5] are needed to perform sensitivity computations of ANM solutions. Assuming  $u_l, u_l^d, \lambda_l$  and  $\lambda_l^d$  ( $l = 0, \dots, k-1$ ) to be known, the sequence of computation at order  $k$  is:

- (a) Evaluation of  $\mathbf{R}^d(u, \lambda, p, u^d, \lambda^d, p^d, k)$  with  $u_k = u_k^d = 0$  and  $\lambda_k = \lambda_k^d = 0$  to get  $\mathcal{R}_k$  and  $\mathcal{R}_k^d$ ,
- (b) Solution of  $K_T \hat{u} = -\mathcal{R}_k$ ,
- (c) Evaluation of  $\mathbf{R}^d(u, \lambda, p, u^d, \lambda^d, p^d, 1)$  with  $u_1 = \hat{u}, u_1^d = 0$  and  $\lambda_1 = \lambda_1^d = 0$  to get  $K_T^d \hat{u} = \mathcal{R}_1^d$ ,
- (d) Solution of  $K_T \hat{u}^d = -\mathcal{R}_k^d - K_T^d \hat{u}$ ,
- (e) Path computations appearing in Eq. (5) are deduced from a differentiation in  $p$  of the ANM ones.

One notices that, in item (c), the product  $K_T^d \hat{u}$  of the sparse Jacobian  $K_T^d$  by the direction  $u_1 = \hat{u}$  is evaluated by means of a call to the tangent linear code  $\mathcal{R}^d$  overloaded with Diamant. This second order derivative is thus equal to the first order Taylor coefficient (with respect to the parameter  $a$ )  $\mathcal{R}_1^d$  of the tangent linear function  $\mathcal{R}^d$  differentiated with respect to the parameter  $p$ .

Exploiting some knowledge on the sensitivity equations (5), this differentiation of the ANM/Diamant solver avoids the differentiation of the linear solver used on  $K_T$  following both the ANM and AD spirit. Implementation details (as well as algorithmic complexities) will be discussed in an extended paper. Meanwhile, Fortran codes related to the Note may be downloaded (LGPL Licence) from the author's webpage.<sup>2</sup>

#### 4. Numerical results

Sensitivities computations are illustrated on the PDE problem:

$$(A(p) - pI)u(p) + \lambda(p) \exp(u(p)) = 0 \quad (6)$$

where  $A$  and  $I$  are respectively the discrete Laplacian matrix and the identity matrix. The computational domain is  $(0, 1)$ . Dirichlet boundary conditions are used at point  $x = 0$  and  $x = 1$ . In (6), the peculiar choice  $p = 0$  leads to the Bratu problem [9]. Fig. 1 plots sensitivities computed for  $p = 0$  and  $p^d = 2$ . Curves  $(u^d, \lambda)$  and  $(u, \lambda^d)$  (Fig. 1A) respectively show the sensitivity in  $p$  of variables  $u(x = 0.5)$  and  $\lambda$ . As expected from (6), a positive perturbation of  $p$  induces a positive perturbation of  $\lambda$ . As presented in Fig. 1B, the limit point moves to the right for positive  $p$ . In addition, Fig. 1B also shows a good agreement between the first order reconstruction  $(u(p) + p^d u^d(p), \lambda(p) + p^d \lambda^d(p))|_{p=0, p^d=2}$  and the exact curve  $(u(p), \lambda(p))|_{p=2}$ .

From a computer point of view, we compared two options. The first one assumes the differentiation in  $p$  (with Tapenade) of a hand-written ANM code. In the second one, the tangent linear code generated by Tapenade is overloaded using Diamant (with a storage of intermediate variables to avoid recalculations [5]). As usual in AD, numerical results are identical up to the machine precision. From the efficiency point of view, the ANM solution of (1)–(2) with Diamant2 requires  $T_{D2} = 1.03$  s meanwhile sensitivity computations require  $T_{D2}^S = 2.86$  s. The ratio  $T_{D2}^S/T_{D2} = 2.78$  agrees with both the theoretical bound of 5 established in [10] and usual numerical ratios [11]. Computations were performed using a tridiagonal solver for the solution of each linear system.

#### 5. Conclusions

Exploiting some knowledge on the ANM solver equations, sensitivity computations may be obtained in an easy and efficient manner through a combination of AD tools. In a first step, the residual code is differentiated with respect to the parameters by a source transformation AD tool (for instance Tapenade) to generate a tangent linear code. In a second step, the Diamant algorithmic pattern is adapted to this tangent linear code. The genericity, reusability and

<sup>2</sup> <http://www.lpmm.fr/charpentier/Diamant>

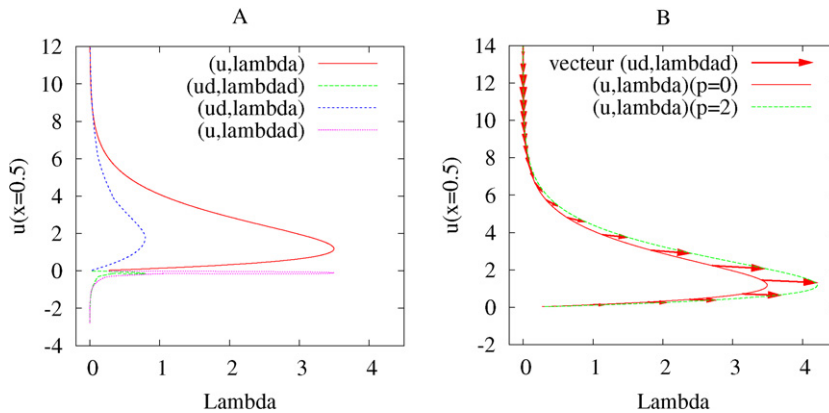


Fig. 1. A: Path  $(u, \lambda)_{|p=0}$ , sensitivities  $(u^d, \lambda^d)_{|p=0, p^d=2}$ , sensitivity of  $u$ , sensitivity of  $\lambda$ . B: Vectors  $(u^d, \lambda^d)_{|p=0, p^d=2}$  plotted along the path  $(u, \lambda)_{|p=0}$ , exact path  $(u, \lambda)_{|p=2}$ .

efficiency properties of Diamant allow for a broad use of the ANM, offering in addition an easy implementation of sensitivity calculations. This differentiation also constitutes a first step towards the identification of parameters (by a gradient method) since tangent multi-directional codes may be generated by Tapenade.

## References

- [1] J.M.T. Thompson, A. Walker, The nonlinear perturbation analysis of discrete structural systems, *International Journal of Solids and Structures* 4 (1968) 757–768.
- [2] B. Cochelin, N. Damil, M. Potier-Ferry, Asymptotic numerical method and Padé approximants for non-linear elastic structures, *International Journal for Numerical Methods in Engineering* 37 (1994) 1187–1213.
- [3] B. Cochelin, N. Damil, M. Potier-Ferry, *Méthode asymptotique numérique*, Hermes Science Publications, 2007, p. 298.
- [4] A. Griewank, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, *Frontiers in Appl. Math.*, vol. 11, SIAM, Philadelphia, 2000, p. 369.
- [5] I. Charpentier, M. Potier-Ferry, Différentiation automatique de la Méthode Asymptotique Numérique Typée : l'approche Diamant, *Comptes Rendus Mecanique* 336 (2008) 336–340.
- [6] I. Charpentier, A. Lejeune, M. Potier-Ferry, The diamant approach for an efficient automatic differentiation of the asymptotic numerical method, in: C.H. Bischof, H.M. Bücker, P. Hovland, U. Naumann, J. Utke (Eds.), *Advances in Automatic Differentiation*, in: *Lecture Notes in Computational Science and Engineering*, vol. 64, 2008, pp. 139–149.
- [7] Y. Koutsawa, I. Charpentier, E.M. Daya, M. Cherkaoui, A generic approach for the solution of nonlinear residual equations. Part I: The Diamant toolbox, *Computer Methods in Applied Mechanics and Engineering*, doi: 10.1016/j.cma.2008.09.003.
- [8] L. Hascoët, V. Pascual, *TAPENADE 2.1 user's guide*, Rapport technique de l'INRIA RT-0300, 2004.
- [9] E. Doedel, H. Keller, J.P. Kernevez, Numerical analysis and control of bifurcation problems (I). Bifurcation in finite dimensions, *International Journal of Bifurcation and Chaos* 1 (1991) 493–520.
- [10] J. Morgenstern, How to compute fast a function and all its derivatives, a variation on the theorem of Baur–Strassen, *SIGACT News* 16 (1985) 60–62.
- [11] I. Charpentier, M. Ghémirès, Efficient adjoint derivatives: Application to the atmospheric model MESO-NH, *Optimization Methods and Software* 13 (2000) 35–63.